

**SISTEMA DE INFORMACIÓN PARA APOYAR EL CONTROL PRENATAL Y  
EPIDEMIOLOGICO EN LA UNIDAD DE MEDICINA MATERNOFETAL DEL  
HOSPITAL UNIVERSITARIO DE SANTANDER.**

**JAIRO ALFONSO SILVA FRANCO  
RAÚL FERNANDO RIVERO OLARTE**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER  
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA  
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS  
BUCARAMANGA**

**2010**

**SISTEMA DE INFORMACIÓN PARA APOYAR EL CONTROL PRENATAL Y  
EPIDEMIOLOGICO EN LA UNIDAD DE MEDICINA MATERNOFETAL DEL  
HOSPITAL UNIVERSITARIO DE SANTANDER.**

**JAIRO ALFONSO SILVA FRANCO  
RAÚL FERNANDO RIVERO OLARTE**

**Trabajo de grado para optar al título de  
Ingeniero de Sistemas**

**Director**

**VÍCTOR EDUARDO MARTÍNEZ ABAUNZA  
Magíster en Ingenierías Área Informática y Ciencia**

**Codirector**

**CARLOS HERNÁN BECERRA MOJICA  
Md. GO Perinatólogo**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER  
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA  
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS  
BUCARAMANGA**

**2010**

## DEDICATORIA

*Mi madre Nelly Cristina Olarte Vargas pues ha sido la mujer más comprensible y cariñosa que he conocido, siempre apoyándome y colaborándome en todo para poder salir adelante y formarme como una persona de bien.*

*Mi padre Raúl Rivero Silva quien siempre ha sido un modelo a seguir, un gran hombre, responsable, honesto, trabajador y entregado al bienestar de su familia.*

*Mi hermana María Cristina Rivero Olarte que siempre ha estado pendiente de mí ofreciéndome su ayuda en los momentos en que la he requerido.*

*Mi tía Gloria Elena Olarte Vargas quien siempre ha estado dispuesta a colaborar en cualquier cosa que fuere necesaria para poder terminar mi carrera profesional.*

*Mis abuelos tanto maternos como paternos quienes ayudaron a mis padres en mi proceso de crianza.*

*Mis amigos y compañeros con quienes he vivido y compartido tantas experiencias a lo largo de mi estancia en la universidad.*

*Raúl Fernando Rivero Olarte*

*De todo corazón a mis padres Jairo Silva Urrea y Elvira Franco Ardila por brindarme su amor, comprensión, escucha y apoyo incondicional en todas las etapas de mi vida, por su inagotable dedicación al trabajo para que nunca me faltara nada por sus infinitas enseñanzas y por inculcarme los valores que me forjaron como hombre y ser humano.*

*A mi hermana Angélica Lilitana, por ser mi amiga y nunca dejarme atrás, llevando el estandarte de esta familia y quien siempre será un modelo a seguir.*

*A mis mejores amigos Carlos Gilberto y Laura Isabel por haber compartido conmigo estos cinco años de lucha, dedicación y esfuerzo, alegrías triunfos y derrotas, sueños y esperanzas. Por escucharme, creer en mí y brindarme su apoyo en los momentos más trascendentales de mi vida universitaria.*

*Al colectivo Fusol-UTOS, por enseñarme tantos paradigmas :P, y darme la visión necesaria para generar el desarrollo social desde mi profesión.*

*A mis amigos, compañeros, profesores y maestros... y en general a todas las personas que de alguna u otra forma aportaron en mi formación como profesional.*

*Jairo Alfonso Silva Franco*

## AGRADECIMIENTOS

Los autores del proyecto expresan sus más sinceros agradecimientos a:

A Víctor E. Martínez, director de este proyecto, quien nos ofreció la oportunidad de realizar el presente proyecto y nos colaboró con sus conocimientos y experiencia.

Al Dr. Carlos H. Becerra, codirector del proyecto, quien aportó parte de su tiempo para la correcta elaboración del proyecto y fue nuestro guía en todo lo que respecta a medicina materna.

A la Dra. Mónica Beltrán médico especialista de la unidad, ya que hizo parte activa del comité de sistemas de información aportando ideas que contribuyeron a la solución de problemas durante el proceso de desarrollo.

Al Dr. Orian Fontalvo médico residente de la unidad, que nos ha enseñado todos los procesos que se manejan en la unidad lo cual ha sido de vital importancia para el buen desarrollo del proyecto.

A nuestro compañero Hugo A. Gualdrón, quien gracias a su amplio conocimiento tecnológico nos aportó y colaboró con su vasta experiencia de manera desinteresada.

A la Comunidad Universitaria de Software Libre, por ser un espacio en el cual se lucha desinteresadamente por un ideal, un lugar en el cual hemos vivido experiencias con personas que al igual que nosotros comparten una pasión por la filosofía del software libre.

A la Universidad Industrial de Santander, que ha sido como nuestro segundo hogar en el cual hemos crecido a nivel tanto personal como profesional.

Al Grupo de Investigación en Ingeniería Biomédica, que ha contribuido mediante el desarrollo de actividades investigativas a la solución de problemas en las áreas de

salud y otras líneas de investigación, con el fin de contribuir a la construcción de conocimiento.

A todas aquellas personas que colaboraron y participaron en el desarrollo de este proyecto ya sea directa o indirectamente.

## GLOSARIO

**Licencia de software:** Contrato entre el licenciante y el licenciario del programa informático, para utilizar el software cumpliendo una serie de términos y condiciones establecidas dentro de las cláusulas.

**Sistema de información:** Conjunto de elementos relacionados entre sí, que se encarga de procesar ya sea manual o automáticamente datos, en función de determinados objetivos.

**Framework:** Estructura de soporte definida, mediante la cual otro proyecto de software puede ser organizado y desarrollado.

**UML:** Es un lenguaje de modelado de sistemas de software, utilizado para visualizar, especificar, construir y documentar un sistema.

**IDE:** Programa informático compuesto por un conjunto de herramientas de programación, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Puede dedicarse en exclusiva a un sólo lenguaje de programación o poder utilizarse para varios.

**ERS:** Es una descripción completa del comportamiento del sistema que se va a desarrollar.

**Perinatología:** Rama de la gineco-obstetricia que se encarga del estudio y control de los embarazos de alto riesgo, durante todo el embarazo y hasta el nacimiento del bebe y el control posterior.

**Patología:** Estudio de las enfermedades en su más amplio sentido, es decir, como procesos o estados anormales de causas conocidas o desconocidas.

**Ultrasonido:** Es una onda acústica cuya frecuencia está por encima del límite perceptible por el oído humano. A este fenómeno también se le conoce como ecolocalización. Se trata de que las ondas emitidas sean tan altas que rebotan fácilmente en todos los objetos alrededor del punto emisor, esto hace que se cree una imagen.

## CONTENIDO

INTRODUCCIÓN .....	1
1. ESPECIFICACIONES DEL PROYECTO .....	2
1.1. TÍTULO .....	2
1.2. MODALIDAD .....	2
1.3. DIRECCIÓN .....	2
1.4. AUTORES .....	3
1.5. ENTIDADES INTERESADAS EN EL PROYECTO .....	4
2. PLANTEAMIENTO DEL PROYECTO .....	5
2.1. OBJETIVOS .....	5
2.1.1. OBJETIVO GENERAL .....	5
2.1.2. OBJETIVOS ESPECÍFICOS .....	5
2.2. DESCRIPCIÓN DE LA SITUACIÓN ACTUAL .....	6
2.3. JUSTIFICACIÓN .....	7
2.4. IMPACTO Y VIABILIDAD .....	8
2.4.1. IMPACTO .....	8
2.4.2. VIABILIDAD .....	8
3. MARCO TEÓRICO .....	10
3.1. CONCEPTOS RELACIONADOS CON EL CONTROL PRENATAL .....	10
3.1.1. EL CONTROL PRENATAL .....	10
3.1.2. PERINATOLOGÍA .....	11
3.1.3. MALFORMACIONES CONGÉNITAS .....	13
3.1.3.1. MALFORMACIONES CONGÉNITAS ENCEFÁLICAS .....	15
3.1.4. ULTRASONIDO FETAL .....	16
3.1.5. EPIDEMIOLOGÍA .....	17
3.1.5.1. EPIDEMIOLOGÍA PERINATAL .....	18

3.2.	MARCO TECNOLÓGICO.....	19
3.2.1.	LOS SISTEMAS DE INFORMACIÓN .....	20
3.2.1.1.	LOS SISTEMAS DE INFORMACIÓN BASADOS EN WEB .....	21
3.2.2.	LA ARQUITECTURA DE UN SISTEMA.....	22
3.2.2.1.	LA ARQUITECTURA CLIENTE-SERVIDOR .....	23
3.2.2.2.	ARQUITECTURA P2P .....	26
3.2.3.	EL HARDWARE .....	26
3.2.3.1.	EL ECÓGRAFO .....	26
3.2.3.2.	EL SERVIDOR .....	27
3.2.4.	EL SOFTWARE.....	28
3.2.4.1.	LAS LICENCIAS DE SOFTWARE.....	28
3.2.4.2.	LENGUAJES DEL LADO DEL SERVIDOR.....	31
3.2.4.3.	JAVA .....	33
3.2.4.4.	LA API.....	34
3.2.4.5.	LOS FRAMEWORK.....	35
4.	COMPONENTES TECNOLÓGICOS UTILIZADOS EN LA ELABORACIÓN DEL SISTEMA DE INFORMACIÓN .....	37
4.1.	HERRAMIENTAS DE LA FASE DE DISEÑO.....	37
4.1.1.	ARGOUML .....	37
4.1.2.	MYSQL WORKBENCH .....	38
4.2.	TECNOLOGÍAS Y HERRAMIENTAS DE LA FASE DE DESARROLLO .....	38
4.2.1.	APACHE STRUTS .....	38
4.2.2.	HIBERNATE .....	39
4.2.2.1.	MAPEO OBJETO-RELACIONAL.....	40
4.2.3.	MYSQL.....	41
4.2.4.	CKEDITOR.....	41

4.2.5.	JASPERREPORTS .....	42
4.2.6.	HTML .....	43
4.2.7.	JSP.....	43
4.2.8.	ECLIPSE.....	44
5.	DESCRIPCIÓN DE LOS MÓDULOS DEL SISTEMA .....	46
5.1.	MÓDULO DE GESTIÓN DE USUARIOS .....	46
5.1.1.	DESCRIPCIÓN .....	46
5.1.2.	REQUISITOS ASOCIADOS.....	47
5.1.3.	FUNCIONES .....	47
5.2.	MÓDULO DE CASOS DE LA UMMFN .....	49
5.2.1.	DESCRIPCIÓN .....	49
5.2.2.	REQUISITOS ASOCIADOS.....	49
5.2.3.	FUNCIONES .....	49
5.3.	MÓDULO DE GESTIÓN DE PATOLOGÍAS GENERALES.....	51
5.3.1.	DESCRIPCIÓN .....	51
5.3.2.	REQUISITOS ASOCIADOS.....	51
5.3.3.	FUNCIONES .....	51
5.4.	MÓDULO DE CONSULTAS.....	52
5.4.1.	DESCRIPCIÓN .....	52
5.4.2.	REQUISITOS ASOCIADOS.....	53
5.4.3.	FUNCIONES .....	53
5.5.	MÓDULO DE INFORMES Y EPIDEMIOLOGÍA .....	56
5.5.1.	DESCRIPCIÓN .....	56
5.5.2.	REQUISITOS ASOCIADOS.....	56
5.5.3.	FUNCIONES .....	56
5.6.	MÓDULO DE GESTIÓN DE COPIAS DE SEGURIDAD.....	57

5.6.1.	DESCRIPCIÓN .....	57
5.6.2.	REQUISITOS ASOCIADOS.....	57
5.6.3.	FUNCIONES .....	58
6.	METODOLOGÍA .....	59
6.1.	RUP.....	59
6.1.1.	FASES DEL RUP.....	61
6.1.2.	CASOS DE USO .....	68
6.1.3.	DIAGRAMAS DE CASOS DE USO.....	70
6.2.	ARQUITECTURA DE LA APLICACIÓN .....	72
6.3.	ITERACIONES DE LA FASE DE CONSTRUCCIÓN .....	73
6.4.	CRONOGRAMA DE ACTIVIDADES PARA UNA ITERACIÓN.....	74
7.	DESARROLLO .....	80
7.1.	ANÁLISIS .....	80
7.2.	DISEÑO .....	81
7.2.1.	DIAGRAMA DE CASOS DE USO .....	81
7.2.2.	DIAGRAMA DE DESPLIEGUE .....	82
7.2.3.	DIAGRAMAS DE ACTIVIDADES .....	83
7.2.4.	DIAGRAMA DE PAQUETES.....	85
7.2.5.	DIAGRAMA ENTIDAD-RELACIÓN.....	86
7.2.6.	DIAGRAMA DE CLASES.....	88
7.3.	IMPLEMENTACIÓN .....	90
7.4.	PRUEBAS.....	95
7.4.1.	PRUEBAS FUNCIONALES .....	95
7.4.2.	PRUEBAS DE CARGA .....	96
7.4.3.	PRUEBAS DE SEGURIDAD .....	97
8.	DOCUMENTACIÓN Y MANUALES.....	99

8.1.	MANUAL DE INSTALACIÓN DE FETHUS .....	99
8.2.	MANUALES DE USUARIO Y DE ADMINISTRADOR .....	101
8.3.	MANUAL DEL PROGRAMADOR.....	102
9.	CONCLUSIONES .....	103
10.	RECOMENDACIONES .....	105
11.	BIBLIOGRAFÍA .....	107

## LISTA DE TABLAS

Tabla 1. Caso de uso “ <i>crear cuentas de usuario</i> ” .....	70
Tabla 2. Cronograma de actividades para la semana 1. ....	74
Tabla 3. Cronograma de actividades para la semana 2. ....	75
Tabla 4. Cronograma de actividades para la semana 3. ....	76
Tabla 5. Tiempos de conexión a la aplicación. ....	96
Tabla 6. Intervalos de tiempo clasificados por porcentajes. ....	97
Tabla 7. Resultado final de la prueba de carga. ....	97

## LISTA DE FIGURAS

Figura 1. Malformaciones más frecuentes. OMS. ....	14
Figura 2. Procedimiento ecográfico. ....	16
Figura 3. Patrón Modelo Vista Controlador. ....	35
Figura 4. Captura del menú para la gestión de usuarios del sistema. ....	47
Figura 5. Captura de registro de un nuevo usuario o modificación de uno existente. ....	48
Figura 6. Captura del módulo de administración de casos UMMFN. ....	50
Figura 7. Captura de actualización o registro de un paciente nuevo. ....	50
Figura 8. Captura del módulo de patologías generales. ....	52
Figura 9. Captura de consulta de usuarios. ....	53
Figura 10. Captura de consulta de Casos de la UMMFN. ....	54
Figura 11. Captura de consulta de pacientes de la UMMFN. ....	55
Figura 12. Captura de consulta por Patologías Generales. ....	55
Figura 13. Reporte de utilización del sistema por parte de los usuarios. ....	57
Figura 14. Realización del backup a la base de datos. ....	58
Figura 15. Cargar backup de la base de datos. ....	58
Figura 16. Los Casos de Uso integran el trabajo ....	60
Figura 17. Una iteración RUP ....	61
Figura 18. Estructura de RUP. ....	62
Figura 19. Plataforma operativa de FETHUS. ....	72
Figura 20. Diagrama de despliegue de FETHUS. ....	82
Figura 21. Diagrama de actividades que muestra el proceso de respuesta a las peticiones del cliente desde el uso de la interfaz grafica de usuario hasta que se utiliza la base de datos. ....	83

Figura 22. Diagrama de actividades que muestra el proceso de salida de los resultados hacia el cliente, efectuado por FETHUS, desde la consulta en la base de datos hasta la presentación final.....	85
Figura 23. Diagrama de paquetes de FETHUS. ....	86
Figura 24. Diseño físico de la base de datos, presentado en forma de diagrama referencial. ....	87
Figura 25. Diagrama de clases general. ....	88
Figura 26. Diagrama de clases del Modelo.....	89
Figura 27. Captura del paquete persistentes y sus clases.....	91
Figura 28. Método “ <i>almacenar</i> ” de la clase Model. ....	91
Figura 29. Clase Criteria que especifica métodos para realizar operaciones lógicas en las consultas .....	92
Figura 30. Archivo de configuración Struts-config.xml. ....	94
Figura 31. Captura de pantalla de los directorios del proyecto que se encuentra alojado en el servidor de versiones de google code. ....	95
Figura 32. Captura del servicio de Tomcat funcionando. ....	99
Figura 33. Captura del inicio de sesión en el “Tomcat Manager Application.” .....	100
Figura 34. Captura de la carga del archivo WAR que contiene la Aplicación Web. ....	100
Figura 35. Captura de la Aplicación Web cargada en el servidor.....	101
Figura 36. Captura del video elaborado como manual de usuario y administrador. ....	102
Figura 37. Diagrama de Casos de Uso para la Gestión de Usuarios.....	120
Figura 38. Diagrama de Casos de Uso para la Gestión de Casos.....	120
Figura 39. Diagrama de Casos de Uso para la Gestión de Patologías.....	121
Figura 40. Diagrama de Casos de Uso para el módulo de Consultas. ....	121
Figura 41. Diagrama de Casos de Uso para la generación de informes.....	122
Figura 42. Diagrama de Casos de Uso para la Gestión de Copias de Seguridad. ....	122

## LISTA DE ANEXOS

ANEXO A. DOCUMENTO DE ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE.....	109
ANEXO B. ACTA DE ENTREGA DEL SISTEMA.....	170

## RESUMEN

### TITULO:

SISTEMA DE INFORMACIÓN PARA APOYAR EL CONTROL PRENATAL Y EPIDEMIOLOGICO EN LA UNIDAD DE MEDICINA MATERNOFETAL DEL HOSPITAL UNIVERSITARIO DE SANTANDER\*

### AUTORES:

SILVA FRANCO, Jairo Alfonso.

RIVERO OLARTE, Raúl Fernando\*\*

### PALABRAS CLAVE:

MaternoFetal, patología, software, ecografía, UML, MySQL, JSP, Web.

### DESCRIPCIÓN:

La Unidad de Medicina MaternoFetal y Neonatal del Hospital Universitario de Santander cuenta con personal médico capacitado, quienes se encargan de hacer especial énfasis en factores de riesgo, hábitos y enfermedades que pudiesen complicar el embarazo en términos maternos y fetales y emite estrategias para controlarlos o eliminarlos mediante seguimientos periódicos a las pacientes, prestando especial atención a los casos que puedan presentar complicaciones obstétricas de alto riesgo.

El sistema de información FETHUS Web, es una herramienta software que permite el almacenamiento de información relevante concerniente a los pacientes que presentan malformaciones fetales brindando la opción de generar informes y analizar los datos para futuros estudios estadísticos. Esta herramienta, diseñada y generada con base a las necesidades particulares de la Unidad de Medicina MaternoFetal y Neonatal del Hospital Universitario de Santander, se constituye como una herramienta de carácter tanto educativo como aplicativo; que permite apoyar el diagnóstico y tratamiento a pacientes.

El sistema de información es altamente parametrizable y además cuenta con diferentes módulos tanto para la consulta, inserción o eliminación de información referente a pacientes dependiendo de los permisos asignados por el superadministrador del sistema; su elaboración fue llevada a cabo utilizando la metodología de "*Rational Unified Process*" y tecnología "*Java Enterprise Edition*", y estará alojado en el servidor del Grupo de Investigación en Ingeniería Biomédica perteneciente a la Escuela de Ingeniería de Sistemas e Informática.

---

\* Trabajo de Grado en la Modalidad de Investigación

\*\* Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería de Sistemas e Informática.  
Director: MSc. Víctor Eduardo Martínez. Codirector: Md. GO Carlos Hernán Becerra.

## **SUMMARY**

### **TITLE:**

INFORMATION SYSTEM SUPPORT FOR PRENATAL AND EPIDEMIOLOGIC CONTROL IN MATERNAL FETAL MEDICINE UNIT AT "HOSPITAL UNIVERSITARIO DE SANTANDER".\*

### **AUTHORS:**

SILVA FRANCO, Jairo Alfonso.

RIVERO OLARTE, Raúl Fernando\*\*

### **KEYWORDS:**

Maternalfetal, pathology, software, ultrasound, UML, MySQL, PHP, Web.

### **ABSTRACT:**

The Unit for Maternal and Neonatal Medicine at "Hospital Universitario de Santander" has qualified medical personal who are responsible for placing special emphasis on risk factors, habits and diseases that could complicate pregnancy in fetal and maternal terms providing strategies to controlling it or eliminating it through regular follow-ups to patients, paying particular attention to cases that may present high risk obstetric complications.

The information system FETHUS Web, is a software tool that allows storage of relevant information about the patients with fetal malformations by providing the option to generate reports and analyze data for future statistical studies. This tool, designed and produced based on the needs of the Unit for Maternal Fetal and Neonatal Medicine at "Hospital Universitario de Santander", is constituted as an applicable and educational tool; that allows supporting the diagnosis and treatment to patients.

The information system is highly customizable and also has different modules for query, insertion or deletion of information about patients depending on the permissions assigned by the SuperAdmin, its development was carry out using the methodology of "Rational Unified Process" and "Java Enterprise Edition" technology and will be hosted on the server of the Research Group Biomedical Engineering belonging of the School of Engineering and Computer Science.

---

\* Graduation paper, Research mode

\*\* Faculty of Physical-Mechanical Engineering. School of Engineering and Computer Science. Director: MSc. Víctor Eduardo Martínez. Codirector: Md. GO Carlos Hernán Becerra.

## ***INTRODUCCIÓN***

El presente documento recopila textualmente el resultado de varios meses de trabajo durante los cuales se ha llevado a cabo el proceso para el desarrollo del Sistema de Información para apoyar el control prenatal y epidemiológico en la Unidad de Medicina MaternoFetal del Hospital Universitario de Santander FETHUS Web.

Este proyecto tiene como objetivo llevar seguimiento a pacientes de Unidad de Medicina MaternoFetal y Neonatal para así poder brindar un servicio de calidad en cuanto al diagnóstico y tratamiento, agilizando los procesos que se llevan a cabo en la Unidad. Además de mejorar la atención al paciente, se podrán realizar estudios a futuro con la información recopilada de dichos pacientes.

En este libro la Unidad de Medicina MaternoFetal y Neonatal será referida como “la UMMFN”, los autores del proyecto como “los autores” y el sistema desarrollado será referido como “FETHUS”. Se presenta al lector las especificaciones del proyecto y el planteamiento del proyecto (capítulo 1 y 2) en donde se describe a grandes rasgos tanto los objetivos como los alcances del proyecto, con lo cual se busca que el lector tenga una idea global del proyecto elaborado.

Se prosigue con la fundamentación teórica (capítulo 3), el marco legal en cuanto a distribución del sistema (capítulo 4), los componentes tecnológicos empleados para el desarrollo (capítulo 5), la descripción modulo a modulo del sistema (capítulo 6), la metodología utilizada durante el proceso de desarrollo (capítulo 7), desarrollo como tal del proyecto (capítulo 9), y los manuales de usuario los cuales sirven como guía para el correcto uso al sistema de información (capítulo 10).

## **1. ESPECIFICACIONES DEL PROYECTO**

### **1.1. TÍTULO**

Sistema de Información para apoyar el control prenatal y epidemiológico en la Unidad de Medicina MaternoFetal del Hospital Universitario de Santander.

### **1.2. MODALIDAD**

Investigación

### **1.3. DIRECCIÓN**

#### **DIRECTOR**

Nombre : Víctor Eduardo Martínez Abaunza  
Profesión : Magíster en Ingenierías Área Informática y Ciencia  
Institución : Universidad Industrial de Santander  
Cargo : Docente Escuela de Ingeniería de Sistemas e Informática

## **CODIRECTOR**

Nombre : Carlos Hernán Becerra Mojica  
Profesión : Md. GO Perinatólogo  
Institución : Universidad Industrial de Santander  
Cargo : Docente Escuela de Medicina

### **1.4. AUTORES**

Nombre : Jairo Alfonso Silva Franco  
Código : 2050216  
Carrera : Ingeniería de Sistemas

Nombre : Raúl Fernando Rivero Olarte  
Código : 2022096  
Carrera : Ingeniería de Sistemas

### **1.5. ENTIDADES INTERESADAS EN EL PROYECTO**

- Escuela de Ingeniería de Sistemas e Informática de la Universidad Industrial de Santander
- Grupo de Investigación en Ingeniería Biomédica
- Facultad de Salud de la Universidad Industrial de Santander
- Unidad de Medicina MaternoFetal del Hospital Universitario de Santander
- Unidad de Recién Nacidos Hospital Universitario de Santander
- Demás entidades relacionadas

## **2. PLANTEAMIENTO DEL PROYECTO**

### **2.1. OBJETIVOS**

#### **2.1.1. OBJETIVO GENERAL**

Construir una herramienta software basada en la web para contribuir a la organización de la Unidad de Medicina MaternoFetal del Hospital Universitario de Santander, brindar apoyo al diagnóstico de pacientes en estado de embarazo, gestionar el control estadístico y epidemiológico y funcionar como centro de consulta al estudiantado de la Facultad de Medicina de la Universidad Industrial de Santander.

#### **2.1.2. OBJETIVOS ESPECÍFICOS**

- Diseñar una base de datos y las interfaces de usuario con base en el análisis de requisitos de software y las especificaciones de diseño.
- Desarrollar un Sistema de Información funcional con módulos para la gestión de patologías y enfermedades perinatales, gestión de casos que se presenten en la UMMFN, gestión de pacientes, gestión de informes y gestión de usuarios del sistema.
- Brindar a la UMMFN una herramienta para gestionar el archivo gráfico de los pacientes obtenido en cada una de las sesiones ecográficas, solucionando las limitaciones de los recursos hardware para almacenar dicha información.
- Elaborar la documentación que sirva como referencia para el usuario sobre el uso del Sistema de Información.
- Elaborar pruebas de funcionalidad y carga pico a la aplicación web.

## **2.2. DESCRIPCIÓN DE LA SITUACIÓN ACTUAL**

La misión de la UMMFN<sup>1</sup>, es atender a pacientes en estado de gestación y hacerle seguimientos periódicos a los casos de embarazo de alto riesgo que puedan presentar complicaciones obstétricas, médicas o quirúrgicas en esta etapa.

El personal médico a cargo de la UMMFN realiza diagnósticos a partir de la captura de las medidas morfológicas internas y externas del feto, utilizando ultrasonido obstétrico y almacenando imágenes, videos, histogramas, gráficas de crecimiento, y otros datos importantes en el seguimiento perinatal.

Cuando se encuentran frente a un nuevo caso de malformación fetal, los asistentes médicos, registran mediante formularios en papel, información del paciente tales como datos personales, antecedentes patológicos, quirúrgicos, farmacológicos, ginecólogos, entre otros; para posteriormente, almacenarla en una hoja de cálculo donde se centralizan los datos para su análisis estadístico.

Los médicos suscritos a la UMMFN, realizan un informe detallado de cada anomalía acontecida, con motivo de socializar dichos casos que fueron encontrados durante la prestación del servicio, el cual se expone ante la Junta de Diagnóstico Prenatal de Malformaciones Fetales y Anomalías Congénitas, quienes se reúnen semanalmente debido al alto número de anomalías presentadas y toman decisiones sobre las pacientes.

Actualmente no existe ningún sistema de información computarizado en la UMMFN, de carácter tanto administrativo como educativo que contribuya a la automatización de los procesos que se realizan a diario en dicha entidad, es por esto, que se reconoce la necesidad de desarrollar una herramienta basada en la web, que permita tanto a estudiantes de la Facultad de Medicina de la Universidad Industrial de Santander como al personal de la UMMFN, el acceso a información

---

<sup>1</sup> Unidad de Medicina MaternoFetal del Hospital Universitario de Santander.

relacionada a casos anómalos, con la cual se puedan generar posteriormente nuevas investigaciones.

### **2.3. JUSTIFICACIÓN**

Desde mucho antes del uso de las computadoras en las organizaciones, éstas reunían, almacenaban y actualizaban información en el transcurso normal de su diario desempeño. Los sistemas de información son procedimientos y reglas establecidas para generar y entregar información a los miembros de determinada organización. Cada una de estas personas, desempeña un determinado rol y requiere información distinta para la realización de su trabajo, las reglas definidas dentro del sistema son las que indican el tipo, momento, formato y cuál es la persona a quien se debería entregar determinada información.

Tratar manualmente un sistema de información, incluso en organizaciones pequeñas donde el flujo de datos e información no es significativamente grande, puede llegar a ser un procedimiento bastante ineficiente y frustrante.

Los sistemas de información automatizados o basados en computadoras, están conformados por hardware, software, personas, procedimientos y datos. Todos estos elementos se conjugan, trabajando juntos, brindando información relevante para la conducción y la toma de decisiones en la empresa u organización. Esto hace posible a las empresas llevar a cabo sus tareas con mayor calidad y facilidad.

Los sistemas de información computarizados, tienen diferentes capacidades implementadas tales como llevar un seguimiento de las transacciones y operaciones diarias, propias de la entidad, además de servir de apoyo al flujo de información interno de la organización. La finalidad de los sistemas de información organizacionales es procesar entradas, mantener archivos de datos relacionados con la organización y producir información, reportes y otras salidas para los

usuarios que las necesitan.

Es por esto que se hace visible la necesidad de implementar un Sistema de Información, desde la perspectiva de la eficiencia, para brindar apoyo en los procesos diarios que se realizan en la UMMFN del Hospital Universitario de Santander, para así, brindar un mejor servicio a la población que se beneficia de los servicios prestados en esta entidad, además de facilitar al personal médico información necesaria para la toma de decisiones con respecto a los casos que se presenten.

## **2.4. IMPACTO Y VIABILIDAD**

### **2.4.1. IMPACTO**

La elaboración del presente proyecto tendrá un impacto positivo en los estudiantes e investigadores de la Facultad de Medicina de la Universidad Industrial de Santander, ya que tendrán una nueva fuente de estudio e información relacionada con la perinatología, y a partir de esta se podrán desarrollar nuevas investigaciones en el campo de la medicina MaternoFetal.

El personal de la UMMFN será ampliamente beneficiado, debido a que podrán utilizar el sistema de información para soportar su trabajo, y realizar sus procesos eficientemente, puesto que pasarían de trabajar con un sistema de información manual a un sistema de información computarizado.

### **2.4.2. VIABILIDAD**

El personal de la UMMFN se ha mostrado entusiasta por la proposición del presente proyecto y están dispuestos a colaborar en todo sentido durante el proceso de diseño, análisis y desarrollo; aportando de su tiempo y abriéndoles las puertas a los autores del presente proyecto para poder observar y tener una visión

clara de las necesidades que se presentan.

El uso de herramientas de desarrollo libres tales como IDE's<sup>2</sup>; manejadores de bases de datos, y frameworks, entre otros, y lenguajes robustos enfocados a la web, representan un ahorro sustancial en el pago de licencias de uso, convirtiéndose en un factor importante en la viabilidad del presente proyecto, además de permitir la futura actualización y elaboración de nuevos módulos al producto final que será el sistema de información a desarrollar.

---

<sup>2</sup> Integrated Development Environment (Entorno de Desarrollo Integrado).

### **3. MARCO TEÓRICO**

#### **3.1. CONCEPTOS RELACIONADOS CON EL CONTROL PRENATAL**

En esta sección se introducen algunos conceptos básicos de epidemiología, perinatología y control prenatal con el objetivo de familiarizar al lector con los términos médicos utilizados en este libro. Dichos conceptos serán necesarios para comprender el significado de los datos manejados por el sistema.

##### **3.1.1. EL CONTROL PRENATAL**

El control prenatal se define como el conjunto de acciones y procedimientos sistemáticos y periódicos, con la finalidad de prevenir y diagnosticar las posibles variables (patologías) que conllevan a la muerte de la madre o del feto, especialmente en la etapa perinatal.

Actualmente las estadísticas<sup>3</sup> de mortalidad perinatal [(TMP: El número de muertes fetales después de 28 semanas de embarazo (muertes fetales tardías) mas el número de muertes de niños menores de 7 días por cada 1.000 nacidos vivos.)], demuestran un aumento en el número de patologías que afectan al feto y así como sus posibles causas, los factores son muy diversos, desde la condición social y laboral hasta factores genéticos y antecedentes familiares.

Para la detección de las anomalías y elaboración del diagnóstico, el médico perinatólogo se vale de la historia clínica de la paciente en conjunto con los datos obtenidos mediante la ecografía obstétrica. Durante el embarazo, para analizar la condición del feto, se realizan ecografías mensuales hasta la semana 28 de

---

<sup>3</sup> Neonatal and Perinatal Mortality: Country, Regional and Global Estimates ©World Health Organization. 2006

gestación, quincenales a partir de la semana 28 de gestación y semanales a partir de las 36 semanas. Siendo las realizadas en el último trimestre las más importantes, pues se hacen evidentes las malformaciones y patologías que afectan los sistemas más importantes del cuerpo.

En general, el control prenatal debe ser:

*Precoz:* tan pronto la paciente se entera de que está encinta, para poder predecir la edad gestacional, realizar acciones de prevención y detectar los posibles factores de riesgo.

*Periódico:* la frecuencia depende directamente del nivel de riesgo las condiciones médicas y psicológicas.

*Completo:* se debe realizar un diagnóstico con base en el estado general de la mujer y no debe limitarse al feto.

### **3.1.2. PERINATOLOGÍA**

La Perinatología es una rama de la gineco-obstetricia que se encarga del estudio y control de los embarazos de alto riesgo, durante todo el embarazo y hasta el nacimiento del bebe y el control posterior. Requiere de la participación de diversas profesiones: médicos obstetras, ginecólogos y neonatólogos, obstetras, enfermeras, psicólogos, genetistas, etc.

El periodo perinatal comprende entre las 28 semanas de gestación y los 28 primeros días de vida. El periodo neonatal es el espacio de tiempo que comprende desde el momento en que el bebé nace hasta los 28 días después del nacimiento; según la Organización Mundial de la Salud, el 50% de las muertes neonatales, tienen como causa el bajo peso de nacimiento y su patología propia o las malformaciones congénitas que se hacen evidentes al momento del nacimiento.

Las principales definiciones de la terminología utilizadas en la perinatología son:

- *Edad gestacional*: es el número completo de semanas que han transcurrido entre el primer día de la última menstruación y la fecha del parto. Debe ser confirmada mediante una ecografía antes de las 26 semanas.
- *Nacido vivo*: Extracción completa del cuerpo de la madre, independiente de la duración del embarazo, el cual después de la separación muestre señales de vida como latidos cardíacos, respiración etc.
- *Muerte fetal*: Muerte ocurrida antes del parto, independientemente de la duración del embarazo.
- *Muerte fetal precoz*: (aborto) Expulsión o extracción de un feto o embrión, con una edad gestacional menor a 22 semanas o con un peso inferior a los 500 g.
- *Muerte fetal tardía*: (mortinato) Muerte antes del parto en que la edad gestacional es mayor a 22 semanas o el peso es mayor o igual a 500 g.
- *Inmaduro*: Neonato entre 20 y 28 semanas de gestación.
- *Pretérmino*: Neonato con menos de 37 semanas de gestación o menor de 253 días, desde el primer día de la última regla.
- *Prematuro extremo*: Neonato con menos de 750 g de peso o con una edad gestacional inferior a 26 semanas.
- *Termino*: Neonato nacido entre la semana 37 y el último día de la semana 42.
- *Postérmino*: Neonato nacido después de la semana 42.
- *Peso insuficiente al nacer*: Neonatos que pesan menos de los 3000 g.
- *Bajo peso al nacer*: Neonatos con menos de 2500 g.
- *Muy bajo peso al nacer*: Neonatos con peso inferior a 1500 g.

- *Extremo bajo peso al nacer*: Neonatos con peso inferior a 1000 g.
- *Microneonatos*: Neonatos con peso inferior a los 750 g.
- *Mortalidad neonatal*: Cantidad de recién nacidos fallecidos antes de llegar a los 28 días de vida por cada mil nacidos vivos.
- *Mortalidad perinatal*: Es la suma de la mortalidad neonatal y la Tasa de mortalidad Fetal.

### **3.1.3. MALFORMACIONES CONGÉNITAS**

Las malformaciones congénitas son anomalías morfológicas que se hacen evidentes al momento del parto. El termino congénito no indica pero tampoco descarta un origen genético de la anormalidad, si no que precisa el hecho de que está presente al momento del parto.

Una malformación, es un defecto morfológico de un órgano, parte de este, o de una región del cuerpo, producto de un proceso de desarrollo intrínsecamente anormal. Las causas de las malformaciones suelen ser factores genéticos o hereditarios, pero el uso o presencia de entidades tóxicas durante todo el desarrollo fetal como fármacos, estupefacientes, radiaciones y productos químicos, pueden perturbar el desarrollo normal del feto, ocasionando daños en la correcta forma y funcionamiento de los órganos del cuerpo.

Debido al gran número de malformaciones congénitas, su diagnostico inicial puede establecer dificultades, en muchas ocasiones se puede prever debido a los antecedentes familiares, en otros casos se detecta mediante la práctica de una ecografía prenatal, o utilizando otros métodos como la amniocentesis que consiste en extraer una pequeña parte del liquido amniótico, con el fin de analizarla y determinar si el bebé presenta anomalías cromosómicas (síndrome de down), defectos congénitos graves y enfermedades producidas por un gen anormal.

La clasificación de malformaciones sugerida por Sprangler<sup>4</sup> se divide en dos grandes grupos. Las malformaciones externas o visibles, y las malformaciones internas o no visibles. En los cuales se presentan subgrupos o familias. Dentro del grupo de malformaciones externas podemos encontrar: Defectos aislados como son deformidad, disfunción, displasia, malformación, y defectos múltiples como defecto de campo, síndromes, asociación, secuencia. Las internas o no visibles se encuentran: pueden ser por búsqueda intencionada o hallazgo incidental.

Una disrupción, es un defecto morfológico debido a una alteración extrínseca durante el proceso de desarrollo de un órgano el cual se había estado desarrollando de forma completamente normal. Se debe principalmente a la destrucción de tejido ya formado. La displasia es una perturbación de la estructura celular de un tejido con implicaciones morfológicas. En otras palabras, es proceso y consecuencia de dishistogénesis. Un síndrome es una agrupación randómica de malformaciones cuya causa se desconoce. Se denomina *secuencia* el patrón de anomalías múltiples ocasionadas por una única anormalidad causante de una cadena de complicaciones en la morfo-génesis, ya que el desarrollo del embrión, es un proceso secuencial en el tiempo y en el espacio.

-Malformaciones mas frecuentes

<u>Malformación</u>	<u>Fcia. por c/10000 nacidos vivos</u>
- Alteraciones del pié, sin anomalías del SNC	25.7
- Defecto Septal Ventricular	16.9
- Defecto Septal Auricular	10.9
- Labio Leporino, con o sin Paladar Hendido	9.1
- Espina Bífida (sin anencefalia)	5.5
- Hidrocefalia (sin anencefalia)	4.8
- Anencefalia	3.8
- Deformidad Músculo esquelética	3.5
- Atresia Intestinal y Anal	3.4

Figura 1. Malformaciones más frecuentes. OMS.

---

<sup>4</sup> Sprangler J, Opitz A. Errors of morphogenesis: concepts and terms. Recommendations of an international working group. J Pediatr 1982; 100(1): 160-5.

### **3.1.3.1. MALFORMACIONES CONGÉNITAS ENCEFÁLICAS**

Dentro de las principales malformaciones congénitas encefálicas está presente la hidrocefalia, la macrocefalia, la microcefalia, la anencefalia y la exencefalia. La palabra hidrocefalia viene de las palabras griegas hydro (agua) y céfalo (cabeza) tal y como indica su nombre es una situación en donde hay una cantidad excesiva de líquido en el cerebro, el líquido presente es el líquido cefalorraquídeo. Esta acumulación concentrada de líquido conlleva a la dilatación anormal de los espacios del cerebro llamados ventrículos. Esta dilatación produce una presión altamente dañina en los tejidos del cerebro.

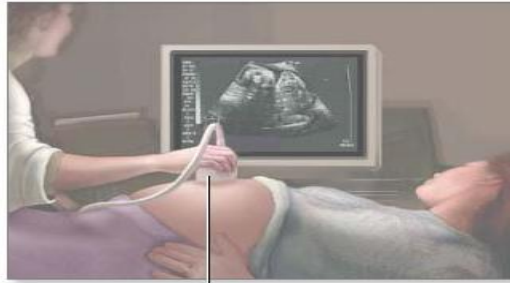
La macrocefalia es una anomalía en la que el perímetro cefálico es mayor al promedio correspondiente al sexo y edad del bebé. Puede ser causada por factores hereditarios. No siempre se le puede relacionar con retraso mental, en la mitad de casos la evolución mental es completamente normal.

La microcefalia es un trastorno neurológico en el cual, el perímetro cefálico es menor al promedio correspondiente al sexo y a la edad del niño. Además de congénita puede manifestarse en los primeros años de edad del infante.

La anencefalia es un defecto del tubo neural el cual se manifiesta cuando la cabeza no logra cerrarse generalmente entre el día 23 y 26 de embarazo, originando la ausencia de una importante parte del cerebro, el cráneo y el cuero cabelludo. La exencefalia es una malformación en la cual el cerebro está ubicado fuera del cráneo, se presenta principalmente en embriones durante las primeras fases de la anencefalia.

Otras malformaciones importantes son la microencefalia, hidraencefalia, octocefalia (trastorno letal), megalencefalia (cerebro excesivamente grande), plagiocefalia, porencefalia, lisencefalia, iniencefalia, holoprosencefalia, ezquisencefalia, escafocefalia, colpocefalia, oxicefalea, braquicefalia y la trigonocefalia.

### 3.1.4. ULTRASONIDO FETAL



Transductor de ultrasonido

Figura 2. Procedimiento ecográfico.

En medicina, el ultrasonido, popularmente conocido como ecografía, es un procedimiento de imagenología, que utiliza los ecos de una emisión de ultrasonidos<sup>5</sup>, enfocada en cuerpo u objeto, para formar una representación gráfica de los órganos y estructuras internas. En medicina, el ultrasonido es utilizado para examinar el corazón, los riñones, el hígado, los pulmones y otros órganos. En la obstetricia, el personal médico usa el ultrasonido para examinar el feto; pues a diferencia de la radiografía, no expone el cuerpo a la radiación ionizante.

Como medio principal de diagnóstico durante el control prenatal, el sonograma abdominal, brinda información detallada del estado del feto; permite conocer el sexo del bebé y realizar un seguimiento al crecimiento y desarrollo de sus sistemas biológicos, como lo son el sistema nervioso central, el sistema cardiovascular, el sistema óseo etc.

La ecografía no conlleva ningún riesgo para el feto o la madre, la embarazada se acuesta sobre una camilla, se le aplica un gel especial sobre la región abdominal y el perinatólogo mueve un transductor sobre la piel que emite señales sonoras de alta frecuencia, que luego un DSP procesa y traduce en imágenes. Sus

---

<sup>5</sup> Onda acústica cuya frecuencia está por encima del espectro audible del oído humano.

principales funciones son confirmar el embarazo; midiendo la circunferencia del cráneo y la longitud del fémur, se confirma que el tamaño es adecuado a la fecha prevista del parto. También se puede averiguar todo tipo de malformaciones, posición de la placenta y alteraciones en el cerebro.

Existen tres tipos de ecografía, 2D, 3D y 4D, siendo la realizada en dos dimensiones la mejor alternativa para el seguimiento del embarazo. Con ella se observa al feto en dos dimensiones, en un corte longitudinal y transversal sin sonido y en blanco y negro. Su principal ventaja es que permite diagnosticar malformaciones genitourinarias, gastrointestinales, en el sistema nervioso central, la cara y el cuello, la pared abdominal y el sistema cardiovascular. Además se pueden detectar hasta un noventa por ciento de las malformaciones fetales.

La ecografía 3D se construye a partir de cientos de fotogramas 2d, permite definir la profundidad de la exploración, examinar los huesos del bebé o solo la piel. Alrededor de la semana 32, es posible obtener una imagen detallada del rostro del niño, las uñas y el cabello. Esta técnica se implementada cada día más, con el fin de descartar patologías internas. La ecografía en 4D, es una sonograma 3D, en movimiento, es decir en tiempo real. No aporta mucho medicinalmente, y su costo es muy elevado.

### **3.1.5. EPIDEMIOLOGÍA**

La epidemiología es la disciplina científica que estudia la distribución, frecuencia, determinantes, relaciones, predicciones y control de los factores relacionados con la salud y enfermedad en poblaciones humanas. La epidemiología en sentido estricto, que podría denominarse humana, ocupa un lugar especial en la intersección entre las ciencias biomédicas y las ciencias sociales y aplica los métodos y principios de estas ciencias al estudio de la salud y la enfermedad en poblaciones humanas determinadas. Pero existe también una epidemiología

veterinaria y también podría hablarse de una epidemiología zoológica y botánica, íntimamente relacionada con la ecología.

La epidemiología se considera una ciencia básica de la medicina preventiva y una fuente de información para la formulación de políticas de salud pública. La epidemiología estudia, sobre todo, la relación causa-efecto entre exposición y enfermedad. Las enfermedades no se producen de forma aleatoria; tienen causas, muchas de ellas sociales, que pueden evitarse. Por tanto, muchas enfermedades podrían prevenirse si se conocieran sus causas. Los métodos epidemiológicos han sido cruciales para identificar numerosos factores etiológicos que, a su vez, han justificado la formulación de políticas sanitarias encaminadas a la prevención de enfermedades, lesiones y muertes prematuras.

#### **3.1.5.1. EPIDEMIOLOGÍA PERINATAL**

El enfoque de la perinatología, es ofrecer atención en salud para personas en gestación, embarazos y sus recién nacidos. La perinatología considera el proceso de 28 semanas de gestación a los primeros 28 días de vida como una línea continua.

La vigilancia epidemiológica perinatal cuenta con las siguientes herramientas:

- La historia perinatal básica
- El sistema de información perinatal
- La tarjeta perinatal (de la madre y del bebé)

Hay algunos conceptos básicos en epidemiología perinatal:

- Tasas para el estudio de problemas perinatales
- Tasas globales

- Tasas específicas
- Tasas de riesgos

### ***Estadísticas básicas en epidemiología perinatal***

Los datos básicos a tener en cuenta en epidemiología perinatal son:

1. Identificación del embarazo
2. Historia obstétrica
3. Datos del embarazo actual
4. Trabajo de parto y parto
5. Problemas maternos y neonatales
6. Alta materna y neonatal

Al trabajar los datos se tiene la capacidad de hacer un reporte epidemiológico y explorar las causas reales de mortalidad para reducirla. La confiabilidad de los indicadores epidemiológicos depende de si las Formas Clínicas Perinatales fueron llenadas completamente. Luego el primer paso es recomendar llenar las historias perinatales tanto como sea posible.

### ***3.2. MARCO TECNOLÓGICO***

Esta sección tiene como objetivo familiarizar tanto al usuario del sistema como al lector del presente libro sobre los diferentes conceptos técnicos involucrados en el desarrollo del proyecto, para que así esté mejor preparado para leer las posteriores secciones de este texto.

### **3.2.1. LOS SISTEMAS DE INFORMACIÓN**

FETHUS es, como su nombre lo indica, un sistema de información. Actualmente existen muchas definiciones las cuales pueden encontrarse en la literatura disponible, que es realmente abundante, una definición acertada es: un sistema de información es un software cuya función es administrar la información que es importante para el funcionamiento de una organización. ¿Pero qué significa administrar la información? En este contexto administrar información implica<sup>6</sup>:

- Compartir la información con cualquier miembro de la organización que la necesite y que esté autorizado para utilizarla.
- Reducir, en lo posible, el tiempo necesario para que un usuario tenga acceso a la información que necesita.
- Evitar en lo posible la redundancia en la información, es decir, que el costo de mantener la información almacenada sea lo menor posible.
- Mantener actualizada, en todo momento, la información sin importar que diferentes usuarios estén operando con ésta. El costo por mantener actualizada dicha información debe ser lo menor posible.
- Extender, en lo posible, la cobertura; es decir, que la información esté disponible al mayor número de usuarios ubicados en un área física mayor.
- Volver transparente al usuario las operaciones técnicas involucradas con la administración de la información.

Los sistemas de información nacieron de la necesidad de mantener organizada la información de la organización y de agilizar los procesos de ésta con el objetivo de ganar tiempo, y por ende dinero.

---

<sup>6</sup> Pressman, Roger, Ingeniería del software: Un Enfoque Práctico. 5 ed. Madrid: McGraw-Hill.

Debido a que los computadores y las tecnologías de la información y la comunicación están en continuo desarrollo, las exigencias a los sistemas de información han aumentado. Se hace necesario el acceso a la información, no solo desde los lugares de trabajo, sino desde los hogares y sitios de descanso. Los ejecutivos por ejemplo necesitan la información estando en pleno vuelo, los ingenieros civiles en el campamento de obras, los médicos donde se encuentren sus pacientes, etc.

### **3.2.1.1. LOS SISTEMAS DE INFORMACIÓN BASADOS EN WEB**

Antes de la popularización de Internet como medio de comunicación masivo, las empresas poseían sus sistemas de información implantados en redes locales en particular, con hardware y software específicos para cada una de ellas. Lo cual conllevaba a un inmenso margen de costo para hacer mantenimiento, soporte y mejoras a los sistemas de información, debido a que la organización no solo debía invertir dinero en mejorar el software del sistema, sino también en mejorar el hardware de la red en la que estaba implementado; labor que únicamente podía realizar personal calificado por la marca propietaria de la tecnología de la red, aumentando los costos en recursos humanos.

Pero con la masificación de Internet el panorama cambió radicalmente: las redes comenzaron a estandarizarse en redes para Internet y el alcance prácticamente global de Internet comenzó a brindar grandes oportunidades de ampliar la cobertura de los sistemas de información de una forma mucho más económica que antes: nacieron los sistemas basados en Web.

Gracias a la aceptación en masa de Internet los desarrolladores de sistemas de información y las compañías aprovecharon la gran oportunidad de expandir sus negocios y la cobertura de sus sistemas. Los sistemas de información que se ejecutan en Web son apetecidos ya que brindan la facilidad de publicar contenido

y ofrecer soporte a los clientes en todo el mundo los servicios de la organización.

Un sistema de información basado en Web es, básicamente, un sistema de información que utiliza la Web como medio de comunicación entre sus partes y los usuarios. Tiene la gran ventaja que un usuario necesita únicamente de una conexión a Internet y un navegador para ser cliente del sistema desde cualquier parte del mundo y las veinticuatro horas al día. De hecho, no es necesario que sea un computador personal el que haga la conexión: los teléfonos celulares, las agendas electrónicas, cualquier dispositivo con conexión a Internet puede, en teoría, conectarse con un sistema Web.

### **3.2.2. LA ARQUITECTURA DE UN SISTEMA**

Una vez que se ha examinado, de forma general, el funcionamiento básico de Internet, el lector está en capacidad de entrar un poco más “en materia”. Comenzamos por el concepto de “arquitectura de un sistema”. Llamaremos arquitectura a la vista conceptual que muestra los componentes que conforman un sistema y las relaciones entre ellos que hacen del sistema un todo (una entidad holística). La arquitectura es un modelo que sirve para ilustrar el diseño del sistema y puede ser vista bajo diferentes puntos de vista. Los empleados en este documento son:

- **Despliegue:** En esta vista se consideran los diferentes componentes *hardware* del sistema, las relaciones entre ellos y el papel que desempeñan dentro de la aplicación.
- **Implementación:** En esta vista se consideran los diferentes archivos y recursos *software* que componen el sistema, las relaciones entre ellos, su organización, y el papel que desempeñan dentro de la aplicación.
- **Procesos:** En esta vista se estudia la topología del sistema considerando

los diferentes procesos que se ejecutan como resultado de la interacción entre el usuario y la aplicación.

- **Diseño:** En esta vista se estudian los diferentes componentes del sistema desde el punto de vista del diseño del software que lo compone, es decir, como están organizadas las diferentes clases, funciones y variables que constituyen la funcionalidad del sistema y que están implementadas en los archivos y componentes descritos en la vista de implementación.

Ahora que el lector ha sido introducido al concepto de arquitectura de un sistema enunciaremos las arquitecturas más utilizadas desde el punto de vista del despliegue y del diseño:

### **3.2.2.1. LA ARQUITECTURA CLIENTE-SERVIDOR<sup>7</sup>**

La arquitectura cliente-servidor es la más tradicional de las arquitecturas empleadas en el diseño de sistemas de información, la causa es que evita de forma significativa la redundancia en la información, al estar organizada y almacena en un sistema llamado *Servidor*, el cual satisface las peticiones de los usuarios que utilizan otros sistemas llamados clientes y que se encargan de transmitir al servidor las peticiones y recibir de éste los servicios requeridos en tales peticiones. En este esquema un sistema es el servidor y muchos otros son los clientes. Los clientes se dividen en dos tipos principales:

- **Clientes delgados:** Si sólo se encargan de la presentación de los datos y no realizan labor alguna de procesamiento de la información.
- **Clientes gruesos:** Si se encargan de parte del procesamiento de la información.

---

<sup>7</sup> <http://es.wikipedia.org/wiki/Cliente-servidor>

Puede existir confusión al considerar los grandes servidores implementados en forma de *cluster*, en donde no hay una sola máquina que ejecuta el programa servidor sino que son varias. A pesar de ello, la arquitectura es la misma ya que un solo sistema es presentado al cliente como servidor, a pesar que tras bambalinas haya muchos sistemas realizando el trabajo.

Los sistemas en Web utilizan formas modificadas de esta arquitectura, en especial la de tres capas que será la estudiada a continuación.

### ➤ **ARQUITECTURA BASADA EN TRES CAPAS**

Antes de comenzar hablando de la arquitectura de tres capas, es necesario enunciar en qué consiste el diseño de aplicaciones por capas. El objetivo primordial consiste en separar la lógica del negocio de las otras labores del sistema como la presentación, el almacenamiento, el control de errores, etc. Este esquema brinda a los desarrolladores las siguientes ventajas:

- Cada capa es independiente de las otras, luego es posible cambiar la implementación de una capa sin afectar el comportamiento de las otras.
- En caso de algún cambio o corrección hay que buscar en la clase perteneciente a la capa afectada sin tener que revisar en un código mezclado, lo que implica probablemente, revisar en todo el código de la aplicación.
- Cada capa puede ser desarrollada, con un buen esquema de diseño, en paralelo disminuyendo el tiempo de desarrollo.
- Cada capa puede ser depurada independiente de las otras.
- Es posible incrementar el rendimiento de la aplicación ejecutando cada capa en hardware separado.

- No es necesario conocer en detalle la implementación de una capa para programar otra, tan solo basta con tener una buena documentación de la API de la capa.
- El sistema es fácilmente escalable, es decir, pueden desarrollarse nuevas capas que amplíen las funciones y prestaciones de la aplicación

La comunicación entre las capas se lleva a cabo por capas especiales llamadas interfaces que se encargan de llevar los datos de una capa a otra.

En esta arquitectura no se considera un solo sistema como servidor, sino tres sistemas generales que interactúan entre sí para conformar el servidor. Es como si en la arquitectura cliente-servidor “abriéramos” la caja negra del servidor para encontrar que hay tres cajas negras más pequeñas que lo componen. A estas cajas se les llama capas y son las siguientes:

- Capa de presentación: Esta capa se encarga de mostrar la salida de resultados al usuario final y de proveer una interfaz gráfica de usuario para recoger las entradas suministradas por éste, como los formularios.
- Capa de reglas del negocio: Esta capa contiene las reglas que regulan la administración de la información dentro del sistema. El diseño de esta capa es altamente influenciado por el análisis que se ha hecho antes del desarrollo del sistema.
- Capa de datos: En esta capa se gestiona el almacenamiento de la información en forma organizada.
- El lector debe tener en cuenta que hablamos de sistemas, no de máquinas hardware, por tanto es posible que los tres sistemas se ejecuten en la misma máquina física teniendo de esta forma la arquitectura de las tres capas, o es posible también que cada uno de estos sistemas se ejecute cada uno en un computador por separado.

- Existe otra forma de ver las capas, desde un punto de vista más afín con el diseño de la aplicación, que consiste en considerar cada capa como un conjunto de clases que realizan el trabajo de dicha capa y que la comunicación entre capas se hace instanciando clases de las capas de interfaz. Este esquema es explicado con mayor detalle en la sección sobre la Arquitectura del Sistema bajo el punto de vista del diseño.

### **3.2.2.2. ARQUITECTURA P2P<sup>8</sup>**

Aunque la arquitectura P2P no está involucrada con el desarrollo de FETHUS, la enunciaremos como contraparte del clásico esquema cliente-servidor. En la arquitectura P2P no hay cliente ni servidor, sino que todos son clientes y todos son servidores al tiempo, es decir, cada sistema conectado posee recursos que otros desean (en este caso su rol es servidor) y desea recursos que otros tienen (en este momento es cliente). Esta arquitectura es la utilizada por las redes en donde se comparten archivos para la descarga de todos los que están conectados a la red, como el caso del programa Emule y la red E-donkey.

### **3.2.3. EL HARDWARE**

En este punto se muestran las características del hardware empleado para el buen funcionamiento del sistema de información web.

#### **3.2.3.1. EL ECÓGRAFO**

El dispositivo utilizado para realizar ultrasonidos es el ecógrafo. Consta de un

---

<sup>8</sup> <http://es.wikipedia.org/wiki/P2p>

monitor, una CPU, y un transductor, cada uno de sus componentes tiene una función específica. El transductor envía las ondas ultrasónicas las cuales una vez reflejadas son recibidas por el mismo aparato, el cual traspasa la información a la CPU, el cual tiene la misión de decodificar (mediante un procesador digital de señales DSP) la información, creando una imagen que es transferida al monitor, donde gracias a la información se puede ver una imagen.

El ecógrafo utiliza un software conformado por un sistema operativo, una base de datos, una interfaz de red y un servidor de impresión. Esto le permite gestionar información de la unidad hospitalaria donde se realiza el procedimiento, seguimiento de las sesiones, elaboración de curvas de crecimiento e histogramas. Permite además tomar fotografías, y sesiones de audio y video.

### **3.2.3.2. EL SERVIDOR**

El servidor en el cual se va a alojar FETHUS, debe contar con las siguientes características en cuanto a software se refiere:

- Servidor web Apache/Tomcat, que es software libre y de distribución gratuita, se puede ejecutar tanto en Windows como en GNU/Linux con la ventaja de que es muy fácil de configurar, el uso de este servidor web se hace necesario ya que el lenguaje escogido para el desarrollo ha sido JSP.
- Sistema gestor de base de datos MySQL, está licenciado bajo la GNU GPL para cualquier uso compatible con esta licencia. El API de Java el cual es empleado para desarrollar el presente proyecto permite acceder a las bases de datos MySQL por medio del driver específico.
- Java Virtual Machine, puesto que se requiere interpretar y ejecutar las instrucciones expresadas en Java bytecode, el cual es generado por el compilador del lenguaje de programación Java.

- GNU/Linux o Windows como sistema operativo, aunque se recomienda el primero, ya que cuenta con ciertas ventajas como lo son la seguridad y la fácil configuración del software a nivel de servidores.

### **3.2.4. EL SOFTWARE**

En este punto es hora de profundizar en el software que hace posible el desarrollo y funcionamiento de un sistema de información en Web. En esta sección se asume que la arquitectura de dicho sistema de información es la arquitectura de las tres capas.

#### **3.2.4.1. LAS LICENCIAS DE SOFTWARE**

Sin embargo antes de hablar de software hay que entrar en detalle sobre un punto crucial a la hora de seleccionar el software en donde habrá de desarrollarse y en donde habrá de funcionar cualquier sistema: la licencia de uso. Una licencia de uso no es más que un acuerdo legal entre el usuario y el propietario o autor de un software que define el uso legal de éste.

##### **➤ LICENCIAS DE CÓDIGO CERRADO**

Una licencia de código cerrado es aquella donde quien adquiere la licencia no puede ver, modificar o redistribuir el código fuente de la aplicación. Este tipo de licencia es la más común en el software comercial. Las licencias de código cerrado más comunes son: Freeware, Shareware, Licencia de prueba (Trial), Licencia de demostración, Adware y Licencia comercial, entre otras.

##### **➤ LICENCIAS DE CÓDIGO ABIERTO**

Como contraparte a las licencias de código cerrado en donde el código fuente no está disponible para el usuario, y por tanto no se pueden hacer modificaciones al mismo, las licencias de código abierto distribuyen el código fuente del producto promoviendo que desarrolladores independientes le hagan mejoras sin pedir autorización alguna siempre y cuando respeten los términos de la licencia. Como FETHUS ha sido desarrollado usando en su mayor parte software de código abierto, en esta sección se profundizará sobre las licencias más comunes en esta modalidad.

### ✓ LICENCIA GPL

El software libre es el software que una vez adquirido puede ser modificado, estudiado, copiado, usado y redistribuido libremente. Aunque en inglés los términos libre y gratis son generalizados en una sola palabra (*free*), el software libre *no* implica que sea gratuito, lo que implica es que el que adquiera una copia del software puede distribuirlo como desee, ya sea cobrando, alquilando o regalando copias del software adquirido.

Hay que hacer diferencia entre el software libre y el software de dominio público, éste último significa que el software hace parte del conocimiento científico de la humanidad, luego no se debe seguir ninguna licencia para su uso por débil que sea, siempre y cuando se respete el crédito y se use el producto con fines legales.

En los inicios de los años ochenta, Richard Stallman<sup>9</sup> comenzó su ambicioso proyecto de un sistema operativo libre (como contraparte al creciente DOS; al robusto UNIX y al notable Macintosh), el proyecto se llamó GNU que significa: GNU No es Unix. El proyecto GNU decidió crear una licencia que reglamentara el

---

<sup>9</sup> Fundador del Movimiento del Software Libre, de la Free Software Foundation y del proyecto GNU.

uso del software libre y que garantizara ciertas libertades<sup>10</sup>.

La licencia GNU/GPL impone ciertas restricciones respecto al uso del programa dentro de otro desarrollo, es decir, si un programa B tiene código proveniente de un programa A, el programa B debe ser licenciado GNU/GPL de forma obligatoria si el programa A es GNU/GPL. Esto ha dividido las licencias de código abierto en dos: las que son compatibles con GNU/GPL y las que no lo son.

### ✓ LICENCIA LGPL

La licencia pública LGPL es como la licencia GNU/GPL pero excluye del alcance de la licencia al software que para su construcción ha necesitado la inclusión de código licenciado LGPL (En la GPL, si un producto usa código licenciado GPL, el producto resultante debe ser también GPL). Particularmente, la licencia LGPL hace hincapié en los términos “trabajo derivativo” y “trabajo que usa la librería” diciendo, en el inciso quinto<sup>11</sup>, que un trabajo que simplemente use la librería o código licenciado LGPL está fuera del alcance de la licencia y por lo tanto no se aplica la obligatoriedad de licenciar el producto desarrollado con LGPL.

### ✓ LICENCIA BSD

La licencia BSD (*Berkeley Software Distribution*) es una licencia de código abierto que tiene menos restricciones que la GNU/GPL y que está muy cerca de volverse una “licencia” de dominio público. La licencia BSD exige que:

- Las distribuciones de código fuente deben retener el texto de la licencia.

---

<sup>10</sup> <http://www.gnu.org/philosophy/free-sw.es.html>

<sup>11</sup> <http://www.gnu.org/licenses/lgpl-3.0.html>

- Las distribuciones de archivos binarios deben incluir esta licencia en la documentación entregada con los mismos.
- Ni los nombres de los autores, ni los de los colaboradores, se pueden incluir para promocionar productos derivados del software sin su permiso.

#### ✓ LICENCIA APACHE

La licencia Apache, es una licencia de software libre creada por la *Apache Software Foundation* (ASF). La licencia Apache (con versiones 1.0, 1.1 y 2.0) requiere la conservación del aviso de copyright y el disclaimer<sup>12</sup>, pero no es una licencia copyleft<sup>13</sup>, ya que no requiere la redistribución del código fuente cuando se distribuyen versiones modificadas.

Todo el software producido por la ASF o cualquiera de sus proyectos está desarrollado bajo los términos de esta licencia. Además algunos proyectos que no pertenece a la ASF también siguen la licencia Apache: en enero de 2007, más de 1000 proyectos no pertenecientes a la ASF en SourceForge<sup>14</sup> estaban disponibles bajo los términos de la Licencia Apache.

#### **3.2.4.2. LENGUAJES DEL LADO DEL SERVIDOR**

El HTML, a pesar de su gran éxito, está muy lejos de ser un lenguaje de programación completo y es insuficiente para crear sitios Web complejos y mucho menos sistemas de información. De hecho, cuando un sitio Web atraviesa las

---

<sup>12</sup> Descargo de responsabilidad, declaración que delimita el alcance de los derechos y obligaciones y aplicadas por las partes en una relación legalmente reconocida.

<sup>13</sup> Forma de licencia empleado para modificar el derecho de autor de obras o trabajos, tales como software, documentos, música y obras de arte.

<sup>14</sup> Software de colaboración para la administración de desarrollos.

barreras de la mera presentación de contenidos e incursiona en la administración de datos y la implementación de la lógica de un negocio ya no puede ser construido usando HTML ni lenguajes que operan en la capa de presentación como JavaScript o los CSS. Por lo tanto las labores de procesamiento de la lógica del negocio y el acceso de la base de datos deben ser implementados por otras tecnologías: los lenguajes del lado del servidor.

Un lenguaje del lado del servidor es un lenguaje de programación que sirve para crear programas que se ejecutan en el servidor, pero cuyas entradas y salidas provienen y van hacia el cliente, de ahí el nombre de “del lado del servidor”.

Entre estos lenguajes hay dos grandes categorías: los lenguajes que generan código ejecutable y los que se ejecutan “al vuelo” llamados lenguajes “script” o de guión.

Los lenguajes que generan código ejecutable son aquellos en donde la compilación y la ejecución se hacen en etapas bien separadas en el intervalo temporal. El proceso de compilación se efectúa una vez por versión de la aplicación tras lo cual sólo ocurre la ejecución. El código ejecutable es muy rápido y permite realizar operaciones más rápidamente que los lenguajes de guión. Tecnologías de este tipo son el PERL y entornos de programación como Delphi, Java (aunque una aplicación Java no posee código compilado real en el sentido que se ejecute directamente sobre el sistema operativo), Visual C++, etc.

Sin embargo en la programación Web se han popularizado los otros lenguajes del lado del servidor, es decir, los lenguajes de guión. Pese a su tiempo de ejecución mayor, pues las operaciones de compilación y ejecución son casi simultáneas, los lenguajes de guión presentan las siguientes ventajas:

- Se pueden embeber dentro de código HTML de forma transparente para el cliente.
- No necesitan compilarse, luego si se hace un cambio en el código, el

cambio queda funcional inmediatamente.

- Como no hay compilación, el código siempre está disponible, lo cual es muy apreciado por desarrolladores de código libre. (Esto hace que haya muchas herramientas y tecnología disponible al desarrollador de aplicaciones Web).
- Sin embargo presentan una gran desventaja que se deriva de la primera ventaja listada: como se puede embeber en el código HTML (un componente de presentación) es frecuente que se revuelva el código de las capas de presentación, reglas del negocio, y datos.

Las tecnologías más comunes en esta categoría son:

- PHP
- ASP
- JSP

### **3.2.4.3. JAVA**

Hasta la fecha, la plataforma Java ha atraído a más de 6,5 millones de desarrolladores de software. Se utiliza en los principales sectores de la industria de todo el mundo y está presente en un gran número de dispositivos, equipos y redes.

La versatilidad y eficiencia de la tecnología Java, la portabilidad de su plataforma y la seguridad que aporta, la han convertido en la tecnología ideal para su aplicación a redes. De portátiles a centros de datos, de consolas de juegos a súper equipos científicos, de teléfonos móviles a Internet, Java está en todas partes.

Java es un lenguaje de alto nivel, es orientado a objetos, independiente de la plataforma, con un alto nivel de seguridad, capacidad multi hilo, permite la

creación de aplicaciones distribuidas, además, su robustez, y su integración con el protocolo TCP/IP lo hacen un lenguaje ideal para aplicaciones web.

Una de las características más importantes que hicieron del lenguaje algo muy innovador y acogido es la de permitir ejecutar las aplicaciones en diferentes plataformas existentes en el mercado.

Las aplicaciones Java están compiladas en bytecode que son interpretadas por la máquina virtual o JRE, gracias a esto permite la portabilidad en ejecución. Es una ventaja significativa para los desarrolladores de software, pues antes tenían que hacer un programa para cada sistema operativo, por ejemplo Windows, GNU/Linux, Apple, etc. Esto lo consigue porque se ha creado una máquina de Java para cada sistema que hace de puente entre el sistema operativo y el programa de Java y posibilita que este último se entienda perfectamente.

En agosto de 2007, Sun microsystems, la empresa dueña de los derechos comerciales de la tecnología Java, liberó la gran mayoría de las especificaciones de Java bajo licencia GNU-GPL, dejando atrás los prejuicios para no escoger esta plataforma como herramienta de desarrollo. Importantes competidores como la tecnología .NET de microsoft han intentado sin éxito, desplazar a Java como líder del mercado de lenguajes de programación.

#### **3.2.4.4. LA API**

Una API (*Application Program Interface*) o Interfaz de Programación de Aplicación, es un conjunto de clases, métodos y funciones que están listos para ser usados por otro programa. Este “programa” puede ser una capa de la aplicación u otra aplicación diferente.

El propósito de una API es maximizar la reutilización del código, de tal forma que si una tarea específica se repite una y otra vez no haya que programar el mismo

código una y otra vez, sino simplemente llamar las funciones de la API que ya poseen dicho código. Otra ventaja de una API es la separación del código, en forma de capas, lo cual facilita el mantenimiento y depuración del software.

### 3.2.4.5. LOS FRAMEWORK

Los frameworks son estructuras definidas para soportar y facilitar el desarrollo de proyectos de software. Incluyen el soporte de programas, bibliotecas y un lenguaje interpretado, entre otros. La ventaja de la utilización de los frameworks es que se puede pasar más tiempo identificando requisitos de software que estar lidiando con los tediosos detalles de bajo nivel para proveer un sistema funcional.

Para garantizar la calidad de un proyecto de programación web es necesario definir un patrón de desarrollo que establezca una independencia clara entre los diferentes componentes que lo constituyen. Los frameworks actuales en su mayoría soportan el patrón Modelo vista controlador, MVC, éste es un patrón de arquitectura de software, que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

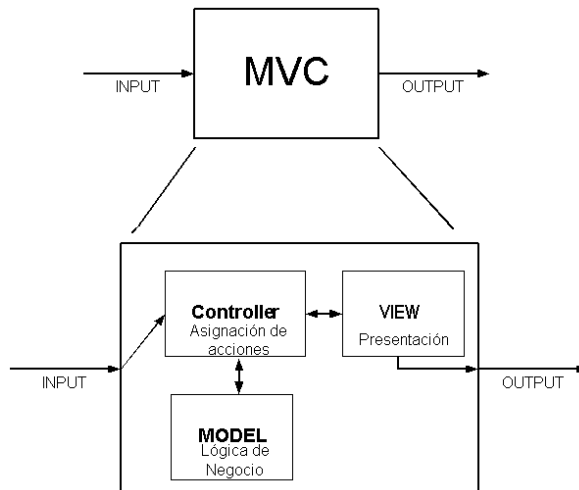


Figura 3. Patrón Modelo Vista Controlador.

El patrón MVC es frecuente en aplicaciones web, donde la vista es la página HTML, y el código que provee de datos dinámicos a la página. El modelo es el sistema de gestión de base de datos y la lógica del negocio; y el controlador es el responsable de recibir los eventos de entrada a partir de la vista o interfaz de usuario.

## **4. COMPONENTES TECNOLÓGICOS UTILIZADOS EN LA ELABORACIÓN DEL SISTEMA DE INFORMACIÓN**

Esta sección tiene por objetivo describir la tecnología utilizada para construir FETHUS, no solo la utilizada para el desarrollo del software como tal, sino las diferentes herramientas como lenguajes de programación, servidores Web usados en la pruebas de la fase de desarrollo, etc. Además los autores explican de forma breve las razones que tuvieron para la selección de la tecnología.

Por último los autores aclaran que ninguna de estas tecnologías es distribuida junto con FETHUS y que no necesariamente son indispensables para el funcionamiento del sistema.

### **4.1. HERRAMIENTAS DE LA FASE DE DISEÑO**

#### **4.1.1. ARGOUML<sup>15</sup>**

ArgoUML ha sido concebida como una herramienta para el diseño y análisis de software orientado a objetos, incluye una serie de características que satisfacen las necesidades de arquitectos y diseñadores de proyectos orientados a objetos, además soporta estándares abiertos.

Esta herramienta es una aplicación 100% Java. Esto permite que ArgoUML funcione en todas las plataformas para las que Java 2 esté disponible. ArgoUML se encuentra licenciado bajo la BSD License, esto significa que si una empresa desea vender ArgoUML o algún trabajo derivado es libre de hacerlo bajo ciertas condiciones de copyright.

---

<sup>15</sup> <http://argouml.tigris.org/>

En vista de la necesidad de realizar una excelente documentación UML, los autores del presente proyecto han escogido esta gran herramienta ya que ofrece gran variedad en cuanto a los tipos de diagramas UML que se pueden implementar para la fase de análisis de cualquier proyecto de software.

#### **4.1.2. MYSQL WORKBENCH**

Esta es una herramienta visual para el diseño de bases de datos que integra diseño, modelado, creación y mantenimiento de bases de datos en un único entorno. MySQL Workbench<sup>16</sup> se encuentra disponible bajo dos tipos de versiones, edición comunitaria, que es licenciada bajo los términos de la GPL, y la edición estándar, que requiere la compra de una suscripción anual y que además incluye módulos y plugins<sup>17</sup> extras.

El diseño y la implementación de la base de datos del presente proyecto han sido elaborados con la edición comunitaria de MySQL Workbench, puesto que es lo suficientemente robusta y estable para suplir las necesidades actuales.

### **4.2. TECNOLOGÍAS Y HERRAMIENTAS DE LA FASE DE DESARROLLO**

#### **4.2.1. APACHE STRUTS**

Apache Struts<sup>18</sup> es un framework open-source para la creación de aplicaciones web Java. Las aplicaciones web difieren de los sitios web convencionales en que las aplicaciones web pueden crear una respuesta dinámica. La mayoría de sitios web ofrecen solo páginas estáticas. Una aplicación web puede interactuar con

---

<sup>16</sup> <http://wb.mysql.com/>

<sup>17</sup> Complemento, es una aplicación que aporta una función nueva a determinada herramienta.

<sup>18</sup> <http://struts.apache.org/>

motores de bases de datos y lógica del negocio para así personalizar una respuesta.

Las aplicaciones web basadas en JavaServer Pages en ocasiones mezclan código de consultas a bases de datos, código de diseño de la página, y el código para el control de flujo. En la práctica, si estos tipos de código se encuentran mezclados, las aplicaciones más grandes llegan a ser difíciles de mantener.

El framework Struts es diseñado para ayudar a los desarrolladores a crear aplicaciones web que utilicen la arquitectura Modelo-Vista-Controlador. Debido al deseo de los autores, de desarrollar el actual proyecto enfocado a la web y con tecnología Java, se ha optado por el uso de Struts como herramienta gracias a su robustez y confiabilidad, además que permite reducir el tiempo de desarrollo.

El código fuente y la documentación de Struts están licenciados por la Apache Software Foundation (ASF) bajo la Licencia Apache, versión 2.0, disponible hasta la fecha de escritura de este documento en <http://www.apache.org/licenses/LICENSE-2.0>.

#### **4.2.2. HIBERNATE**

Hibernate<sup>19</sup> es una herramienta de Mapeo objeto-relacional para la plataforma Java que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos XML que permiten establecer estas relaciones.

Esta herramienta posee un lenguaje de consulta llamado HQL<sup>20</sup>, al mismo tiempo que una API para construir consultas programáticamente, conocida como Criteria.

---

<sup>19</sup> <https://www.hibernate.org/>

<sup>20</sup> Hibernate Query Language.

Los autores hemos decidido optar por el uso de Hibernate gracias a que genera las sentencias SQL y libera a los desarrolladores del manejo manual de los datos que resultan de la ejecución de dichas sentencias, manteniendo la portabilidad entre todos los motores de bases de datos con un ligero incremento en el tiempo de ejecución.

Hibernate es software libre, distribuido bajo los términos de la licencia GNU LGPL que permite el uso de esta herramienta en proyectos open-source o proyectos comerciales.

#### **4.2.2.1. MAPEO OBJETO-RELACIONAL**

Como se ha dicho anteriormente Hibernate es una herramienta que basa su funcionamiento en la técnica de mapeo objeto-relacional, permitiendo trabajar con los datos persistidos en bases de datos relacionales como si fueran parte de una base de datos orientada a objetos, para así poder transformar un registro en objeto y viceversa.

Actualmente el problema radica en que ambos modelos (el relacional y el orientado a objetos), están basados en diferentes objetivos, es decir, buscan distintos fines. El fin de la orientación a datos, es la optimización de la integridad de los datos. Por otra parte la orientación a objetos busca la modularización natural del dominio del problema, que se basa en el comportamiento, de tal forma que cada objeto tenga una responsabilidad en el sistema, y que sea posible que colabore con otros objetos.

¿Pero en que se basa y como se comporta internamente el proceso de mapeo objeto-relacional? Con el mapeo objeto-relacional declaramos en que tablas y columnas de la base de datos relacional se guarda la información de la entidad. Esto se logra haciendo mapeos a las entidades, las propiedades, las referencias, y las colecciones, para después proceder a realizar las conversiones de tipo.

### **4.2.3. MYSQL**

MySQL<sup>21</sup> es un sistema que se utiliza para la gestión de base de datos relacional, multihilo y multiusuario que cuenta con un esquema de licenciamiento dual. Se desarrolla como software libre y se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero aquellas empresas que quieran hacer uso de esta herramienta para incorporarlo en productos privativos deben adquirir una licencia específica que les permita este uso.

Entre las tantas razones por las cuales los autores del presente proyecto eligieron la implementación de la base de datos con MySQL se encuentran: La fácil administración de las bases de datos gracias a las diferentes herramientas existentes en el mercado que brindan esta posibilidad de manera fácil y gráfica, la gran cantidad de controladores y API's que hay para interactuar con MySQL desde cualquier lenguaje de programación, la extensa documentación tanto en inglés como en español y las diferentes plataformas soportadas para su funcionamiento.

### **4.2.4. CKEDITOR**

CKEditor<sup>22</sup> es un editor de texto que puede ser usado en páginas web, es del tipo "What You See Is What You Get" (WYSIWYG), es decir, lo que ves es lo que obtienes y permite escribir un documento viendo directamente el resultado final. Se hace necesaria la utilización de ésta tecnología puesto que ciertos usuarios del sistema tienen la tarea de insertar grandes cantidades de texto con formato para introducir en algunos campos de la base de datos.

---

<sup>21</sup> <http://www.mysql.com/>

<sup>22</sup> <http://ckeditor.com/>

El núcleo principal de este editor de texto está escrito en JavaScript contando con la ventaja que funciona con diferentes interfaces tales como ASP, ASP.NET, PHP, Perl, Python, Java entre otros. CKEditor es compatible con la mayoría de navegadores de internet, entre los principales se encuentran Internet Explorer 5.5+ (Windows), Firefox 1.5+, Safari 3.0+, Google Chrome (Windows), Opera 9.50.

CKEditor es distribuido bajo las licencias de código abierto GPL, LGPL y MPL. Este triple modelo de licencia copyleft evita la incompatibilidad con otras licencias de código abierto.

#### **4.2.5. JASPERREPORTS**

Debido a la necesidad de generar informes y reportes dinámicos solicitados por los usuarios de FETHUS y también a que el desarrollo del presente proyecto se ha hecho con tecnología Java, JasperReports<sup>23</sup> es la herramienta ideal para suplir estas necesidades, puesto que cuenta con la habilidad de entregar contenido enriquecido ya sea en el monitor, la impresora o ficheros PDF, HTML, XLS, XML, entre otros.

JasperReports cuenta con un gran número de descargas desde que salió al mercado en el año 2001, es desarrollado por Jaspersoft y puede ser encontrado bajo dos tipos de licencias: la comercial, la cual brinda soporte personalizado, servicios profesionales y capacitación directamente desde Jaspersoft, y la licencia open-source que cuenta con el respaldo de la gran y activa comunidad que apuesta por la libertad del software.

---

<sup>23</sup> <http://jasperforge.org/>

#### **4.2.6. HTML**

HTML es el lenguaje de marcado estándar diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web, sus siglas significan “HyperText Markup Language”.

Entre las características más importantes de HTML se encuentran el manejo de formularios, el elemento FIG para imágenes, notas marginales y de pie de página, identificadores de fragmento, entre otras.

HTML es una aplicación SGML conforme al estándar ISO-8879. XHTML es una adaptación del HTML basada en el lenguaje XML. La masificación de Internet ha ocasionado que el HTML se haya convertido en uno de los formatos más populares para la elaboración de documentos y es uno de los más fáciles de aprender.

#### **4.2.7. JSP**

JSP es una tecnología Java que permite generar contenido dinámico web, y permite la utilización de código Java mediante scripts, además cuenta con sus propias acciones predefinidas mediante etiquetas. Estas etiquetas pueden ser enriquecidas mediante la utilización de Bibliotecas de etiquetas también llamadas TagLibs o TagLibraries externas e incluso personalizadas.

Generalmente el funcionamiento de la tecnología JSP es de la siguiente manera: el Servidor de Aplicaciones interpreta el código contenido en la página JSP para construir el código Java del servlet a generar. Este servlet será el que genere el documento (típicamente HTML) que se presentará en la pantalla del Navegador del usuario.

El rendimiento de una página JSP es el mismo que tendría el servidor equivalente, ya que el código es compilado como cualquier otra clase Java. A su vez, la

máquina virtual compilará dinámicamente a código de máquina las partes de la aplicación que lo requieran. Esto hace que JSP tenga un buen desempeño y sea más eficiente que otras tecnologías web que ejecutan el código de una manera puramente interpretada.

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

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

#### **4.2.8. ECLIPSE<sup>24</sup>**

Es un entorno de desarrollo integrado básico, multi-lenguaje y cuenta con un sistema de plug-ins que amplía el rango de cubrimiento sobre sus funcionalidades. Ha sido diseñado especialmente para el desarrollo de "Aplicaciones de Cliente Enriquecido", algo opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores.

Esta plataforma ha sido escrita principalmente en Java y por medio de los diferentes plug-ins extiende su soporte para el desarrollo de aplicaciones en otros lenguajes tales como C, C++, Python, PHP, entre otros. Actualmente se encuentra en continuo desarrollo, y dispone de paquetes de idioma que proporcionan traducciones a más de una docena de diferentes lenguajes naturales.

---

<sup>24</sup> <http://www.eclipse.org/>

Eclipse cuenta con una extensión que aporta valiosos añadidos llamada Myeclipse, que facilita la creación de arquitecturas multinivel J2EE. Los autores del presente proyecto han decidido la utilización de esta herramienta porque mejora la productividad de los desarrolladores mediante la simplificación del ciclo de vida del desarrollo, debido a que se integra con gran variedad de frameworks entre los cuales se encuentran Struts e Hibernate.

Eclipse fue liberado originalmente bajo la Common Public License, pero después fue re-licenciado bajo la Eclipse Public License. La Free Software Foundation ha dicho que ambas licencias son licencias de software libre, pero son incompatibles con Licencia Pública General de GNU (GNU GPL). Por otra parte Myeclipse se encuentra disponible bajo licencia comercial.

## **5. DESCRIPCIÓN DE LOS MÓDULOS DEL SISTEMA**

En esta sección los autores describen el proyecto de grado del cual es objeto este libro para que el lector tenga una visión global de las funciones y módulos implantados en el mismo con base en la Especificación de Requisitos de Software contenida en este documento. En esta sección se muestran los módulos y características que posee FETHUS:

**Descripción:** En este ítem se brinda una breve descripción textual del módulo.

**Requisitos asociados:** Aquí se identifican los requisitos consignados en la especificación de requisitos de software, correspondientes a la fase de análisis del módulo en cuestión. Este documento hace parte de la sección de anexos, exactamente el anexo A.

**Funciones:** En esta subsección se listan las funciones que puede realizar el usuario y que se relacionan con el módulo descrito.

### **5.1. MÓDULO DE GESTIÓN DE USUARIOS**

#### **5.1.1. DESCRIPCIÓN**

Este módulo es aquel que se encarga de administrar, desde la autenticación de los usuarios en el sistema hasta la información personal referente a éstos, ya sea root, residente o estudiante. El módulo de gestión de usuarios es de vital importancia puesto que se encarga de asignar privilegios y además de la seguridad del sistema, porque no cualquier persona puede acceder a determinada información de la UMMFN, manteniendo así la integridad y privacidad de los datos.

### 5.1.2. REQUISITOS ASOCIADOS

Los requisitos asociados, consignados en la Especificación de Requisitos de Software, corresponden a los requisitos funcionales desde el RF-01 (Crear cuentas de usuario) hasta el RF-06 (Modificar datos personales).

### 5.1.3. FUNCIONES

La gestión de usuarios es una labor que puede ser llevada a cabo única y exclusivamente por el superadministrador, quien lleva a cabo las funciones relacionadas con su rol dentro de la organización. El presente módulo dispone de un menú estéticamente agradable y fácil de usar, como se muestra en la Figura 4, para la creación, modificación y eliminación de los usuarios del sistema.



Figura 4. Captura del menú para la gestión de usuarios del sistema.

Al solicitar el registro de un nuevo usuario del sistema, se muestra un formulario en el cual se solicita la información personal referente a dicho usuario, como lo es, Documento de Identificación, Nombres, Apellidos, Correo, entre otros como se muestra en la siguiente Figura. Estos datos son almacenados en la base de datos, permitiendo tener un mayor control sobre el sistema.

The screenshot shows a web browser window with the title 'Fethus Web: Sistema De Informacion...'. The page header features the 'Fethus Web' logo and the text 'unidad de medicina Materno Fetal'. Navigation buttons for 'usuarios', 'pacientes', 'casos', 'patologias', and 'ayuda' are visible. The main content area is titled 'Gestión de Usuarios' and contains a registration form with the following fields and values:

- Documento: 2022098
- Nombres: Raúl Fernando
- Apellidos: Rivero Olarte
- Correo: raulinux899@gmail.com
- Permisos: SUPERADMINISTRADOR (dropdown menu)
- Login: [empty]
- Contraseña: [masked with dots]

Buttons for 'Actualizar' and 'Cancelar' are located below the form. The footer contains the text: '©2009 Hospital Universitario de Santander Unidad Medicina Materno Fetal • FethusWeb 1.0 • Proporcionado por: Grupo GIB-UIS.'

Figura 5. Captura de registro de un nuevo usuario o modificación de uno existente.

Implícitamente este módulo se encarga de la asignación de permisos y la autenticación del usuario ante el sistema, habilitando y deshabilitando las funcionalidades que pueden o no ser accedidas por determinados tipos de usuarios, manejando adecuadamente el control de acceso a la aplicación y autorización.

## **5.2. MÓDULO DE CASOS DE LA UMMFN**

### **5.2.1. DESCRIPCIÓN**

Tal y como se dice en la sección llamada “*Descripción de la situación actual*” (véase la sección 2.2), una de las funciones más importantes desempeñadas por el personal de la UMMFN, es la de la realización de tomas ecográficas a los pacientes de la Unidad. Siendo FETHUS el sistema de información de la UMMFN, no podía obviar un punto tan importante como el de brindar mecanismos para facilitar la organización de la información obtenida en las sesiones ecográficas.

### **5.2.2. REQUISITOS ASOCIADOS**

Los requisitos asociados, consignados en la Especificación de Requisitos de Software, corresponden a los requisitos funcionales desde el RF-07 (Registro de casos de la UMMFN) hasta el RF-15 (Registrar conclusiones del caso de la UMMFN).

### **5.2.3. FUNCIONES**

El menú de casos de la UMMFN puede ser accedido únicamente por los usuarios superadministrador y residente, este módulo cuenta con seis secciones que se muestran en la Figura 6. Cuando el médico a cargo sospecha o tiene la certeza de una anomalía fetal con base en las ecografías practicadas, ingresa a la opción “crear caso”, en donde debe ingresar los datos y antecedentes del paciente.

Si el caso ya se encuentra registrado en el sistema, el superadministrador o el residente pueden registrar alguna de las sesiones ecográficas, ya sea la de primer, segundo o tercer trimestre, asociándola al paciente en cuestión. Una vez se da por terminado el seguimiento al caso en cuestión, se procede a dar de alta a dicho caso registrando las conclusiones de acierto en el pronóstico realizado al paciente.

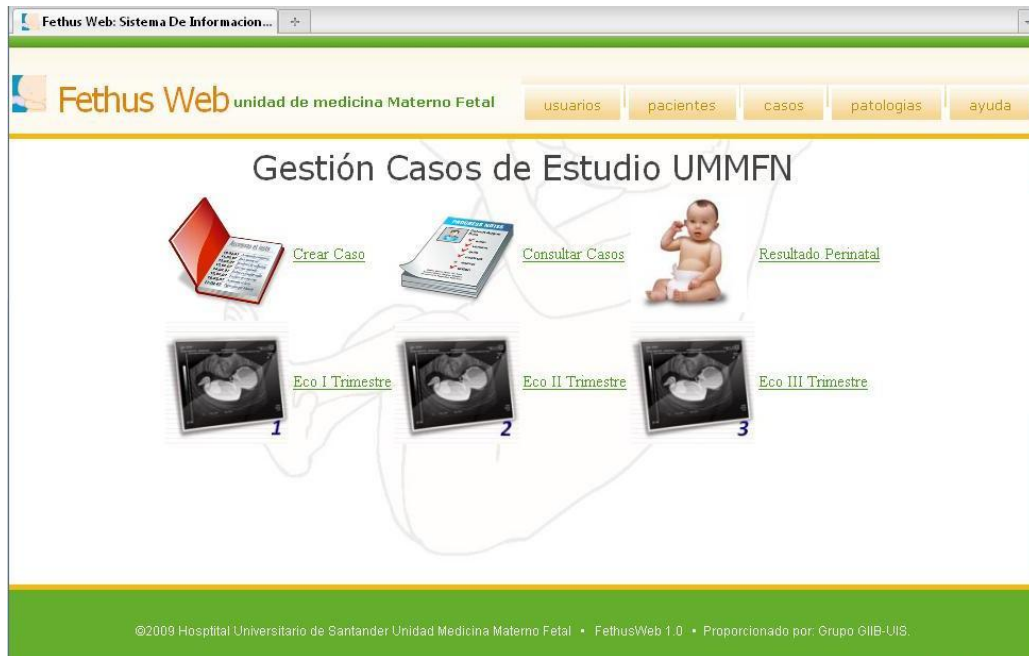


Figura 6. Captura del módulo de administración de casos UMMFN.

Como se mencionó anteriormente, los casos también se encuentran ligados a los pacientes, por lo cual la gestión de pacientes hace parte del presente módulo, esto con el fin de llevar un control más detallado acerca del caso.

Figura 7. Captura de actualización o registro de un paciente nuevo.

### **5.3. MÓDULO DE GESTIÓN DE PATOLOGÍAS GENERALES**

#### **5.3.1. DESCRIPCIÓN**

El uso de este módulo es de uso único y exclusivo del usuario root y residente. Permite la creación y modificación de registros de patologías generales. La opción de eliminación no ha sido tomada en cuenta porque según el jefe de la UMMFN, estas patologías han sido, son y serán siempre las mismas, sólo en muy pocos casos varía con el tiempo y si esto llega a suceder, simplemente se modifica la patología en cuestión.

#### **5.3.2. REQUISITOS ASOCIADOS**

Los requisitos asociados, consignados en la Especificación de Requisitos de Software, corresponden a los requisitos funcionales RF-16 (Registrar patologías generales) y RF-17 (Modificar patologías generales).

#### **5.3.3. FUNCIONES**

Como ya se explicó en la descripción del presente módulo, las patologías generales están estandarizadas a nivel mundial con las enfermedades ya conocidas, es por esto que la única función de este módulo es la de registrar una nueva patología general como se muestra en la Figura 8, solicitando al residente o al superadministrador la descripción de dicha patología junto con las imágenes asociadas a ésta. La consulta de patologías generales forma parte del módulo de consultas que es explicado en la siguiente sección y puede ser accedido también por el rol estudiante.

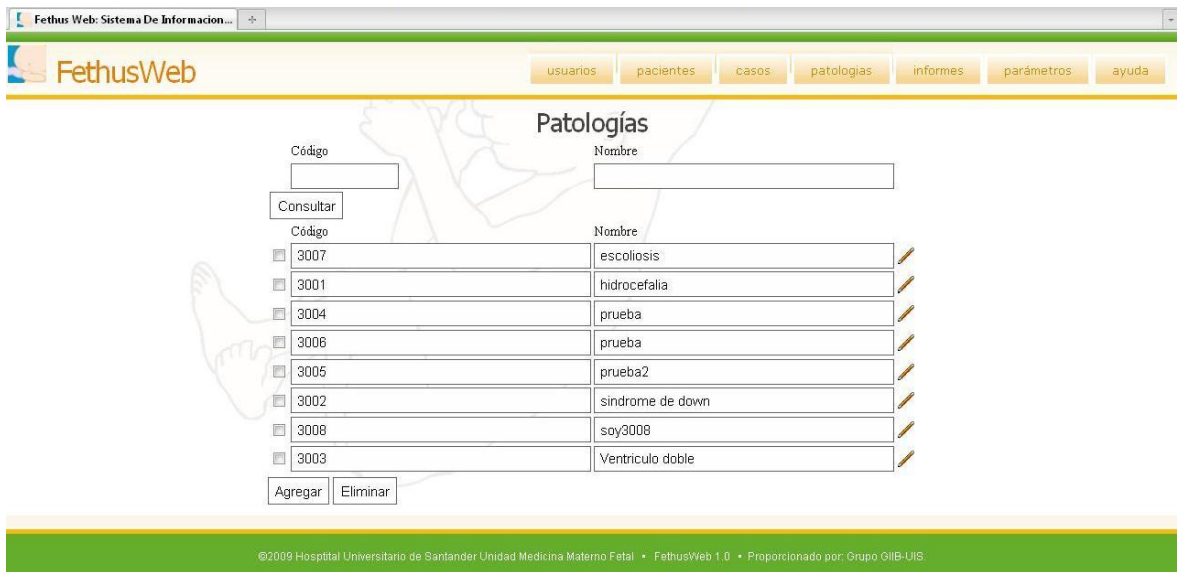


Figura 8. Captura del módulo de patologías generales.

## 5.4. MÓDULO DE CONSULTAS

### 5.4.1. DESCRIPCIÓN

Como su nombre lo indica, el módulo de consultas es el encargado de proveer al usuario la información concerniente a epidemiologías y enfermedades relacionadas con casos y patologías generales que se encuentran almacenados en la base de datos de FETHUS y de proveer mecanismos para la consulta de dicha información mediante ciertos parámetros de acotación. Además, si el usuario cuenta con los permisos requeridos (root), este módulo brinda la función de consultar información referente a otros usuarios que se encuentren activos para el uso de la herramienta.

### 5.4.2. REQUISITOS ASOCIADOS

Los requisitos asociados, consignados en la Especificación de Requisitos de Software, corresponden a los requisitos funcionales desde el RF-18 (Consultar por casos de pacientes) hasta el RF-34 (Ver detalles de usuario).

### 5.4.3. FUNCIONES

Esta es la sección más extensa dentro de FETHUS, ya que se encuentra profundamente ligado a los diferentes módulos del sistema mediante sus respectivas secciones de búsqueda. A continuación se pueden observar los diferentes parámetros que soporta la búsqueda en la sección de usuarios, permitiendo al root o residente los cuales son los únicos que pueden acceder a esta función, realizar una búsqueda más eficiente y veraz.



Figura 9. Captura de consulta de usuarios.

El comportamiento es el mismo para los diferentes tipos de consultas que se pueden hacer en el sistema de información. Las Figuras 10, 11 y 12 respectivamente muestran los resultados de realizar una búsqueda en los casos, pacientes y las patologías.

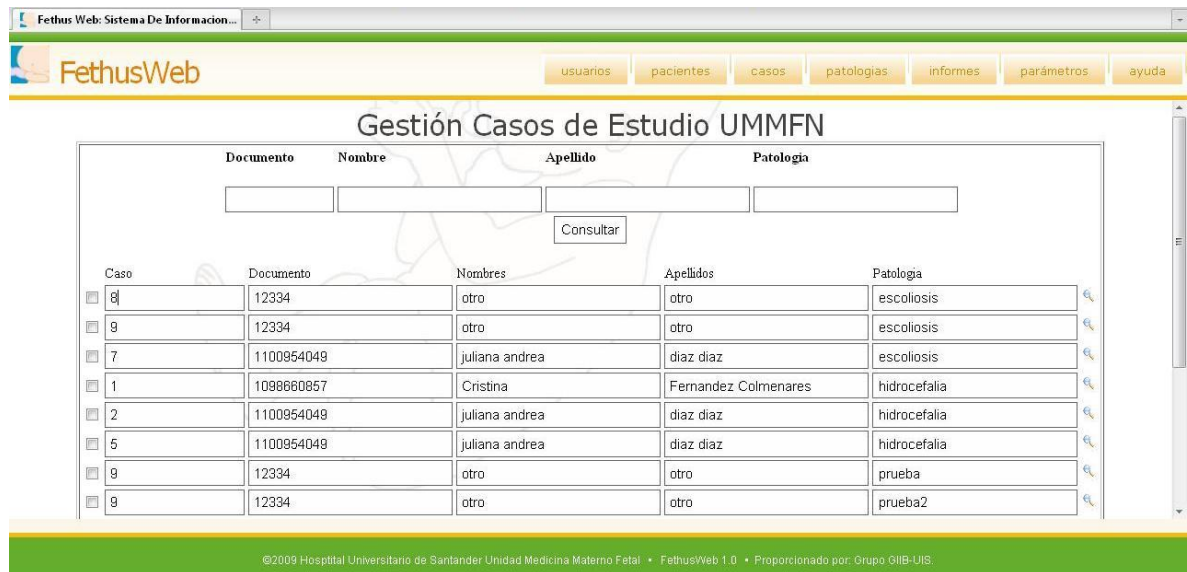


Figura 10. Captura de consulta de Casos de la UMMFN.

El usuario de tipo estudiantes únicamente puede hacer uso de la búsqueda por patologías generales, arrojándole ésta como resultado la información concerniente a dicha patología junto con algunos de los casos de la UMMFN que se relacionan con la patología en cuestión.

Para todas las clases de consultas que se puedan realizar al sistema los usuarios root y residente son los únicos que tienen acceso total.



Figura 11. Captura de consulta de pacientes de la UMMFN.

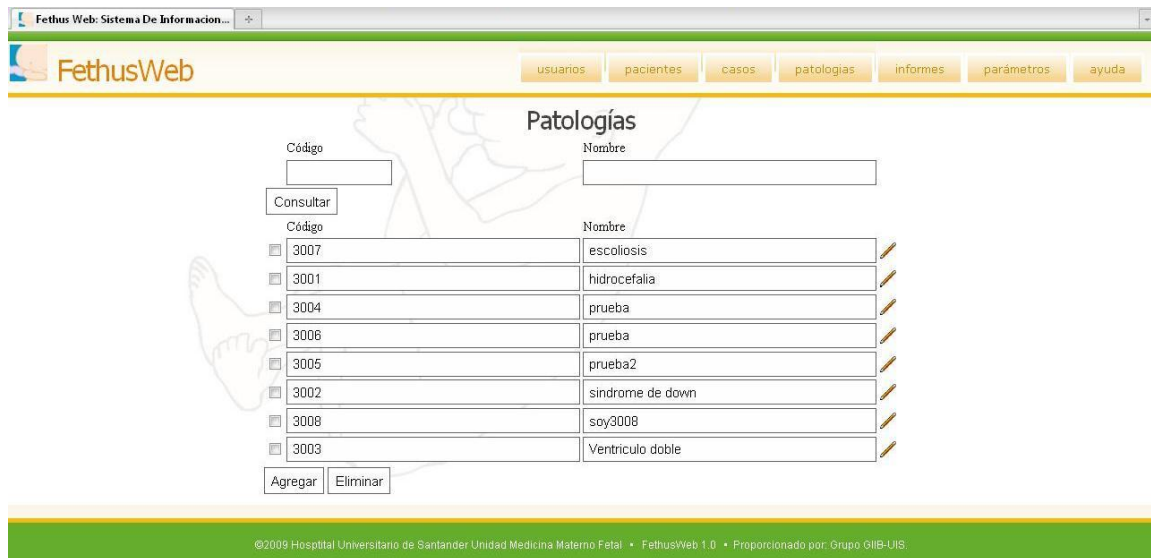


Figura 12. Captura de consulta por Patologías Generales.

## **5.5. MÓDULO DE INFORMES Y EPIDEMIOLOGÍA**

### **5.5.1. DESCRIPCIÓN**

El presente módulo tiene como principal objetivo generar diferentes tipos de informes a solicitud única y exclusiva de los usuarios root y residente. Estos informes pueden ser con diferentes fines, ya sea para presentarlos ante la junta médica de la UMMFN, o para realizar un estudio estadístico basado en la información almacenada en la base de datos de FETHUS.

### **5.5.2. REQUISITOS ASOCIADOS**

Los requisitos asociados, consignados en la Especificación de Requisitos de Software, corresponden a los requisitos funcionales desde el RF- 35 (Generar informe de usuarios) hasta el RF- 37 (Generar informe de mortalidad).

### **5.5.3. FUNCIONES**

Una parte importante de cualquier sistema de información es la generación de reportes. Aunque FETHUS tiene un generador de reportes muy sencillo, brinda suficiente información sobre los aspectos más relevantes para la UMMFN tales como informe de seguimiento a usuarios, informe de patologías relacionadas con casos de la Unidad e informe de resultado perinatal e informe sobre el índice de mortalidad en la Unidad, a continuación se muestra la vista de la generación de un reporte sobre el índice de uso del sistema por parte de los distintos usuarios en un intervalo de tiempo dado.

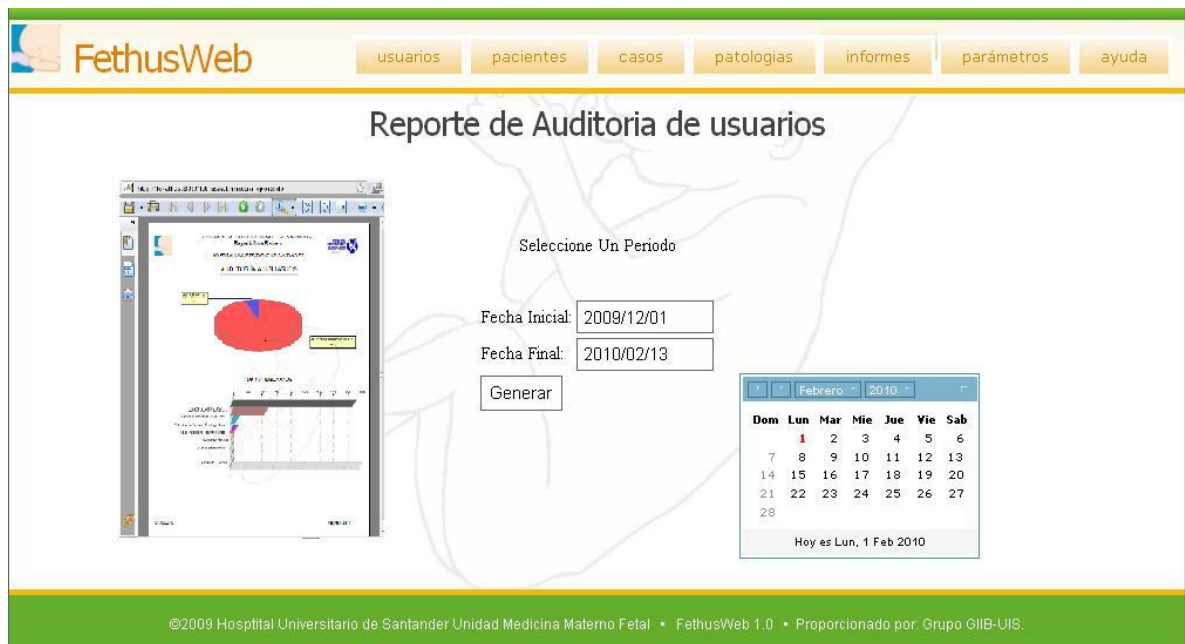


Figura 13. Reporte de utilización del sistema por parte de los usuarios.

## **5.6. MÓDULO DE GESTIÓN DE COPIAS DE SEGURIDAD**

### **5.6.1. DESCRIPCIÓN**

Todo sistema necesita de mantenimiento, y FETHUS no es la excepción. Por ello los autores han implementado un caso de uso fundamental a la hora de administrar cualquier software: la creación de copias de seguridad. Las funciones de este módulo se encaminan hacia este aspecto primordial.

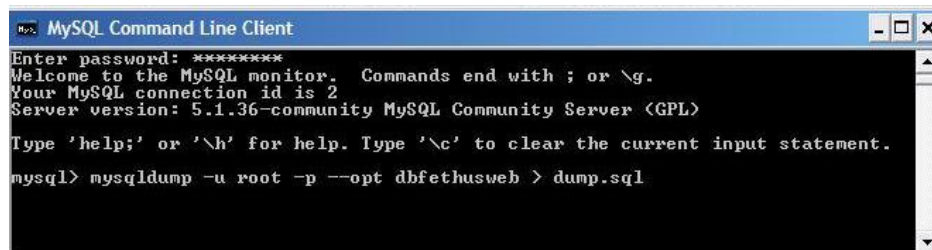
### **5.6.2. REQUISITOS ASOCIADOS**

Los requisitos asociados, consignados en la Especificación de Requisitos de Software, corresponden a los requisitos funcionales desde el RF-38 (Realizar copia de seguridad) hasta el RF-41 (Cargar contenido multimedia).

### 5.6.3. FUNCIONES

La creación y recuperación de copias de seguridad (Figuras 14 y 15) es de extrema importancia en todo sistema de información y el encargado de realizar dicha tarea es el administrador del servidor donde se encuentra alojada la aplicación FETHUS, con lo cual se generará un archivo de volcado escrito en lenguaje SQL que posteriormente debe ser cargado en caso tal que se necesite.

Esta función únicamente hace una copia de seguridad del contenido actual de la base de datos de FETHUS, la copia de las imágenes ecográficas tiene que hacerse manualmente, tarea que es responsabilidad del administrador del servidor en el cual se encuentra alojada la aplicación. En la sección 8.1 se explican los pasos a seguir para no perder las imágenes al momento de realizar una copia de seguridad o una actualización del sistema.

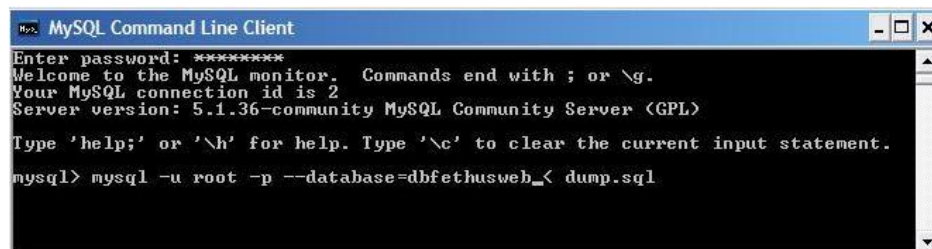


```
MySQL Command Line Client
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.1.36-community MySQL Community Server (GPL)

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> mysqldump -u root -p --opt dbfethusweb > dump.sql
```

Figura 14. Realización del backup a la base de datos.



```
MySQL Command Line Client
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.1.36-community MySQL Community Server (GPL)

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> mysql -u root -p --database=dbfethusweb < dump.sql
```

Figura 15. Cargar backup de la base de datos.

## 6. METODOLOGÍA

En este capítulo del libro se profundizará detalladamente sobre la metodología empleada en el desarrollo de sistema de información FETHUS. Después de estudiar y analizar los procedimientos que se deben seguir con diferentes tipos de metodologías, los autores del presente proyecto han optado por escoger la de Proceso Unificado de Racional (*RUP, Rational Unified Process*).

### 6.1. RUP

La metodología de Proceso Unificado Racional<sup>25</sup> es uno de los procesos más generales existentes actualmente, ya que en realidad está pensado para adaptarse a cualquier tipo de proyecto, y no tan solo al desarrollo de software. Se ha fortalecido con el tiempo y se enfoca en tres características principales: está dirigido por los Casos de Uso, está centrado en la arquitectura, y es iterativo e incremental.

#### Proceso dirigido por Casos de Uso

Los Casos de Uso son una técnica de captura de requisitos que fuerza a pensar en términos de importancia para el usuario y no sólo en términos de funciones que sería bueno contemplar. Se define un Caso de Uso como un fragmento de funcionalidad del sistema que proporciona al usuario un valor añadido. Los Casos de Uso representan los requisitos funcionales del sistema.

En RUP los Casos de Uso no son sólo una herramienta para especificar los requisitos del sistema. También guían su diseño, implementación y prueba. Los

---

<sup>25</sup> Rumbaugh, James; Jacobson, Ivar; Booch, Grady, El Lenguaje Unificado de Modelado, Addison Wesley, Madrid, 1999.

Casos de Uso constituyen un elemento integrador y una guía del trabajo como se muestra en la siguiente Figura.

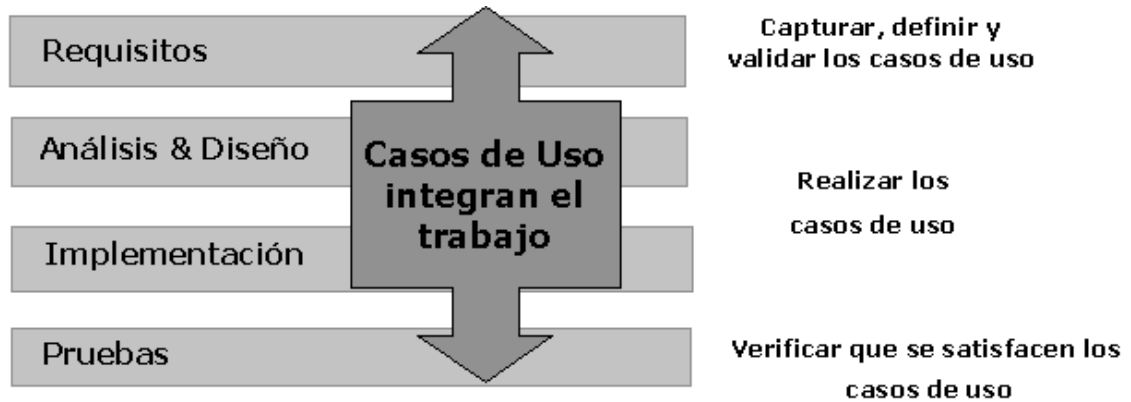


Figura 16. Los Casos de Uso integran el trabajo

### **Proceso centrado en la arquitectura**

La arquitectura de un sistema es la organización o estructura de sus partes más relevantes, e involucra los aspectos estáticos y dinámicos más significativos del sistema, está relacionada con la toma de decisiones que indican cómo tiene que ser construido el sistema y ayuda a determinar en qué orden. La arquitectura se ve influenciada por la plataforma software, sistema operativo, gestor de bases de datos, protocolos, consideraciones de desarrollo como sistemas heredados. Muchas de estas restricciones constituyen requisitos no funcionales del sistema. RUP presta especial atención al establecimiento temprano de una buena arquitectura que no se vea fuertemente impactada ante cambios posteriores durante la construcción y mantenimiento.

### **Proceso iterativo e incremental**

Para lograr un equilibrio correcto entre los Casos de Uso y la arquitectura, la estrategia que se propone en RUP es tener un proceso iterativo e incremental en

donde el trabajo se divide en partes más pequeñas o mini proyectos. Permitiendo que el equilibrio entre Casos de Uso y arquitectura se vaya logrando durante cada mini proyecto, así durante todo el proceso de desarrollo. Cada mini proyecto se puede ver como una iteración del cual se obtiene un incremento que produce un crecimiento en el producto. Una iteración puede realizarse por medio de una cascada como se muestra en la Figura 17.

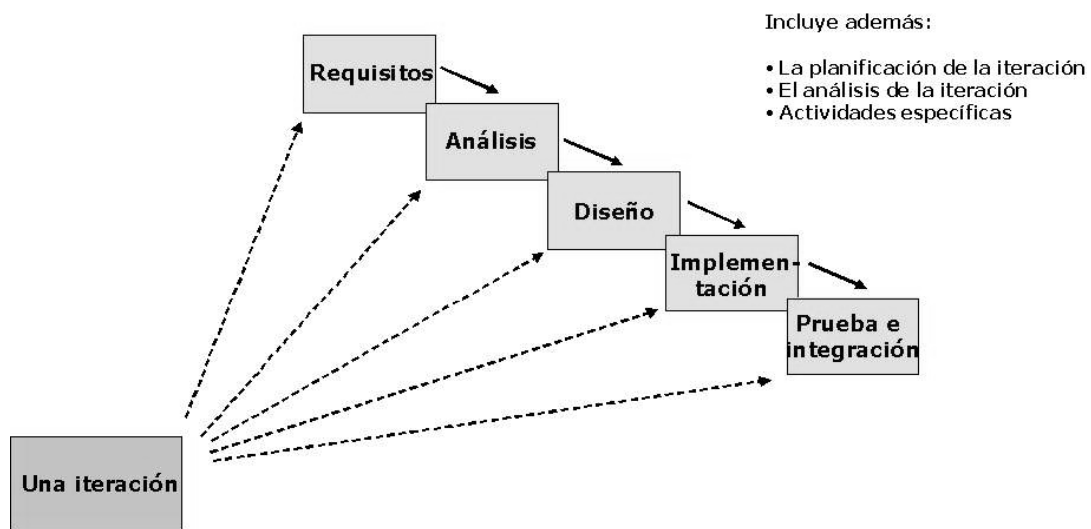


Figura 17. Una iteración RUP

### 6.1.1. FASES DEL RUP

RUP se repite a lo largo de una serie de ciclos que constituyen la vida de un producto. Cada ciclo concluye con una generación del producto para los clientes. Cada ciclo consta de cuatro fases: Inicio, Elaboración, Construcción y Transición. Cada fase se subdivide a la vez en iteraciones, el número de iteraciones en cada fase es variable.

Cada fase se concluye con un hito bien definido, un punto en el tiempo en el cual se deben tomar ciertas decisiones críticas y alcanzar las metas clave antes de pasar a la siguiente fase, ese hito principal de cada fase se compone de hitos menores que podrían ser los criterios aplicables a cada iteración. Las fases y sus respectivos hitos se ilustran en la siguiente Figura.

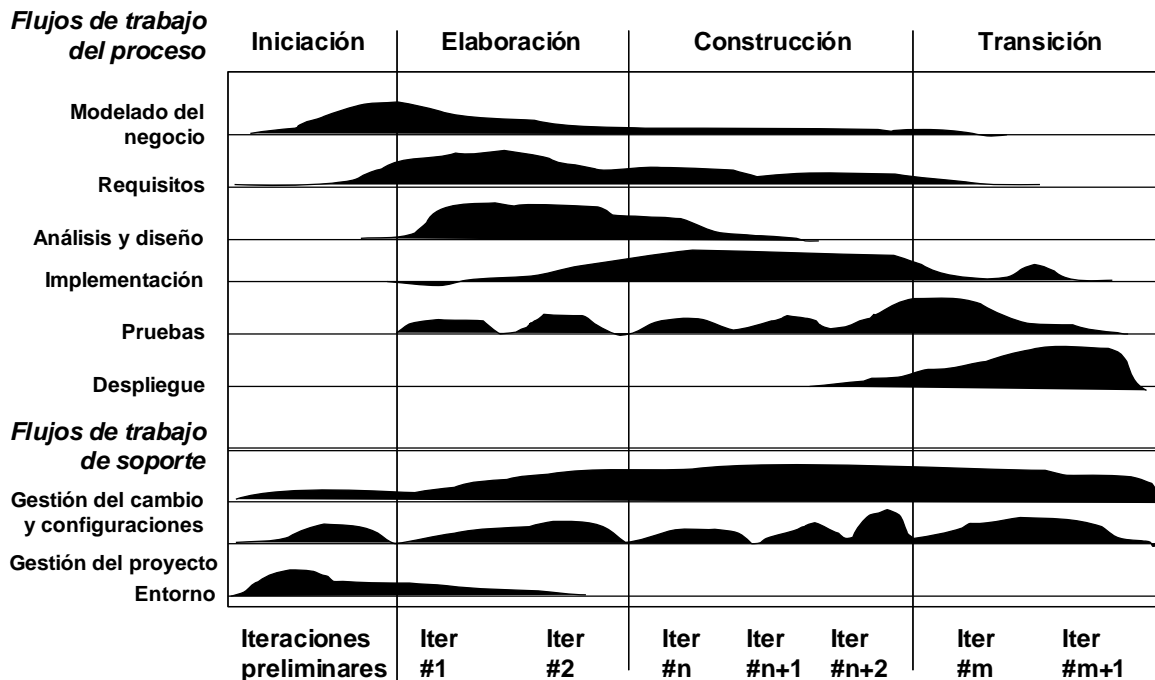


Figura 18. Estructura de RUP.

## Inicio

Durante la fase de inicio se define el modelo del negocio y el alcance del proyecto. Se identifican todos los actores y Casos de Uso, y se diseñan los Casos de Uso más esenciales (aproximadamente el 20% del modelo completo). Se desarrolla, un plan de negocio para determinar qué recursos deben ser asignados al proyecto.

Al terminar la fase de inicio se deben comprobar los criterios de evaluación para continuar, si el proyecto no pasa estos criterios hay que plantearse abandonarlo o repensarlo profundamente.

## **Elaboración**

El propósito de la fase de elaboración es analizar el dominio del problema, establecer los cimientos de la arquitectura, desarrollar el plan del proyecto y eliminar los mayores riesgos.

En esta fase se construye un prototipo de la arquitectura, que debe evolucionar en iteraciones sucesivas hasta convertirse en el sistema final. Este prototipo debe contener los Casos de Uso críticos identificados en la fase de inicio. También debe demostrarse que se han evitado los riesgos más graves.

En esta fase se debe tratar de abarcar todo el proyecto con la profundidad mínima. Sólo se profundiza en los puntos críticos de la arquitectura o riesgos importantes. En la fase de elaboración se actualizan todos los productos de la fase de inicio y se analizan los criterios de evaluación de esta fase, si no se superan los criterios de evaluación quizá sea necesario abandonar el proyecto o replanteárselo considerablemente.

## **Construcción**

La finalidad principal de esta fase es alcanzar la capacidad operacional del producto de forma incremental a través de las sucesivas iteraciones. Durante esta fase todos los componentes, características y requisitos deben ser implementados, integrados y probados en su totalidad, obteniendo una versión aceptable del producto.

## **Transición**

La finalidad de la fase de transición es poner el producto en manos de los usuarios finales, para lo que se requiere desarrollar nuevas versiones actualizadas del producto, completar la documentación, entrenar al usuario en el manejo del producto, y en general tareas relacionadas con el ajuste, configuración, instalación

y facilidad de uso del producto.

### **Flujos de trabajo<sup>26</sup>**

El RUP organiza a los proyectos en términos de flujos de trabajo y fases, en cada fase se ejecutarán una o varias iteraciones (de tamaño variable según el proyecto), y dentro de cada una de ellas seguirá un modelo de cascada para los flujos de trabajo que requieren las nuevas actividades.

### **Modelado del negocio**

Con este flujo de trabajo se pretende llegar a un mejor entendimiento de la organización donde se va a implantar el producto.

Para lograr los objetivos que se plantean en este flujo, el modelo de negocio describe como desarrollar una visión de la nueva organización, basado en esta visión se definen procesos, roles y responsabilidades de la organización por medio de un modelo de Casos de Uso del negocio y un Modelo de Objetos del Negocio. Complementario a estos modelos, se desarrollan otras especificaciones tales como un Glosario.

### **Requisitos**

Este es uno de los flujos de trabajo más importantes, porque en él se establece qué tiene que hacer exactamente el sistema que construyamos. En esta línea los requisitos son el contrato que se debe cumplir, de modo que los usuarios finales tienen que comprender y aceptar los requisitos que se especifiquen.

Los requisitos se dividen en dos grupos. Los requisitos funcionales representan la funcionalidad del sistema. Se modelan mediante diagramas de Casos de Uso. Los requisitos no funcionales representan aquellos atributos que debe exhibir el

---

<sup>26</sup> BOLAÑOS CASTRO, Sandro Javier. Análisis Y Diseño del Portal de Conocimiento de la Universidad Distrital Francisco José de Caldas. Bogotá 2004.

sistema, pero que no son una funcionalidad específica. Por ejemplo requisitos de facilidad de uso, fiabilidad, eficiencia, portabilidad, etc.

Para capturar los requisitos es preciso entrevistar a todos los interesados en el proyecto, no sólo a los usuarios finales, y anotar todas sus peticiones. A partir de ellas hay que descubrir lo que necesitan y expresarlo en forma de requisitos.

## **Análisis y Diseño**

El objetivo de este flujo de trabajo es traducir los requisitos a una especificación que describe cómo implementar el sistema.

El análisis consiste en obtener una visión del sistema que se preocupa de ver qué hace, de modo que sólo se interesa por los requisitos funcionales. Por otro lado el diseño es un refinamiento del análisis que tiene en cuenta los requisitos no funcionales, en definitiva cómo cumple el sistema sus objetivos.

Al principio de la fase de elaboración hay que definir una arquitectura candidata: crear un esquema inicial de la arquitectura del sistema, identificar clases de análisis y actualizar las realizaciones de los Casos de Uso con las interacciones de las clases de análisis. Durante la fase de elaboración se va refinando esta arquitectura hasta llegar a su forma definitiva. En cada iteración hay que analizar el comportamiento para diseñar componentes. Además si el sistema usará una base de datos, habrá que diseñarla también, obteniendo un modelo de datos.

El resultado final más importante de este flujo de trabajo será el modelo de diseño. Consiste en colaboraciones de clases, que pueden ser agregadas en paquetes y subsistemas. Otro producto importante de este flujo es la documentación de la arquitectura de software, que captura varias vistas arquitectónicas del sistema.

## **Implementación**

En este flujo de trabajo se implementan las clases y objetos en ficheros fuente,

binarios, ejecutables y demás. Además se deben hacer las pruebas de unidad: cada implementador es responsable de probar las unidades que produzca. El resultado final de este flujo de trabajo es un sistema ejecutable.

La estructura de todos los elementos implementados forma el modelo de implementación. La integración debe ser incremental, es decir, en cada momento sólo se añade un elemento. De este modo es más fácil localizar fallos y los componentes se prueban más a fondo. En fases tempranas del proceso se pueden implementar prototipos para reducir el riesgo. Su utilidad puede ir desde ver si el sistema es viable desde el principio, probar tecnologías o diseñar la interfaz de usuario. Los prototipos pueden ser exploratorios (desechables) o evolutivos. Estos últimos llegan a transformarse en el sistema final.

## **Pruebas**

Este flujo de trabajo es el encargado de evaluar la calidad del producto que estamos desarrollando, pero no para aceptar o rechazar el producto al final del proceso de desarrollo, sino que debe ir integrado en todo el ciclo de vida.

Esta disciplina brinda soporte a las otras disciplinas. Las actividades de este flujo comienzan pronto en el proyecto con el plan de prueba (el cual contiene información sobre los objetivos generales y específicos de las prueba en el proyecto, así como las estrategias y recursos con que se dotará a esta tarea), o incluso antes con alguna evaluación durante la fase de inicio, y continuará durante todo el proyecto.

El desarrollo del flujo de trabajo consistirá en planificar que es lo que hay que probar, diseñar cómo se va a hacer, implementar lo necesario para llevarlos a cabo, ejecutarlos en los niveles necesarios y obtener los resultados, de forma que la información obtenida nos sirva para ir refinando el producto a desarrollar.

## **Despliegue**

El objetivo de este flujo de trabajo es producir con éxito distribuciones del producto y distribuirlo a los usuarios.

Este flujo de trabajo se desarrolla con mayor intensidad en la fase de transición, ya que el propósito del flujo es asegurar una aceptación y adaptación sin complicaciones del software por parte de los usuarios. Su ejecución inicia en fases anteriores, para preparar el camino, sobre todo con actividades de planificación, en la elaboración del manual de usuario y tutoriales.

### **Gestión del proyecto**

La Gestión del proyecto es el arte de lograr un balance al gestionar objetivos, riesgos y restricciones para desarrollar un producto que sea acorde a los requisitos de los clientes y los usuarios.

La planeación de un proyecto posee dos niveles de abstracción: un plan para las fases y un plan para cada iteración.

### **Configuración y control de cambios**

La finalidad de este flujo de trabajo es mantener la integridad de todos los artefactos que se crean en el proceso, así como de mantener información del proceso evolutivo que han seguido.

### **Entorno**

La finalidad de este flujo de trabajo es dar soporte al proyecto con las adecuadas herramientas, procesos y métodos. Brinda una especificación de las herramientas que se van a necesitar en cada momento, así como definir la instancia concreta del proceso que se va a seguir.

El principal artefacto que se usa en este flujo de trabajo es el *caso de desarrollo* que especifica para el proyecto actual en concreto, como se aplicará el proceso, que productos se van a utilizar y como van a ser utilizados. Además se tendrán que definir las guías para los distintos aspectos del proceso, como pueden ser el

modelado del negocio y los Casos de Uso, para la interfaz de usuario, el diseño, la programación, el manual de usuario.

### **6.1.2. CASOS DE USO**

Como se ha dicho anteriormente, el RUP basa su estructura en los casos de uso. Pero ¿En qué consiste exactamente un caso de uso?

Un caso de uso se define como un fragmento de funcionalidad del sistema que especifica el comportamiento de un sistema o de una parte del mismo. Los casos de uso representan los requisitos funcionales del sistema. En RUP los casos de uso no son sólo una herramienta para especificar los requisitos del sistema, también guían su diseño, implementación y prueba. Los casos de uso constituyen un elemento integrador y una guía del trabajo.

Los casos de uso no sólo inician el proceso de desarrollo sino que proporcionan un hilo conductor, permitiendo establecer trazabilidad entre los artefactos que son generados en las diferentes actividades del proceso de desarrollo. Basándose en los casos de uso se crean los modelos de análisis y diseño, luego la implementación que los lleva a cabo, y se verifica que efectivamente el producto implemente adecuadamente cada caso de uso. Todos los modelos deben estar sincronizados con el modelo de casos de uso. A continuación se muestra como ejemplo un caso de uso especificado para el presente proyecto.

<b>RF-01</b>	<b>Crear cuentas de usuario</b>
<b>Versión</b>	1.0
<b>Objetivos</b>	OBJ-06 Administrar el acceso y los permisos de los usuarios

<b>asociados</b>		
<b>Requisitos asociados</b>	RI-03 Información sobre usuarios del sistema RF-02 Asignar permisos	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el root desee crear una cuenta de usuario.	
<b>Precondición</b>	El usuario no está registrado en el sistema de información. El root del sistema conoce la información personal del usuario.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El root ingresa al menú de administración de cuentas de usuario y selecciona la opción: Crear nueva cuenta de Usuario.
	2	El sistema solicita mediante un formulario toda la información referente al usuario. (Nombre, apellidos, login password, pregunta secreta, respuesta pregunta secreta, e-mail, tipo de usuario.)
	3	El root proporciona los datos solicitados.
	4	El sistema valida la información, si existe alguna incongruencia, se vuelve al paso 2.
	5	El sistema muestra un mensaje de confirmación y envía un correo electrónico de notificación al nuevo usuario.
<b>Pos condición</b>	El nuevo usuario esta registrado con determinados privilegios.	

<b>Excepciones</b>	Ninguna
--------------------	---------

Tabla 1. Caso de uso “*crear cuentas de usuario*”.

### **6.1.3. DIAGRAMAS DE CASOS DE USO**

Un Diagrama de Casos de Uso muestra la relación entre los actores y los casos de uso del sistema. Representa la funcionalidad que ofrece el sistema en lo que se refiere a su interacción externa. En el diagrama de casos de uso se representa también el sistema como una caja rectangular con el nombre en su interior. Los casos de uso están en el interior de la caja del sistema, y los actores fuera, y cada actor está unido a los casos de uso en los que participa mediante una línea. En la sección 8.5.1 se muestran los Diagramas de casos de uso para el sistema FETHUS.

Los elementos que pueden aparecer en un Diagrama de Casos de Uso son: actores, casos de uso y relaciones entre casos de uso.

**Actores:** Un actor es algo con comportamiento, como una persona (identificada por un rol), un sistema informatizado u organización, y que realiza algún tipo de interacción con el sistema. Se representa mediante una figura humana dibujada con líneas. Esta representación sirve tanto para actores que son personas como para otro tipo de actores.

**Casos de Uso:** Un caso de uso es una descripción de la secuencia de interacciones que se producen entre un actor y el sistema, cuando el actor usa el sistema para llevar a cabo una tarea específica. Expresa una unidad coherente de funcionalidad, y se representa en el Diagrama de Casos de Uso mediante una elipse con el nombre del caso de uso en su interior. El nombre del caso de uso debe reflejar la tarea específica que el actor desea llevar a cabo usando el sistema.

**Relaciones entre casos de uso:** Un caso de uso, en principio, debería describir una tarea que tiene un sentido completo para el usuario. Sin embargo, hay ocasiones en las que es útil describir una interacción con un alcance menor como caso de uso. La razón para utilizar estos casos de uso no completos en algunos casos, es mejorar la comunicación con el equipo de desarrollo. Para el caso de que queramos utilizar estos casos de uso más pequeños, las relaciones entre estos y los casos de uso ordinarios pueden ser de los siguientes tres tipos:

- **Inclusión (<<include>>):** Un caso de uso base incorpora explícitamente a otro caso de uso en un lugar especificado en dicho caso base. Se suele utilizar para encapsular un comportamiento parcial común a varios casos de uso.
- **Extensión (<<extend>>):** Cuando un caso de uso base tiene ciertos puntos (puntos de extensión) en los cuales, dependiendo de ciertos criterios, se va a realizar una interacción adicional. El caso de uso que extiende describe un comportamiento opcional del sistema (a diferencia de la relación incluye que se da siempre que se realiza la interacción descrita).

Ambos tipos de relación se representan como una flecha con línea discontinua etiquetada con el estereotipo correspondiente (<<include>> o <<extend>>), de tal forma que la flecha indique el sentido en el que debe leerse la etiqueta, teniendo en cuenta que dicha flecha representa una relación de dependencia.

- **Generalización:** Cuando un caso de uso definido de forma abstracta se particulariza por medio de otro caso de uso más específico. Se representa por una línea continua entre los dos casos de uso, con el triángulo que simboliza generalización pegado al extremo del caso de uso más general. El caso de uso hijo hereda las asociaciones y características del caso de uso padre. El caso de uso padre se trata de un caso de uso abstracto, que no está definido completamente.

## 6.2. ARQUITECTURA DE LA APLICACIÓN

La aplicación está soportada por una plataforma operativa como se muestra en la Figura 19 conformada por el sistema operativo, el sistema manejador de base de datos, el servidor Web y los navegadores de Internet. Sobre dicha plataforma operativa se encuentran interactuando las tres capas del modelo basado en Web a través de interfaces seguras para proteger la aplicación.



Figura 19. Plataforma operativa de FETHUS.

La capa de las reglas del negocio se encuentra dividida en subsistemas que poseen funciones bien definidas así:

**Subsistema de administración de usuarios:** es el encargado de manejar la información relacionada con las personas que interactúan con el sistema de información, ya sean root, residente o estudiante. Permite la creación, modificación y eliminación de usuarios y perfiles de usuario.

**Subsistema de administración de casos de la UMMFN:** permite gestionar los casos de anomalías perinatales detectados en la UMMFN. Entre sus funciones están las de registrar, editar, consultar y eliminar casos.

**Subsistema de administración de patologías generales:** su función principal es gestionar toda la información referente a las patologías tales como incidencia,

características patológicas, características ecográficas, estudios complementarios, pronóstico y tratamiento.

**Subsistema de generación de consultas:** es el encargado de apoyar y facilitar las consultas del usuario Web.

**Subsistema de informes epidemiológicos:** permite generar informes estadísticos a solicitud del usuario, respecto a las variables de cada caso y/o a las características demográficas del paciente.

**Subsistema de gestión de copias de seguridad:** es el encargado de ofrecer funciones de respaldo y copias de seguridad.

### ***6.3. ITERACIONES DE LA FASE DE CONSTRUCCIÓN***

Para la fase de construcción se realizaron 5 iteraciones en las cuales se ejecutaron los flujos de trabajo de requisitos, análisis y diseño, implementación, prueba, administración de cambios y configuraciones y administración del proyecto. Para cada iteración se asignaron los subsistemas de la siguiente manera:

**Iteración 1:** subsistema de administración de usuarios.

**Iteración 2:** subsistema de administración de casos de la UMMFN.

**Iteración 3:** subsistema de administración de patologías generales.

**Iteración 4:** subsistema de generación de consultas.

**Iteración 5:** subsistema de informes epidemiológicos. Y subsistema de gestión de copias de seguridad.

#### 6.4. CRONOGRAMA DE ACTIVIDADES PARA UNA ITERACIÓN

El siguiente cronograma fue el que se elaboró inicialmente para cada una de las iteraciones de la fase de construcción, como se podrá observar, cada iteración está conformada por tres semanas de trabajo de la siguiente manera:

<b>Semana 1</b>	
<b>Día</b>	<b>Actividad</b>
<b>Lunes</b>	<ul style="list-style-type: none"><li>• Elaborar el diagrama de casos de uso.</li><li>• Elaborar la especificación de Requisitos Funcionales.</li></ul>
<b>Martes</b>	<ul style="list-style-type: none"><li>• Elaborar diagrama de clases para las reglas del negocio.</li><li>• Elaborar Diagrama entidad-relación.</li><li>• Reunión del comité de sistemas de información.<ul style="list-style-type: none"><li>○ Revisión de la especificación de requisitos.</li></ul></li></ul>
<b>Miércoles</b>	<ul style="list-style-type: none"><li>• Corregir diagramas.</li><li>• Corregir requisitos.</li><li>• Corregir código.</li><li>• Hacer pruebas.</li></ul>
<b>Jueves</b>	<ul style="list-style-type: none"><li>• Elaborar el modelo de implementación.</li><li>• Desarrollar las clases de reglas del negocio.</li><li>• Desarrollar la base de datos.</li><li>• Hacer pruebas.</li></ul>
<b>Viernes</b>	<ul style="list-style-type: none"><li>• Desarrollar las clases de reglas del negocio.</li><li>• Desarrollar la base de datos.</li><li>• Hacer pruebas.</li></ul>

Tabla 2. Cronograma de actividades para la semana 1.

<b>Semana 2</b>	
<b>Día</b>	<b>Actividad</b>
<b>Lunes</b>	<ul style="list-style-type: none"> <li>• Desarrollar clases de reglas del negocio.</li> <li>• Desarrollar las interfaces entre la base de datos y las clases de reglas del negocio.</li> <li>• Hacer pruebas.</li> </ul>
<b>Martes</b>	<ul style="list-style-type: none"> <li>• Desarrollar las interfaces entre las bases de datos y las clases de reglas del negocio.</li> <li>• Hacer pruebas.</li> <li>• Reunión del comité de sistemas de información.</li> </ul>
<b>Miércoles</b>	<ul style="list-style-type: none"> <li>• Corregir diagramas.</li> <li>• Corregir requisitos.</li> <li>• Corregir código.</li> <li>• Hacer pruebas.</li> </ul>
<b>Jueves</b>	<ul style="list-style-type: none"> <li>• Elaborar el diagrama de clases para la interfaz de usuario.</li> <li>• Desarrollar las clases de interfaz de usuario.</li> <li>• Hacer pruebas.</li> </ul>
<b>Viernes</b>	<ul style="list-style-type: none"> <li>• Desarrollar las clases de interfaz de usuario.</li> <li>• Desarrollar las interfaces entre las clases de reglas del negocio y las clases de interfaz de usuario.</li> <li>• Elaborar las alternativas de la interfaz gráfica de usuario.</li> <li>• Hacer pruebas.</li> </ul>

Tabla 3. Cronograma de actividades para la semana 2.

<b>Semana 3</b>	
<b>Día</b>	<b>Actividad</b>
<b>Lunes</b>	<ul style="list-style-type: none"> <li>• Elaborar las alternativas de la interfaz gráfica de usuario.</li> <li>• Implementar el prototipo funcional.</li> <li>• Hacer pruebas.</li> </ul>
<b>Martes</b>	<ul style="list-style-type: none"> <li>• Elaborar las alternativas de la interfaz gráfica de usuario.</li> <li>• Implementar el prototipo funcional.</li> <li>• Hacer pruebas.</li> <li>• Reunión del comité de sistemas de información. <ul style="list-style-type: none"> <li>○ Selección, por parte del comité, de una alternativa de interfaz gráfica de usuario.</li> <li>○ Revisión del prototipo funcional.</li> </ul> </li> </ul>
<b>Miércoles</b>	<ul style="list-style-type: none"> <li>• Corregir diagramas.</li> <li>• Corregir requisitos.</li> <li>• Corregir código.</li> <li>• Hacer pruebas.</li> </ul>
<b>Jueves</b>	<ul style="list-style-type: none"> <li>• Elaborar manuales de usuario y administrador.</li> </ul>
<b>Viernes</b>	<ul style="list-style-type: none"> <li>• Elaborar la documentación del proyecto.</li> <li>• Gestión del proyecto.</li> <li>• Mostrar el avance del proyecto al director.</li> </ul>

Tabla 4. Cronograma de actividades para la semana 3.

Las actividades enunciadas anteriormente han sido realizadas en su totalidad, pero no se cumplió con el tiempo estimado para este cronograma debido a la inexperiencia de los autores.

La primera iteración tomo más tiempo que las demás, debido a que los autores tuvieron que aprender a utilizar las herramientas y tecnología seleccionada. Aunque otro factor que incidió en el retraso fue el haber incluido demasiados casos de uso para una sola iteración, tomando incluso el doble del tiempo que se tenía presupuestado inicialmente. A partir de la segunda iteración el tiempo de desarrollo se redujo sustancialmente lo que permitió recuperar parte del tiempo invertido en la primera iteración.

A continuación se describirá el plan de trabajo real llevado a cabo por los autores durante la fase de construcción.

### **Actividades realizadas en cada iteración de la fase de construcción**

✓ Actividad 1.

- Elaborar el diagrama de casos de uso.
- Elaborar la especificación de Requisitos Funcionales.

✓ Actividad 2.

- Elaborar diagrama de clases para las reglas del negocio.
- Elaborar diagrama de clases para interfaz con la base de datos.
- Elaborar el diagrama de clases para la interfaz de usuario.
- Elaborar Diagrama entidad-relación.

✓ Actividad 3.

- Reunión del comité de sistemas de información.
  - Revisión de la especificación de requisitos y diseño preliminar.
- Corregir diagramas.
- Corregir requisitos.

- Gestión del proyecto (dividir el trabajo).
- ✓ Actividad 4.
  - Elaborar el modelo de implementación.
  - Desarrollar la base de datos.
  - Desarrollar las clases de reglas del negocio.
- ✓ Actividad 5.
  - Elaborar pruebas automáticas para las clases de reglas del negocio desarrolladas.
  - Corregir código defectuoso.
- ✓ Actividad 6.
  - Desarrollar las clases de interfaz entre las bases de datos y las clases de reglas del negocio.
- ✓ Actividad 7.
  - Elaborar pruebas automáticas para las clases de interfaz con la base de datos desarrolladas.
  - Corregir código defectuoso.
- ✓ Actividad 8.
  - Desarrollar las clases de interfaz de usuario.
- ✓ Actividad 9.
  - Elaborar manual del programador.
- ✓ Actividad 10.

- Hacer pruebas de todos los elementos desarrollados a través de la interfaz grafica de usuario.
- Corregir código defectuoso.
- ✓ Actividad 11.
  - Reunión del comité de sistemas de información para revisar el prototipo funcional.

**Actividades realizadas al final de la fase de construcción**

- Elaborar manuales de usuario.
- Elaborar la documentación del proyecto.
- Probar el sistema en su totalidad.
- Elaborar el acta de entrega.

## **7. DESARROLLO**

### **7.1. ANÁLISIS**

Los autores del presente proyecto en esta primera etapa, han analizado los requisitos básicos del personal de la UMMFN. Para hacer esto, se han realizado diferentes reuniones con el objetivo de identificar los potenciales usuarios del sistema a elaborar y sus necesidades. También se ha entrevistado al personal de la UMMFN para tener información de primera mano sobre los procedimientos que se realizan allí. Las entrevistas han sido grabadas para no perder detalle.

Una vez se conocen los procedimientos de la UMMFN, se da inicio a la elaboración del plan de proyecto que ha sido presentado ante la EISI y respectivamente aprobado. Ya con el plan de proyecto aprobado se prosigue a la investigación de las diferentes metodologías y tecnologías existentes para el desarrollo de proyectos de software, llegando a la conclusión de implementar el proyecto bajo la metodología RUP, utilizando tecnología Java, más exactamente "*Java 2 plataform, Enterprise Edition*".

El resultado de toda la fase de análisis se encuentra plasmado en el documento d Especificación de Requisitos de Software, elaborado por los autores del presente proyecto con base en la información recolectada durante las reuniones y entrevistas realiza al personal de la UMMFN, en el Hospital Universitario de Santander. Dicho documento está disponible en el anexo A del presente texto.

## **7.2. DISEÑO**

En este apartado se profundizará en el proceso del diseño por medio del modelado, describiendo todos los aspectos que el sistema FETHUS debe cumplir. Una vez se finaliza con esta etapa, se tiene una idea completa de lo que es en sí el software, enfocando los dominios de datos, funcional y comportamiento desde el punto de vista de la Implementación.

Los siguientes criterios han sido establecidos para el diseño de FETHUS:

- El diseño debe presentar una organización jerárquica que haga un uso inteligente del control entre los componentes del software.
- El diseño debe ser modular, es decir, se debe hacer una partición lógica del Software en elementos que realicen funciones y subfunciones específicas.
- El diseño debe contener abstracciones de datos y procedimientos.
- Producir módulos que presenten características de funcionamiento independiente.

Esta fase de diseño ha exigido la aplicación de principios fundamentales de diseño, metodología sistemática y una revisión exhaustiva, obteniendo como resultado la elaboración de algunos de los diagramas UML más importantes que se mostrarán a continuación.

### **7.2.1. DIAGRAMA DE CASOS DE USO**

Los diagramas de casos de uso han sido elaborados con base en las entrevistas realizadas al personal de la UMMFN durante la fase de análisis. Dichos diagramas que corresponden a FETHUS se encuentran adjuntos en el Anexo A, ya que hacen parte del documento de Especificación de Requisitos de Software y por esta razón no serán presentados en esta sección.

### 7.2.2. DIAGRAMA DE DESPLIEGUE

Como podemos observar en la Figura 20, la interacción entre el usuario y el sistema es del tipo cliente-servidor. Los componentes que forman parte del servidor son la base de datos *dbfethusweb*, el servidor web, y la aplicación FETHUS. Entre tanto, el nodo del cliente cuenta simplemente con un navegador web y la interfaz gráfica de usuario.

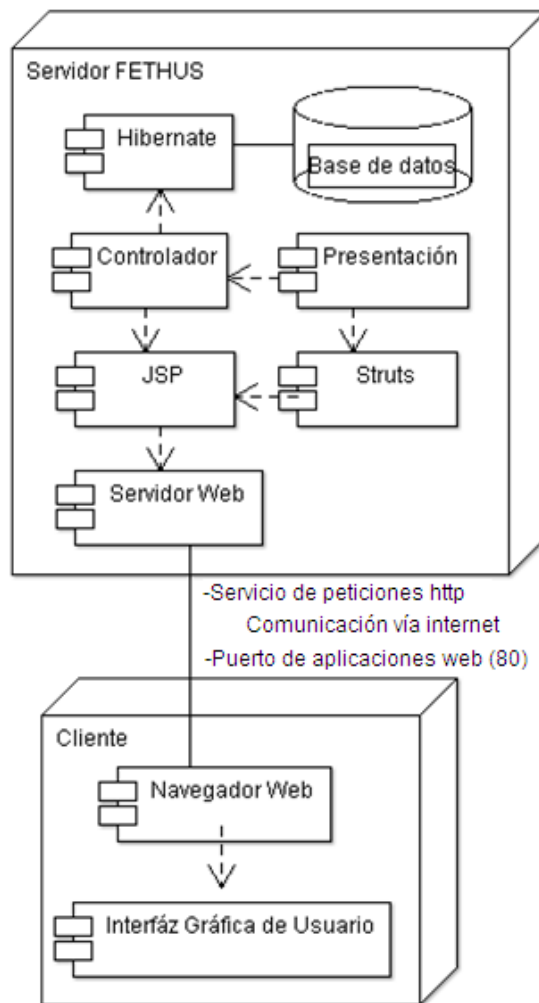


Figura 20. Diagrama de despliegue de FETHUS.

### 7.2.3. DIAGRAMAS DE ACTIVIDADES

A continuación se muestran los flujos de trabajo del negocio y las operaciones de los componentes de FETHUS, mostrando así el flujo de control general de la aplicación, desde que el cliente realiza una petición por medio del navegador web hasta que se utiliza la base de datos, y viceversa.

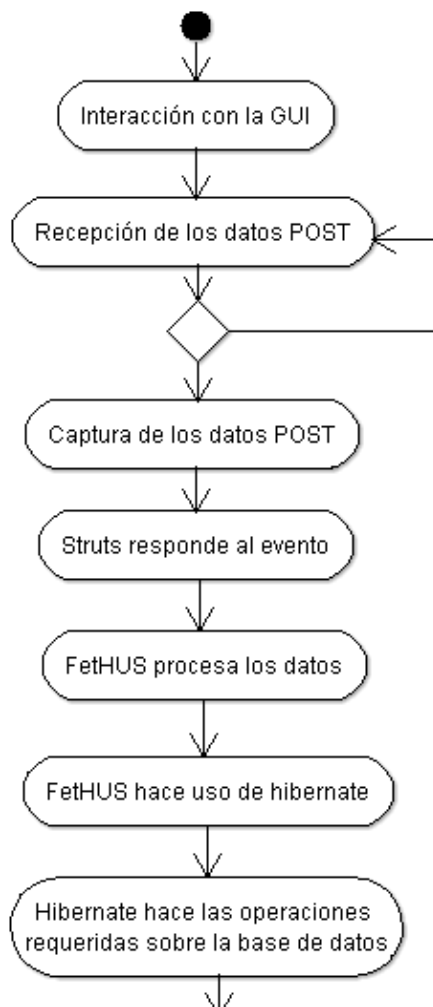


Figura 21. Diagrama de actividades que muestra el proceso de respuesta a las peticiones del cliente desde el uso de la interfaz grafica de usuario hasta que se utiliza la base de datos.

En la Figura 21, se describe gráficamente el flujo de trabajo desde que el usuario interactúa con la interfaz gráfica del sistema, hasta que se utiliza la base de datos. En primer lugar, el usuario diligencia el formulario, utilizando los controles proveídos por el código HTML y JSP de la página. El usuario pulsa el botón que envía el formulario (POST) e inmediatamente el sistema verifica que la información del formulario cumple con ciertos requisitos válidos. Una vez la información es validada correctamente, el contenido del formulario es enviado vía HTTP hacia el servidor, que recibe los datos enviados por el usuario.

Es aquí donde entran en acción los métodos encargados de manejar el evento ocasionado por el usuario y recolectar los datos enviados, y a partir de este momento, el control del proceso es llevado a cabo por FETHUS mediante las clases implementadas en la capa de las reglas del negocio. Estas clases toman los datos por medio de los objetos instanciados para encapsular la información proveniente del usuario y los métodos se encargan de procesar dicha información.

Finalmente se instancia la clase de la capa de datos, la cual hace uso de Hibernate para realizar las tareas que el usuario ha ordenado inicialmente, ya sea inserción, actualización, eliminación o consulta de determinados parámetros.

El flujo de vuelta se inicia en el momento en que el manejador de bases de datos ejecuta las consultas enviadas por el Hibernate, las cuales son devueltas al modelo y son recuperadas por la clase instanciada en la interfaz de datos, es en este punto donde FETHUS retoma el control sobre la aplicación.

Las clases implementadas en el controlador procesan los datos mediante los diferentes métodos y devuelven a la capa de presentación los resultados, siendo esta última la encargada de mostrar dichos resultados al usuario final, este flujo de vuelta se describe gráficamente en la Figura 22.

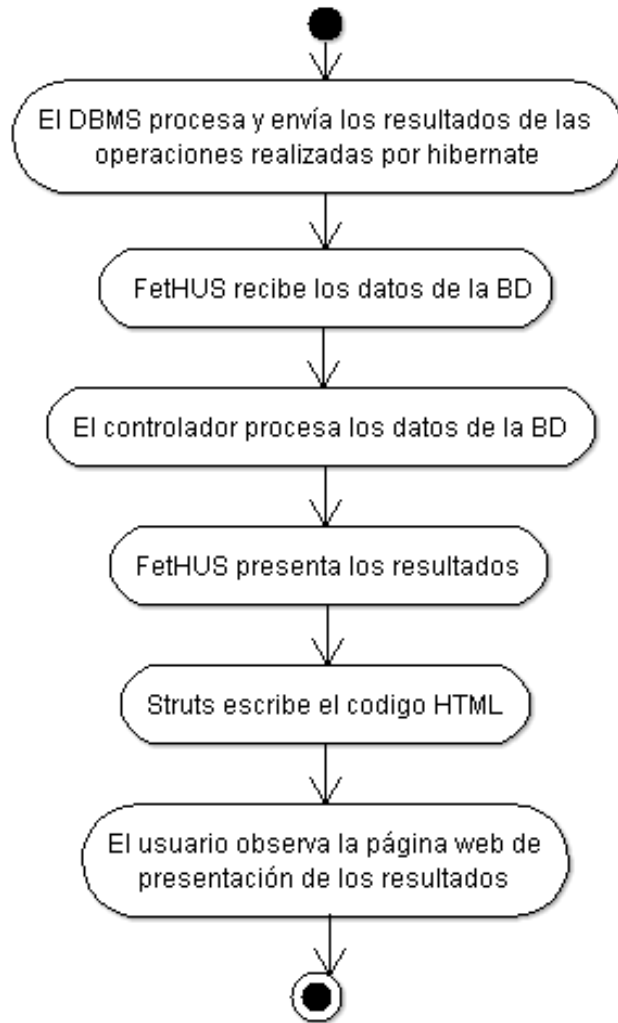


Figura 22. Diagrama de actividades que muestra el proceso de salida de los resultados hacia el cliente, efectuado por FETHUS, desde la consulta en la base de datos hasta la presentación final.

#### **7.2.4. DIAGRAMA DE PAQUETES**

En el diagrama de paquetes mostrado a continuación, se muestra claramente la estructura general de las capas en FETHUS, pudiéndose ver el uso del patrón MVC y la dependencia entre las diferentes capas que conforman el sistema.

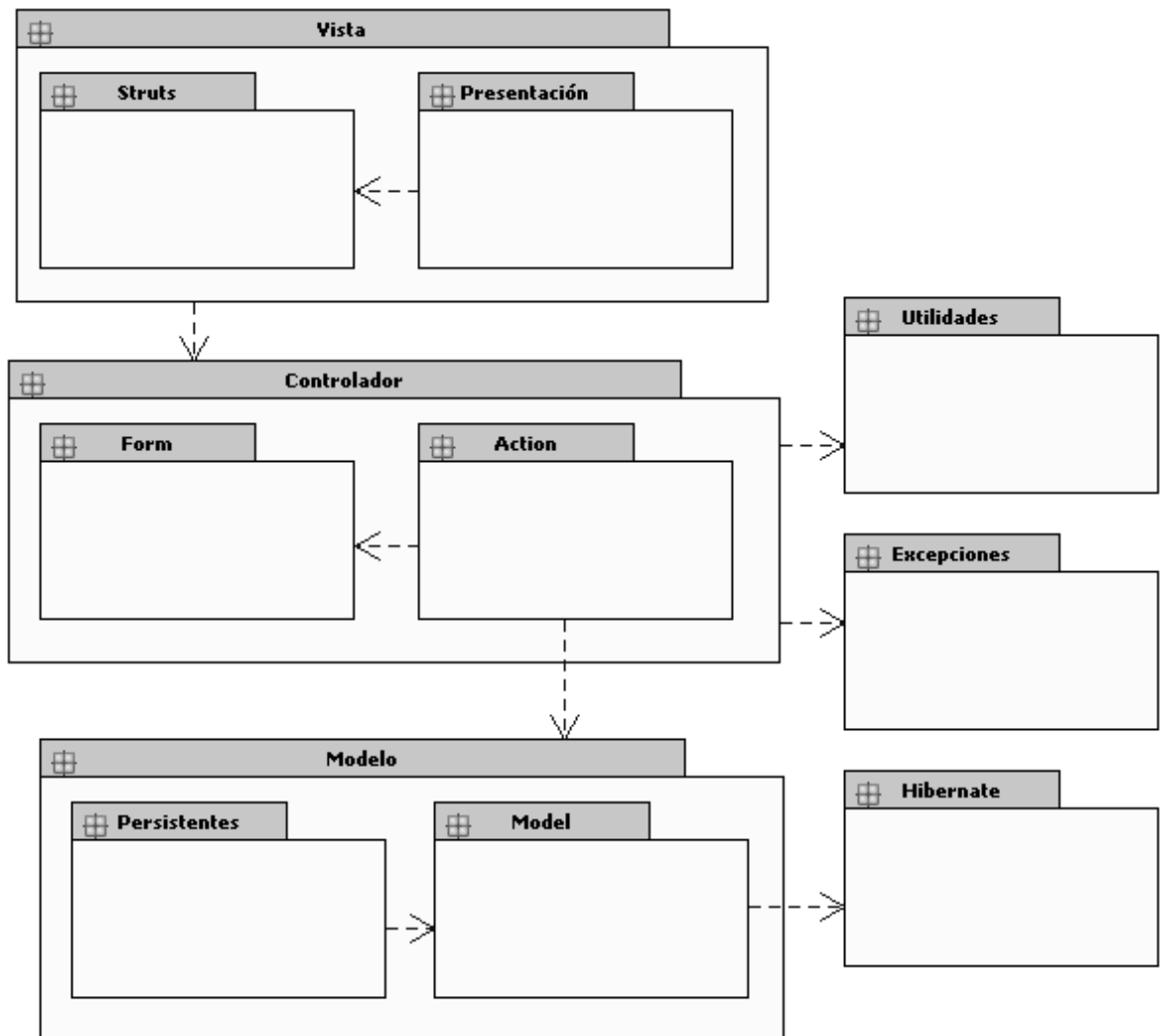


Figura 23. Diagrama de paquetes de FETHUS.

### 7.2.5. DIAGRAMA ENTIDAD-RELACIÓN

Aunque el diagrama entidad-relación no forma parte de la documentación UML, si es fundamental a la hora del diseño de cualquier sistema de información. La estructura de diseño de la base de datos de FETHUS está ilustrada en el diagrama entidad-relación que se muestra a continuación.

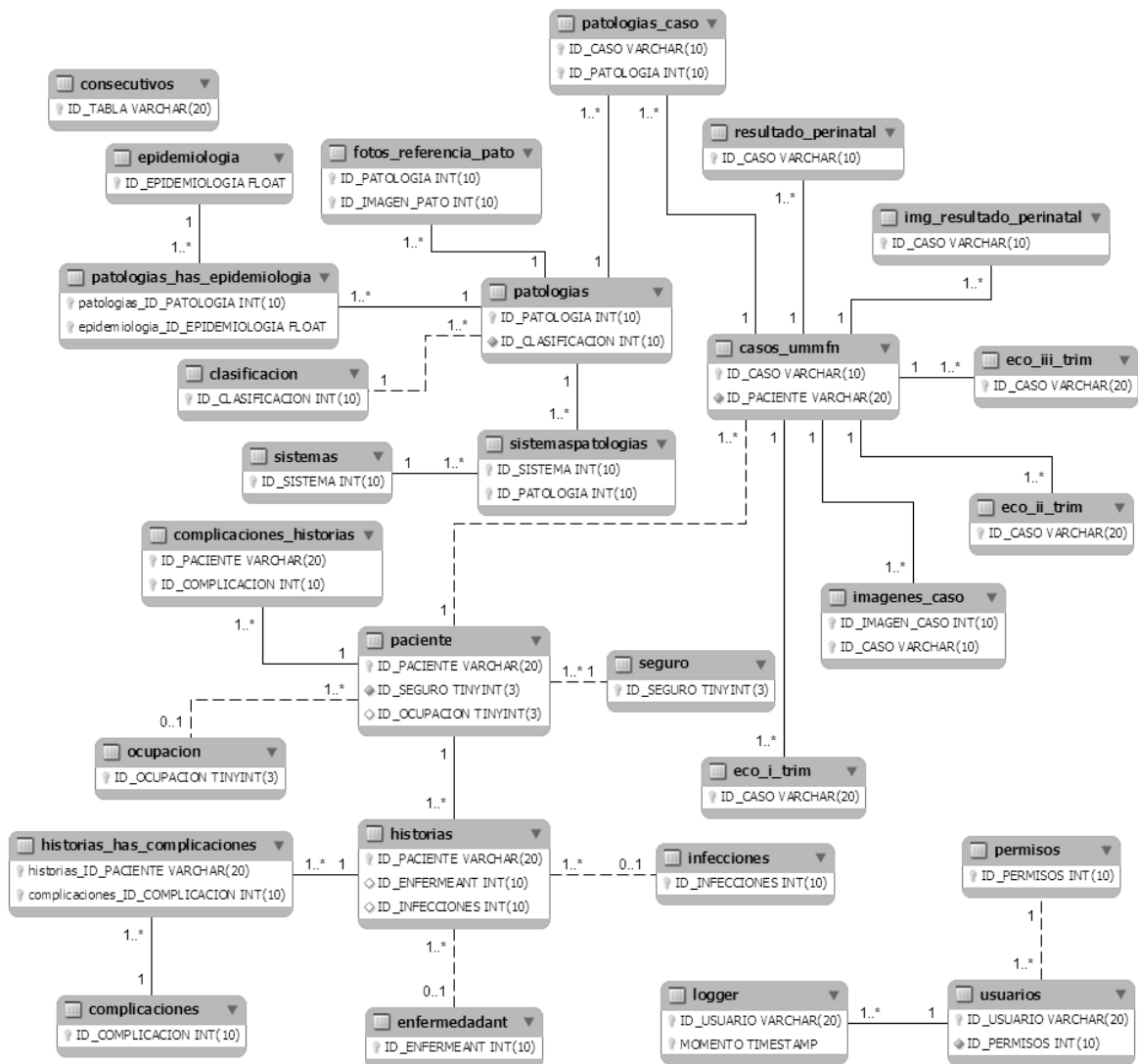


Figura 24. Diseño físico de la base de datos, presentado en forma de diagrama referencial.

En la Figura 24, se pueden observar todas las tablas que componen la base de datos de FETHUS, en donde se visualizan solamente las llaves primarias y foráneas debido a que la mayoría de tablas cuentan con gran cantidad de campos. También se pueden ver las relaciones entre las entidades.

## 7.2.6. DIAGRAMA DE CLASES

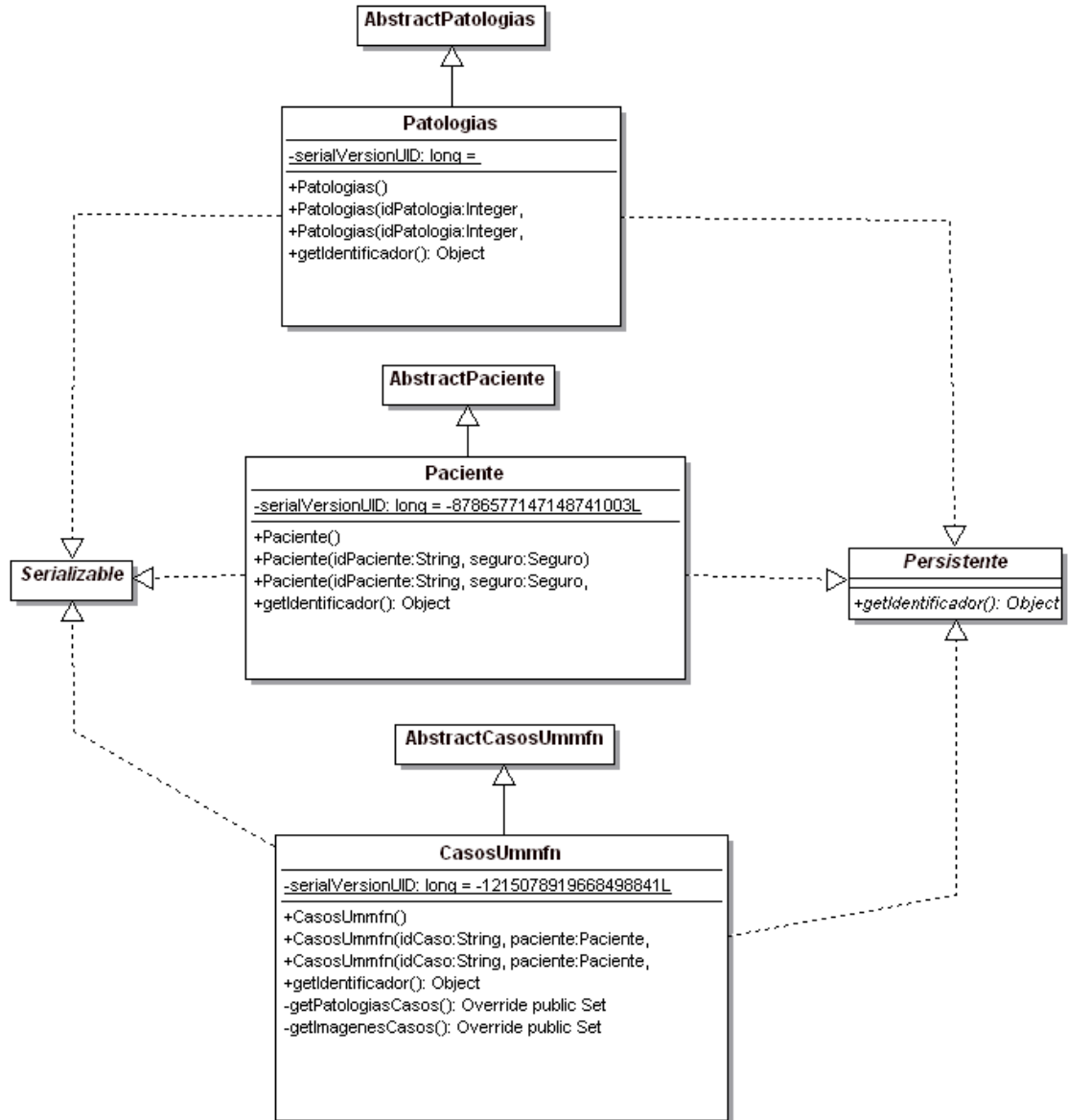


Figura 25. Diagrama de clases general.



### **7.3. IMPLEMENTACIÓN**

FETHUS es un producto independiente y completamente autónomo, es decir, no forma parte de ningún sistema mayor ni utiliza funcionalidades de sistemas externos. Se han utilizado diferentes componentes complementarios para la ayuda en el desarrollo como lo son ciertas librerías, o clases de validación y presentación que contribuyen a un desarrollo más rápido del sistema.

El desarrollo de la herramienta FETHUS, se ha cimentado sobre el paradigma de la programación orientada a objetos, ya que aunque FETHUS sea un sistema basado en web, el hecho de que se utilice la tecnología J2EE y el framework Struts implica la obtención de un entorno de trabajo mucho más poderoso y segmentado, en el cual es posible la especialización de roles y se facilita la reutilización y la robustez de código.

Struts posee compatibilidad con todas las plataformas disponibles de *Java Enterprise Edition*. La característica más importante y llamativa de este framework es que se basa en el patrón Modelo-Vista-Control, el cual provee un conjunto de clases y *Tag-Libs* que conforman el controlador, facilitando la integración con el modelo y la construcción de vistas. A continuación se explican estas categorías y su utilización en FETHUS:

- Modelo: Tiene como función manejar la lógica del negocio que responde a las peticiones. Una Clase mapeada se denomina Persistente, y un objeto de tipo persistente puede manipularse fácilmente, pudiéndose ingresar y extraer mediante getters y setters genéricos proporcionados por el framework de tipo ORM, ver Figura 27.

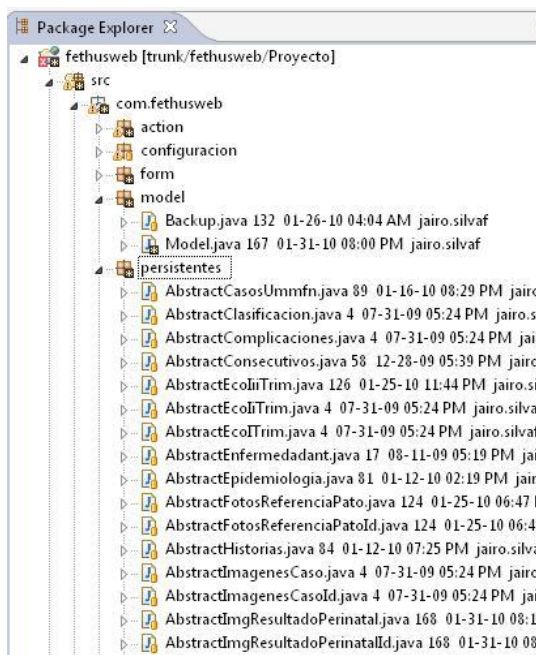


Figura 27. Captura del paquete persistentes y sus clases.

Las clases persistentes implementan la interfaz Persistente lo cual permite manipular a cualquier objeto llevándolo a la primitiva Object de java accediéndolo mediante su identificador.

```

/**
 * @author Jairo Alfonso Silva
 * metodo para actualizar y almacenar persistentes.
 * @param persistente
 */
public void almacenar(Persistente persistente) {
    Object objeto = this.getObjeto(persistente);
    if(objeto == null){ //insercion
        this.getSession().save(persistente);
    }
    else{ //actualizacion
        this.getSession().evict(objeto);
        this.getSession().update(persistente);
    }

    this.getSession().flush();
}

```

Figura 28. Método “almacenar” de la clase Model.

La codificación de los métodos para el mantenimiento de los persistentes (almacenar, eliminar, actualizar, obtener) es muy similar en todos los casos de uso de la aplicación FETHUS. Para tareas específicas Hibernate proporciona su propio lenguaje de consultas llamado HQL (*Hibernate Query Language*) el cual es el equivalente al SQL en el modelo de objetos. Cómo las relaciones entre tablas ya se encuentran definidas, al extraer un objeto se tiene acceso a todos los demás objetos relacionados en dicha clase.

```

public List<Persistente> getecosIiTrimestre(String idPaciente, String nombre, String apellido){

    List<Persistente> lista = new ArrayList<Persistente>();
    List<EcoIiTrim> ecografiasIt = null;
    idPaciente = Util.null2String(idPaciente);
    nombre = Util.null2String(nombre);
    apellido = Util.null2String(apellido);

    ecografiasIt = this.getSession().createCriteria(EcoIiTrim.class)
        .createAlias("casosUmmfn", "Caso")
        .createAlias("casosUmmfn.paciente", "Paciente")
        .add(Restrictions.ilike("Paciente.idPaciente", idPaciente, MatchMode.ANYWHERE))
        .add(Restrictions.ilike("Paciente.nombre", nombre, MatchMode.ANYWHERE))
        .add(Restrictions.ilike("Paciente.apellido", apellido, MatchMode.ANYWHERE))
        .addOrder(Order.asc("Paciente.apellido"))
        .addOrder(Order.asc("Paciente.nombre"))
        .list();

    for(EcoIiTrim ecografi : ecografiasIt)
        lista.add(ecografi);
    return lista;
}

```

Figura 29. Clase Criteria que especifica métodos para realizar operaciones lógicas en las consultas

- Vista: Es la encargada de generar las respuestas que deben ser enviadas al cliente. Durante el desarrollo de la presente herramienta web se han tenido que generar tanto respuestas que incluyen, como respuestas que no incluyen datos proporcionados por el controlador.

Las páginas JSP con las que interactúa el usuario del sistema, fueron escritas utilizando las Tag-Libs JSP de Struts, JSTL, HTML4 y JavaScript, brindando un código más ordenado y de fácil mantenimiento para el diseñador web que quiera modificar la interfaz gráfica del sistema.

La mayoría de validaciones se realizaron en el lado del cliente utilizando JavaScript para no sobrecargar el servidor. Si el usuario deshabilita el motor JavaScript de su navegador, no podrá siquiera enviar el formulario, y además todas las transacciones se especificaron con el método POST. Si de alguna forma se envía un dato incongruente con la base de datos, Hibernate lo detectará y realizará un rollback en lugar de commit.

- Controlador: Se puede decir que el controlador es el “cerebro” de la aplicación. Todas las peticiones a la capa intermedia que se realicen desde el cliente son dirigidas al controlador, quien determina las acciones a realizar para cada una de estas peticiones e invocar al resto de los componentes, coordinando así todo el proceso.

Las acciones y el comportamiento de la aplicación FETHUS se han implementado mediante Struts cuya principal función es controlar la interacción usuario-sistema. Para cumplir su objetivo, éste define sus propios elementos HTML y se encarga de relacionarlos con el FormBean utilizado para capturar y mostrar los datos en pantalla. Todo esto manteniendo separados de forma transparente la Interfaz Gráfica del Controlador, esto permite escribir las vistas sin necesidad de usar una sola línea de código Java, haciendo el código más ordenado y reutilizable.

En la Figura 30 se muestra el archivo XML que define el flujo de interacción de FETHUS. Se observa una captura del esquema de configuración de Struts en el cual se pueden identificar 3 elementos. Un primer Servlet encargado de preparar la vista para ser mostrada al usuario, en este se definen parámetros y validaciones cómo por ejemplo la sesión. La vista encargada de la entrada/salida en pantalla de la información y un segundo Servlet Modelo, encargado de construir los objetos a partir del formulario diligenciado por el usuario y redireccionar el flujo a una nueva vista o a otro controlador.

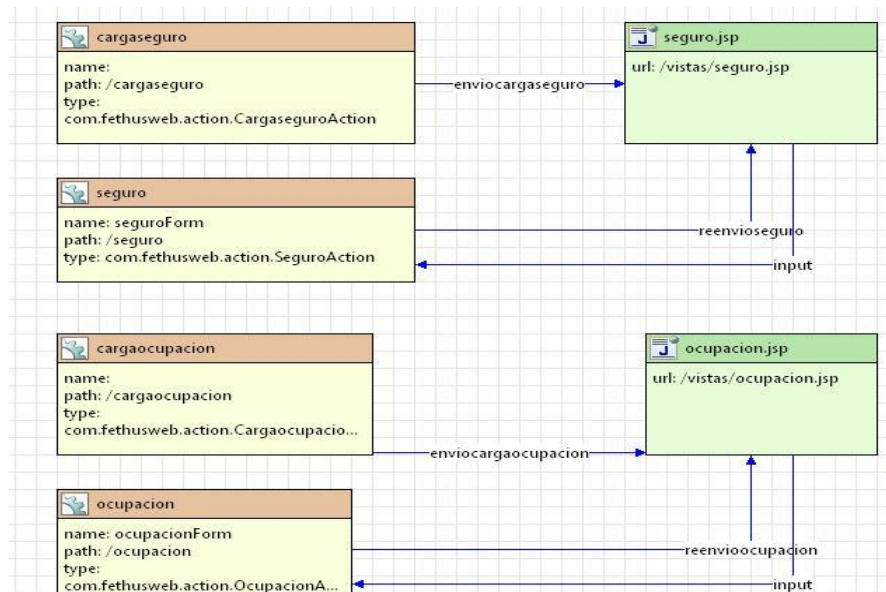


Figura 30. Archivo de configuración Struts-config.xml.

Debido a que el desarrollo de FETHUS ha sido de manera conjunta entre los autores, dividiendo la elaboración de tareas, se hace necesaria la utilización de una herramienta que solucione el problema que se genera cuando se desea unir los avances referentes a la programación del software que cada integrante ha realizado.

Tras la investigación sobre las herramientas existentes en el mercado que solucionan este tipo de problemas, los autores del presente proyecto han optado por la elección de Subversion<sup>27</sup>. Además el repositorio que se ha utilizado para el alojamiento de los archivos fuentes de FETHUS es googlecode<sup>28</sup>, gracias a que ofrece grandes ventajas para la administración y el hospedaje del proyecto. En la siguiente imagen se muestra una captura de pantalla de FETHUS alojado en el

<sup>27</sup> Software de sistema de control de versiones.

<sup>28</sup> Sitio de Google que permite alojar el código fuente de proyectos de software.



Los diferentes tipos de usuarios de FETHUS han corroborado el buen funcionamiento de los diferentes módulos del sistema, verificando la apropiada aceptación de los datos, el procesamiento, y la recuperación e implementación adecuada de las reglas del negocio. De esta manera se ha podido concluir que las pruebas funcionales efectuadas han sido satisfactorias y aprobadas.

### 7.4.2. PRUEBAS DE CARGA

Los autores del presente proyecto, han recurrido a la utilización de la herramienta llamada “ab benchmark” para verificar la capacidad y el rendimiento del sistema en ejecución. Con esta herramienta ha sido posible simular cargas pesadas en el servidor con una gran cantidad de tráfico en la red, simulando acciones de múltiples usuarios. La prueba ha sido realizada a la aplicación FETHUS la cual se encuentra alojada en el servidor del GIIB “<http://sis25.uis.edu.co>”, simulando la conexión de un máximo de 100 usuarios y un envío máximo de 10 peticiones, siendo el resultado el que se muestra tabulado a continuación:

Connection Times (ms)					
	Min	Mean[+]	Mean[-sd]	Median	Max
Connect	0	0	0	0	0
Processing	937	2210	744.4	2156	3203
Waiting	937	2210	744.2	2156	3203
Total	937	2210	744.2	2156	3203

Tabla 5. Tiempos de conexión a la aplicación.

Percentage of the requests served within a certain time (ms)	
Percentage	Time
50%	2156
66%	2562
75%	3171
80%	3187
90%	3187

95%	3203
98%	3203
99%	3203
100%	3203

Tabla 6. Intervalos de tiempo clasificados por porcentajes.

Activity	Results
Complete requests	100
Failed requests	0
Write errors	0
Non-2xx responses	100
Keep-Alive requests	100
Total transferred	30750 bytes
HTML transferred	0 bytes
Requests per second	18.08 [#/sec] (mean)
Time per request	2765.625 [ms] (mean)
Time per request	55.313 [ms] (across all concurrent requests)
Transfer rate	5.42 [Kbytes/sec] received

Tabla 7. Resultado final de la prueba de carga.

Como podemos observar en esta última tabla, todas las peticiones fueron exitosas con un promedio bastante bueno en tiempo de respuesta.

### **7.4.3. PRUEBAS DE SEGURIDAD**

El objetivo principal ha sido probar la vulnerabilidad del sistema frente a accesos o manipulaciones no autorizadas. Al igual que con las pruebas funcionales, el personal de la UMMFN ha auditado este tipo de prueba mediante el uso de FETHUS, dando por aprobada su funcionalidad. Con esto se da por garantizado que, con base en la seguridad planteada en los casos de uso modelados en el Documento de Especificación de Requisitos de Software, el sistema cumple satisfactoriamente con las siguientes características:

- Autenticación de usuarios.
- Manejo de sesiones.
- Control de acceso y autorización.
- Registro de eventos.
- Validación de entrada de datos.

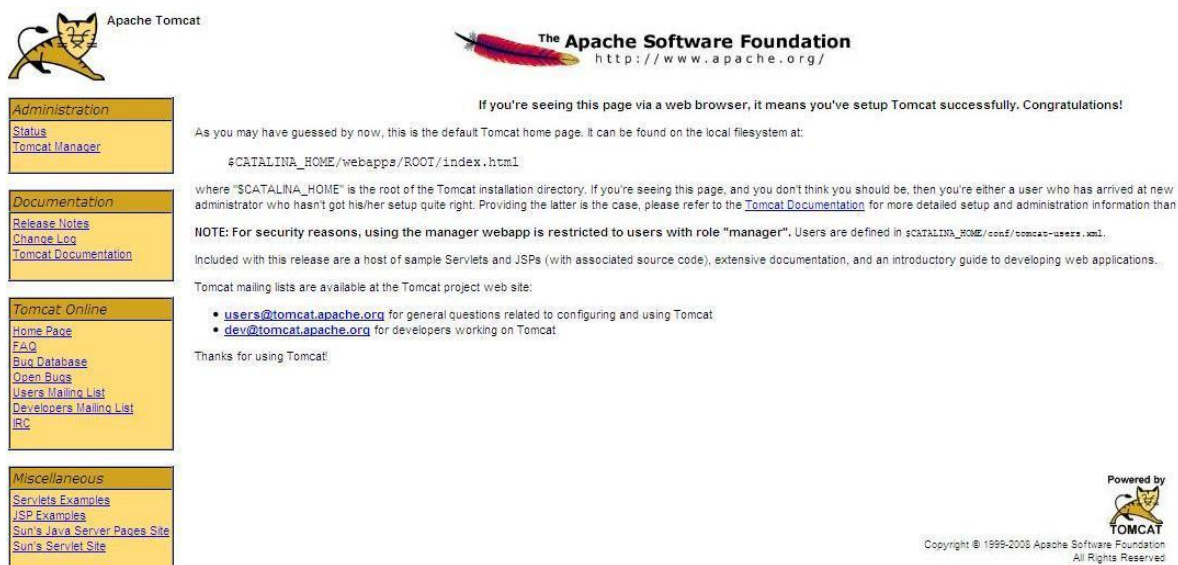
## 8. DOCUMENTACIÓN Y MANUALES

A lo largo del presente capítulo se presentan los diferentes tipos de manuales para que tanto desde los usuarios finales, hasta los desarrolladores de software tengan una guía con la que pueden profundizar en el uso y la construcción del sistema.

### 8.1. MANUAL DE INSTALACIÓN DE FETHUS

A continuación se muestran los pasos a seguir para instalar FETHUS en cualquier servidor, válidos desde que se cumplan los requisitos de hardware y software:

- Tener la base de datos previamente montada y configurada en el servidor.
- En un navegador web ingresar a la raíz del apache/tomcat mediante la url <http://localhost:8080/>.



Apache Tomcat

The Apache Software Foundation  
<http://www.apache.org/>

If you're seeing this page via a web browser, it means you've setup Tomcat successfully. Congratulations!

As you may have guessed by now, this is the default Tomcat home page. It can be found on the local filesystem at:

```
#CATALINA_HOME/webapps/ROOT/index.html
```

where "CATALINA\_HOME" is the root of the Tomcat installation directory. If you're seeing this page, and you don't think you should be, then you're either a user who has arrived at new administrator who hasn't got his/her setup quite right. Providing the latter is the case, please refer to the [Tomcat Documentation](#) for more detailed setup and administration information than

**NOTE: For security reasons, using the manager webapp is restricted to users with role "manager".** Users are defined in `CATALINA_HOME/conf/tomcat-users.xml`.

Included with this release are a host of sample Servlets and JSPs (with associated source code), extensive documentation, and an introductory guide to developing web applications.

Tomcat mailing lists are available at the Tomcat project web site:

- [users@tomcat.apache.org](mailto:users@tomcat.apache.org) for general questions related to configuring and using Tomcat
- [dev@tomcat.apache.org](mailto:dev@tomcat.apache.org) for developers working on Tomcat.

Thanks for using Tomcat!

Administration

- Status
- Tomcat Manager

Documentation

- Release Notes
- Change Log
- Tomcat Documentation

Tomcat Online

- Home Page
- FAQ
- Bug Database
- Open Bugs
- Users Mailing List
- Developers Mailing List
- IRC

Miscellaneous

- Servlets Examples
- JSP Examples
- Sun's Java Server Pages Site
- Sun's Servlet Site

Powered by  
TOMCAT

Copyright © 1999-2005 Apache Software Foundation  
All Rights Reserved

Figura 32. Captura del servicio de Tomcat funcionando.

- Ingresar al modulo de administración llamado “Tomcat Manager” y validarse como un usuario con permisos de administrador.

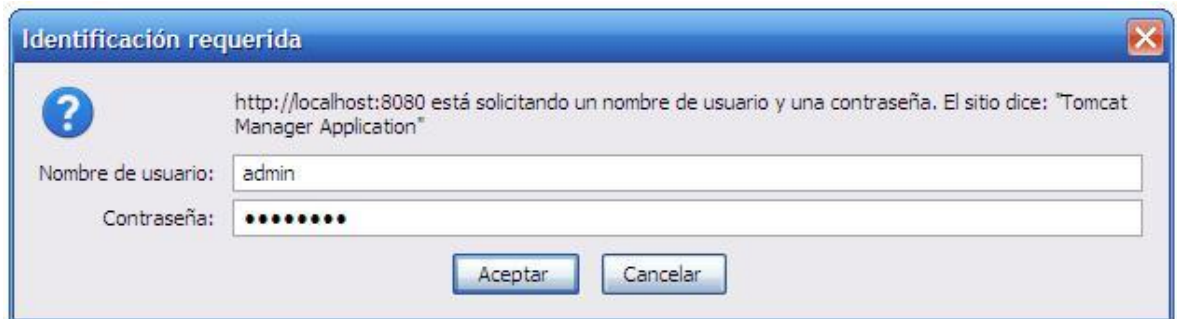


Figura 33. Captura del inicio de sesión en el “Tomcat Manager Application.”

- En la sección “Desplegar” más exactamente “Archivo WAR a desplegar”, seleccionar el archivo WAR a cargar y dar clic al botón llamado “Desplegar”.

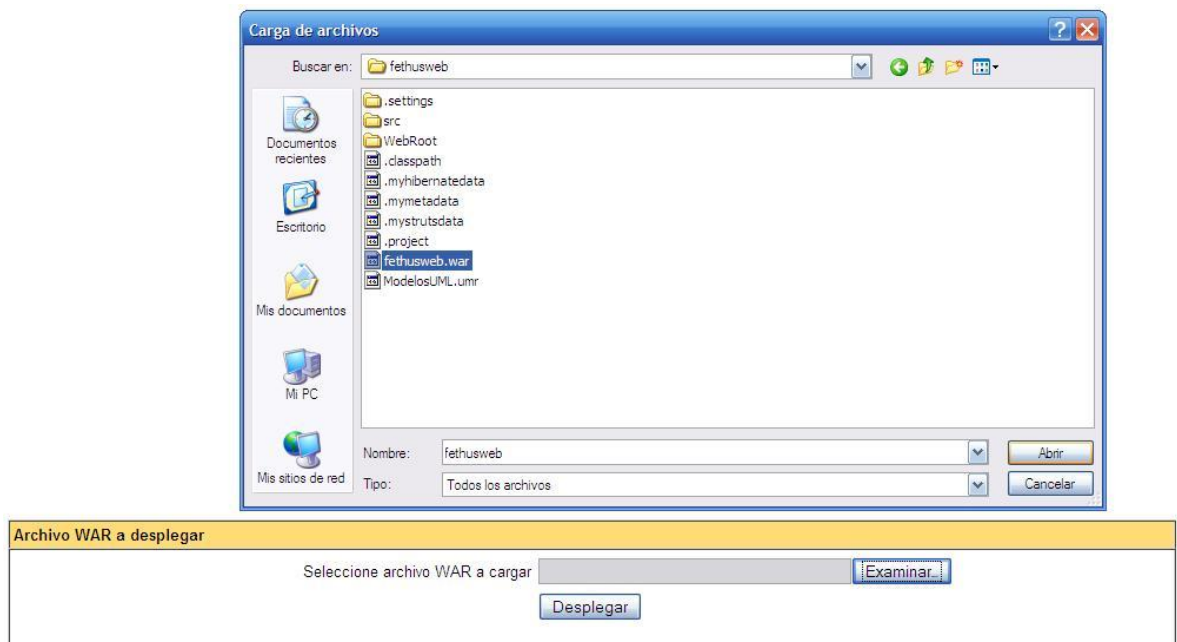


Figura 34. Captura de la carga del archivo WAR que contiene la Aplicación Web.

Siguiendo los pasos anteriormente mencionados la instalación del sistema será exitosa y el resultado de la carga de la herramienta se verá como la siguiente Figura.

Aplicaciones				
Trayectoria	Nombre a Mostrar	Ejecutándose	Sesiones	Comandos
/	Welcome to Tomcat	true	0	<a href="#">Arrancar</a> <a href="#">Parar</a> <a href="#">Recargar</a> <a href="#">Replegar</a> <input type="button" value="Expirar sesiones"/> sin trabajar x <input type="text" value="30"/> minutos
/docs	Tomcat Documentation	true	0	<a href="#">Arrancar</a> <a href="#">Parar</a> <a href="#">Recargar</a> <a href="#">Replegar</a> <input type="button" value="Expirar sesiones"/> sin trabajar x <input type="text" value="30"/> minutos
/fethusweb	m5pc	true	0	<a href="#">Arrancar</a> <a href="#">Parar</a> <a href="#">Recargar</a> <a href="#">Replegar</a> <input type="button" value="Expirar sesiones"/> sin trabajar x <input type="text" value="30"/> minutos
/host-manager	Tomcat Manager Application	true	0	<a href="#">Arrancar</a> <a href="#">Parar</a> <a href="#">Recargar</a> <a href="#">Replegar</a> <input type="button" value="Expirar sesiones"/> sin trabajar x <input type="text" value="30"/> minutos
/manager	Tomcat Manager Application	true	0	<a href="#">Arrancar</a> <a href="#">Parar</a> <a href="#">Recargar</a> <a href="#">Replegar</a> <input type="button" value="Expirar sesiones"/> sin trabajar x <input type="text" value="30"/> minutos

Figura 35. Captura de la Aplicación Web cargada en el servidor.

Es de vital importancia que al momento de realizar alguna actualización del sistema, se haga una copia de seguridad de la carpeta en la cual se encuentran las imágenes que han sido cargadas por el root o el residente, dichas imágenes se encuentran en el directorio “raíz\_servidor\_tomcat/webapps/fethusweb/fotos/”, donde “raíz\_servidor\_tomcat” varía dependiendo del sistema operativo instalado en el servidor. Una vez realizada la actualización se procede a retornar la carpeta de fotos a su directorio original.

## 8.2. MANUALES DE USUARIO Y DE ADMINISTRADOR

Los manuales de usuario y administrador de FETHUS están elaborados en formato de película SWF<sup>29</sup> o Flash y se accede de forma autónoma o mediante el uso del vínculo “ayuda” ubicado en la presentación gráfica de FETHUS. Estos manuales ilustran de forma gráfica y animada la operación normal de las funciones más complicadas del sistema, de modo que el usuario sepa exactamente qué

<sup>29</sup> Formato de archivo de gráficos vectoriales.

hacer en cada caso. A continuación se presenta una captura de pantalla del video realizado como manual de usuario el cual muestra las funcionalidades y guía al cliente en el uso de FETHUS.



Figura 36. Captura del video elaborado como manual de usuario y administrador.

### **8.3. MANUAL DEL PROGRAMADOR**

El manual del programador tiene como propósito documentar al lector sobre la función que cumple cada una de las clases y explicar los métodos más importantes (con sus entradas y salidas), atributos y paquetes que fueron desarrollados para construir FETHUS, con el fin del que el usuario que desee realizarle mantenimiento al sistema pueda modificar a su gusto los valores y parámetros que crea convenientes para darle continuidad al desarrollo de la presente herramienta. Dicho manual se encuentra en formato digital adjunto al software dentro del CD que ha sido entregado a biblioteca.

## **9. CONCLUSIONES**

Se ha desarrollado un sistema de información robusto, funcional y modular pasando por las diferentes etapas de análisis, diseño e implementación, con base en el documento de Especificación de Requisitos de Software previamente aprobado por el Comité de Sistemas de Información.

La propuesta y elaboración de proyectos interdisciplinarios, por parte de los grupos de investigación y los estudiantes de la escuela de Ingeniería de Sistemas con la colaboración de profesionales de diferentes áreas, como los docentes y personal de la Unidad de Medicina MaternoFetal y Neonatal del Hospital Universitario de Santander, conlleva al fortalecimiento y creación de acuerdos para el planteamiento de nuevos trabajos más ambiciosos que fortalecen la imagen institucional de la UIS y la escuela de Ingeniería de Sistemas e Informática.

El producto final desarrollado es un sistema altamente parametrizable y adaptable a otras áreas de la medicina, no sólo en el campo de la perinatología, ya que la mayoría de funciones implementadas son genéricas y se adaptan a diferentes tipos de objetos.

La herramienta desarrollada se convertirá apoyo a estudios investigativos y de soporte a la academia, ya que el sistema será utilizado como fuente de consulta por parte de la comunidad académica interesada. Esto es posible gracias a que la información referente a perinatología (fotografías y texto) estará disponible para la consulta en línea a través de Internet.

El personal de la Unidad de Medicina MaternoFetal y Neonatal podrá brindar un servicio de mayor calidad a sus pacientes, debido a que el sistema cuenta con un control y seguimiento a la información de éstos. Para ejercer dicho control el sistema ofrece la característica de generación de reportes.

Se ha cumplido con la construcción de los requisitos de la interfaz de usuario, de la interfaz de software, de rendimiento y de diseño y desarrollo, estipulados en el documento de Especificación de Requisitos de Software que cimienta la metodología de Proceso Unificado Racional y se basa en el Lenguaje Unificado de Modelado, consolidándose en una herramienta metodológica estándar para el desarrollo de software.

La utilización de herramientas de carácter libre en proyectos de gran envergadura es una realidad viviente, ya que actualmente existe toda una comunidad de desarrolladores y de grandes empresas que soportan y respaldan los diferentes proyectos, convirtiéndolos en verdaderas alternativas lo suficientemente robustas y competitivas en el mercado.

## **10. RECOMENDACIONES**

Es importante darle continuidad al desarrollo del sistema de información FETHUS mediante una segunda fase, agregando nuevos módulos y funcionalidades como lo son la transmisión de las sesiones ecográficas en tiempo real a través de internet, el alojamiento de videos previamente grabados por el personal de la UMMFN y la implementación de un nuevo módulo que realice análisis estadísticos complejos con la información de la base de datos del sistema.

Se recomienda la utilización del paradigma de la programación orientada a objetos y continuar con el estilo del patrón Modelo Vista Controlador (MVC), para que a su vez sea más fácil la realización del mantenimiento al sistema.

La persona encargada en administrar el sitio deberá tener caución con los privilegios de los usuarios del sistema. Aunque también podrá aprovechar la delegación parcial de privilegios, para que el mantenimiento del sitio no recaiga únicamente sobre una persona.

Se aconseja la ejecución periódica de copias de seguridad de la base de datos, puesto que ésta almacena información importante concerniente con los pacientes y las diferentes patologías que se presentan en la Unidad de Medicina MaternoFetal, que a futuro será empleada para la realización de estudios estadísticos en nuestra región.

Se recomienda cuidado especialmente con las cuentas y contraseñas FTP del servidor en el cual se va alojar el sitio, ya que esta es más importante incluso que la contraseña de administrador del sistema de información.

Se sugiere la proposición de más proyectos investigativos, de carácter asociativo entre el Grupo de Investigación en Ingeniería Biomédica y el Hospital Universitario de Santander, porque dichas entidades actualmente cuentan con la infraestructura

necesaria para llevar a cabo trabajos de grado relacionados con la web y dispositivos móviles.

Se recomienda que el servidor en el cual se va a instalar FETHUS sea de alta disponibilidad, puesto que lo ideal es que el servicio que brinda la herramienta esté los siete días a la semana, veinticuatro horas del día sin interrupciones.

## **11. BIBLIOGRAFÍA**

CÁRCAMO SEPÚLVEDA, José. Bases de Datos Relacionales: Un Enfoque Práctico de Diseño. Bucaramanga: Ediciones UIS, 1996.

DURAN TORO, Amador y BERNÁRDEZ JIMÉNEZ, Beatriz. Metodología para elicitación de requisitos software. Versión 2.1. Sevilla: 2000. Universidad de Sevilla. Facultad de Informática y Estadística. Departamento de Lenguajes y Sistemas Informáticos.

IEEE. IEEE Recommended Practice for Software Requirements Specifications. IEEE/ANSI Standard 830–1998, Institute of Electrical and Electronics Engineers, 1998. Es un estándar con recomendaciones para la especificación de requisitos software.

RUMBAUGH, James; JACOBSON, Ivar y BOOCH, Grady. El Lenguaje Unificado de Modelado. Madrid: Addison Wesley, 1999.

Pressman, Roger, Ingeniería del Software: Un Enfoque Práctico. 5 ed. Madrid: McGraw-Hill, 2002.

WIKIMEDIA FOUNDATION, INC. Wikipedia, la enciclopedia libre. Disponible en: <<http://es.wikipedia.org/>>

ACOG Committee opinion. Perinatal and infant mortality statistics. Number 167, December 1995. Committee on Obstetric Practice. American College of Obstetricians and Gynecologists. Int J Gynaecol Obstet 1996; 53:86.

Carey, CM, Tullous, MW, Walker, ML. Hydrocephalus: Etiology, Pathologic Effects, Diagnosis, and Natural History. In: Pediatric Neurosurgery, 3rd ed, Cheek, WR (Ed), WB Saunders Company, Philadelphia 1994.

Martín Sierra Antonio J. STRUTS. Primera Edición. Alfaomega Grupo Editor ISBN: 978-970-15-1414-6

National Center for Health Statistics. [www.cdc.gov/nchs/](http://www.cdc.gov/nchs/) (Accessed on September 26, 2006).

KielyJL et al. Fetal death during labor: an epidemiologic indicator of level of obstetric care. *American Journal of Obstetrics and Gynecology*, 1985, 153(7):721-727.

Wilkins, E Weisman, Kim: Perinatal mortality; UpToDate DB. version 17.1:January 2009

## **ANEXO A. DOCUMENTO DE ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE.**

El presente documento concerniente a la Especificación de requisitos Software (ERS) del Sistema de Información FETHUS, se ha estructurado siguiendo las recomendaciones formuladas por el estándar IEEE<sup>30</sup> y utilizando la metodología para elicitación de requisitos software. Fue realizado con base en la información recolectada en las reuniones periódicas llevadas a cabo por el comité de sistemas de información durante la fase de análisis.

### ***INTRODUCCIÓN***

#### ***PROPÓSITO***

El propósito es definir cuáles son los requisitos que debe tener un sistema de información que gestione la información sobre malformaciones congénitas de la Unidad de Medicina MaternoFetal del Hospital Universitario de Santander. Además, se utilizará para determinar el avance del proyecto y como documento de respaldo en el momento de entregar el producto terminado.

El documento va dirigido al comité de sistemas de información de la Unidad de Medicina MaternoFetal y alcanzará su aprobación hasta que todos los integrantes del comité lleguen a un consenso sobre su contenido.

#### ***PARTICIPANTES***

El Comité de Sistemas de Información está conformado por los siguientes integrantes:

---

<sup>30</sup> IEEE. IEEE Recommended Practice for Software Requirements Specifications. IEEE/ANSI Standard 830–1998, Institute of Electrical and Electronics Engineers, 1993.

- Jefe de la Unidad de Medicina MaternoFetal y Neonatal.

**Dr. Carlos Hernán Becerra, Md GO Perinatólogo.**

- Médico Especialista Unidad de Medicina MaternoFetal y Neonatal.

**Dra. Mónica Andrea Beltrán, Md. GO MMF**

- Médico Especialista Unidad de Medicina MaternoFetal y Neonatal.

**Dr. Juan Carlos Otero, Md. GO Perinatólogo.**

- Md Residente de Ginecología II año UMMFN - HUS

**Dr. Orian Fontalvo F.**

- Gerente de proyecto.

**M.I.I.C.C. Víctor Eduardo Martínez A.**

- Estudiantes de la escuela de ingeniería de sistemas e informática, autores del proyecto FETHUS y del presente documento:

**Jairo Alfonso Silva Franco**

**Raúl Fernando Rivero Olarte**

### ***DEFINICIÓN DEL PROBLEMA***

Como resultado de las reuniones del comité y de la visita a las instalaciones de la Unidad, en donde se observó la operación cotidiana de interacción médico-paciente en circunstancias normales, se llegó a la descripción del escenario problema y las funciones que viene desempeñando actualmente, la cuales fueron consignadas en el documento de Especificación de Requisitos de Software (ERS) versión 1.0 que fue revisado por el comité de Sistemas de Información. Al mismo tiempo se pudieron identificar los siguientes problemas:

- **No existe una herramienta para la vigilancia epidemiológica.**
- **Es muy difícil consultar información acerca de una patología que ya se haya tratado.**
- **No existen copias de seguridad de los registros ecográficos. La capacidad de hardware y software del ecógrafo es muy limitada.**
- **No hay orden en la forma como se organizan los casos y la demanda es muy elevada.**
- **No existe una herramienta de referencia rápida que permita comparar imágenes.**

En conclusión, el presente proyecto informático se encargará de construir una herramienta software que permitirá solucionar los problemas de consulta de información acerca de patologías de la UMMFN, realización de informes estadísticos, del cual el sistema será parte integral y contribuirá a su correcta administración y mantenimiento del material fotográfico, e informativo.

### ***ALCANCE DEL SISTEMA***

El sistema de información FETHUS WEB, se presenta como una herramienta para la organización, centralización y soporte en la toma de decisiones en la Unidad de Medicina Materno Fetal del HUS, y que además, actúe como centro de consulta para los estudiantes de la UIS y residentes del Hospital Universitario de Santander.

El software se ocupa principalmente de administrar la información referente a los casos de patologías, anormalidades o malformaciones que se presentan durante el embarazo, con el fin de realizar un análisis estadístico-epidemiológico, que brinde soporte a las decisiones de la Junta Médica suscrita a la UMMFN y también sea base informativa para líneas de investigación. Al finalizar la implantación de este proyecto se esperan los siguientes beneficios:

- Organización y seguimiento sistematizado de los casos objeto de estudio de la Junta Médica de la UMMFN: El personal médico contará con una herramienta que le permita gestionar los casos, mantener el registro (inventario), actualizarlos, generar Informes estadísticos en el muy corto plazo, y dar referencias para el análisis de los futuros casos de estudio.
- Existencia de un centro de consulta de patologías perinatales: La comunidad educativa podrá consultar no solo la información bibliográfica de una patología específica, sino también los casos reales sobre estas patologías que se hayan presentado a nivel local comparando imágenes reales con imágenes de referencia.
- Existencia de un centro de estadística epidemiológica y mortalidad perinatal del Hospital Universitario de Santander.
- Mejores prácticas de clase.

## ***DESCRIPCIÓN GLOBAL DEL PRODUCTO A DESARROLLAR***

### ***PERSPECTIVA DEL PRODUCTO.***

El sistema de Información FETHUS WEB 1.0 es un producto software independiente, autónomo desarrollado desde el análisis, utilizando herramientas libres. No hace parte de otros S.I aunque está diseñado para ser una herramienta escalable y fácilmente integrable a otros sistemas gracias al modelo de datos y la amplia documentación existente acerca de su arquitectura.

La mayor parte de la información administrada por FETHUS WEB es de carácter confidencial, y para uso exclusivo de las entidades y personas suscritas a la Unidad de Medicina Materno Fetal del Hospital Universitario de Santander. Sin embargo los administradores del sistema pueden asignar cuentas de usuario a las personas o entidades que se definan como usuarios de cada nivel.

La interfaz de FETHUS WEB debería ser amigable, intuitiva y facilitar la familiarización del usuario con el sistema. Debe contar con un módulo de ayuda para cada uno de los procesos disponibles al usuario.

***FUNCIONES DEL PRODUCTO.***

- Publicar en internet a los estudiantes, mediante un sistema de consulta en línea, la información clínica, fotográfica y patológica disponible para cada malformación.
- Almacenar información concluyente sobre los casos dados de alta.
- Mostrar reportes sobre el estado actual de los pacientes en estudio.
- Imprimir reportes, catálogos y rótulos de casos de anomalías perinatales.
- Mostrar ayuda para todas las funciones del producto.

***CARACTERÍSTICAS DE USUARIO.***

Los usuarios en general deberán tener conocimientos en perinatología, obstetricia e interpretación de Ultrasonidos (manejo del ecógrafo).

***OBJETIVOS DEL SISTEMA***

<b>OBJ-01</b>	<b>Gestionar los casos de anomalías perinatales de la UMMFN</b>
<b>Descripción</b>	El sistema deberá gestionar los casos de anomalías perinatales detectados en la UMMFN: registrar, editar, consultar y eliminar casos.
<b>Importancia</b>	Esencial

<b>Estado</b>	Estable
---------------	---------

<b>OBJ-02</b>	<b>Gestionar los pacientes</b>
<b>Descripción</b>	El sistema deberá gestionar toda la información referente a los pacientes que presenten algún tipo de anormalidad o patología: registrar, editar, consultar y eliminar pacientes.
<b>Importancia</b>	Esencial
<b>Estado</b>	Estable

<b>OBJ-03</b>	<b>Gestionar patologías generales</b>
<b>Descripción</b>	El sistema deberá gestionar toda la información referente a las patologías tales como incidencia, características patológicas, características ecográficas, estudios complementarios, pronóstico y tratamiento.
<b>Importancia</b>	Esencial
<b>Estado</b>	Estable

<b>OBJ-04</b>	<b>Gestionar imágenes</b>
<b>Descripción</b>	El sistema deberá gestionar el repositorio de imágenes mediante una interfaz de usuario amigable, que permita la administración de imágenes presentadas en los pacientes de la UMMFN y de imágenes de referencia.

<b>Importancia</b>	Esencial
<b>Estado</b>	Estable

<b>OBJ-05</b>	<b>Generar informes estadísticos</b>
<b>Descripción</b>	El sistema deberá generar diferentes informes estadísticos a solicitud del usuario, respecto a las variables de cada caso y/o a las características demográficas del paciente.
<b>Importancia</b>	Esencial
<b>Estado</b>	Estable

<b>OBJ-06</b>	<b>Administrar el acceso y los privilegios de los usuarios</b>
<b>Descripción</b>	El sistema deberá establecer privilegios para acceder a las diferentes funcionalidades y los diferentes módulos.
<b>Importancia</b>	Esencial
<b>Estado</b>	Estable

<b>OBJ-07</b>	<b>Proporcionar ayuda</b>
<b>Descripción</b>	El sistema deberá proporcionar ayuda de todas las funciones a las que tenga acceso el usuario.

<b>Importancia</b>	Normal
<b>Estado</b>	Estable

***REQUISITOS DE ALMACENAMIENTO DE INFORMACIÓN***

<b>RI-01</b>	<b>Información sobre anomalías perinatales de la UMMFN</b>
<b>Objetivos asociados</b>	OBJ-01 Gestionar los casos de anomalías perinatales de la UMMFN OBJ-05 Generar Informes estadísticos
<b>Requisitos asociados</b>	RF-09 Registro de casos de la UMMFN RF-13 Modificar casos de la UMMFN RF-14 Actualizar casos de la UMMFN RF-16 Dar de alta casos de la UMMFN RF-17 Registrar conclusiones del caso de la UMMFN
<b>Descripción</b>	El sistema deberá almacenar la información referente a las anomalías que se presenten en la UMMFN.
<b>Datos específicos</b>	Información personal del paciente Antecedentes Enfermedades del caso Registro de Sesiones ecográficas Conclusiones del caso (al darse de alta)

<b>Intervalo temporal</b>	Pasado y presente.
---------------------------	--------------------

<b>RI-02</b>	<b>Información sobre patologías generales</b>
<b>Objetivos asociados</b>	OBJ-03 Gestionar patologías generales
<b>Requisitos asociados</b>	RF-18 Registrar patologías generales RF-19 Actualizar patologías generales RF-20 Consultar patologías generales RF-21 Eliminar patologías generales
<b>Descripción</b>	El sistema deberá almacenar información referente a las patologías existentes en forma general.
<b>Datos específicos</b>	Nombre de la patología Descripción larga Descripción corta Imágenes de referencia Información epidemiológica Bibliografía
<b>Intervalo temporal</b>	Presente

<b>RI-03</b>	<b>Información sobre usuarios del sistema</b>
<b>Objetivos asociados</b>	OBJ-06 Administrar el acceso y los privilegios de los usuarios
<b>Requisitos asociados</b>	RF-01 Crear cuentas de usuario RF-02 Asignar permisos RF-03 Modificar cuentas de usuario RF-04 Eliminar cuentas de usuario RF-05 Listar usuarios RF-07 Iniciar sesión de usuario RF-08 Modificar datos personales.
<b>Descripción</b>	El sistema deberá almacenar la información personal de los usuarios que harán uso del sistema.
<b>Datos específicos</b>	Nombres y apellidos Correo electrónico Login y password Tipo de usuario
<b>Intervalo temporal</b>	Presente

<b>RI-04</b>	<b>Banco de Imágenes</b>
--------------	--------------------------

<b>Objetivos asociados</b>	<p>OBJ-01 Gestionar los casos de anomalías de la UMMFN</p> <p>OBJ-03 Gestionar patologías generales</p> <p>OBJ-04 Gestionar Imágenes</p>
<b>Requisitos asociados</b>	<p>RF-15 Registro de información de la sesión ecográfica</p> <p>RF-09 Registrar casos de la UMMFN</p> <p>RF-18 Registrar patologías generales</p>
<b>Descripción</b>	<p>El sistema deberá almacenar todas las imágenes de referencia de los casos locales y patologías generales.</p>
<b>Datos específicos</b>	<p>Caso asociado o Patología</p> <p>Nombre de archivo</p> <p>Descripción</p> <p>Tipo de imagen</p> <p>Url</p>
<b>Intervalo temporal</b>	<p>Pasado y presente.</p>

## REQUISITOS FUNCIONALES

### DIAGRAMAS DE CASOS DE USO

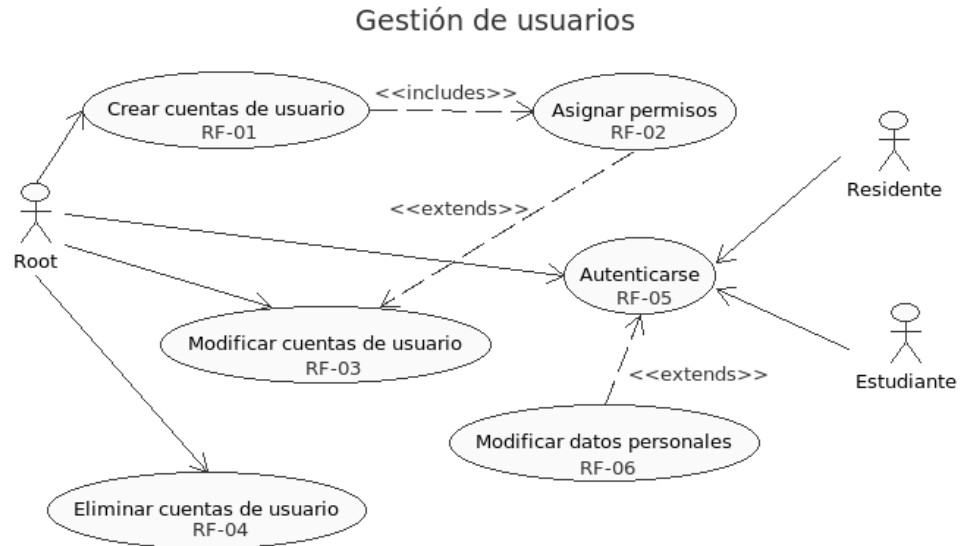


Figura 37. Diagrama de Casos de Uso para la Gestión de Usuarios.

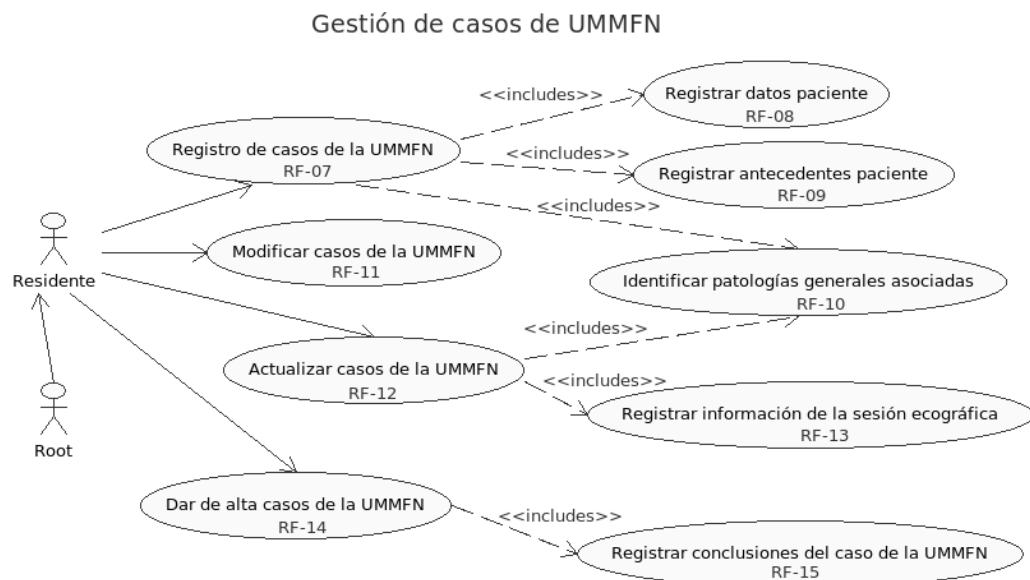


Figura 38. Diagrama de Casos de Uso para la Gestión de Casos.

## Gestión de patologías generales

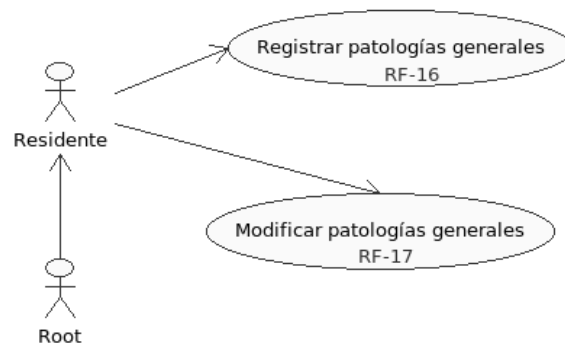


Figura 39. Diagrama de Casos de Uso para la Gestión de Patologías.

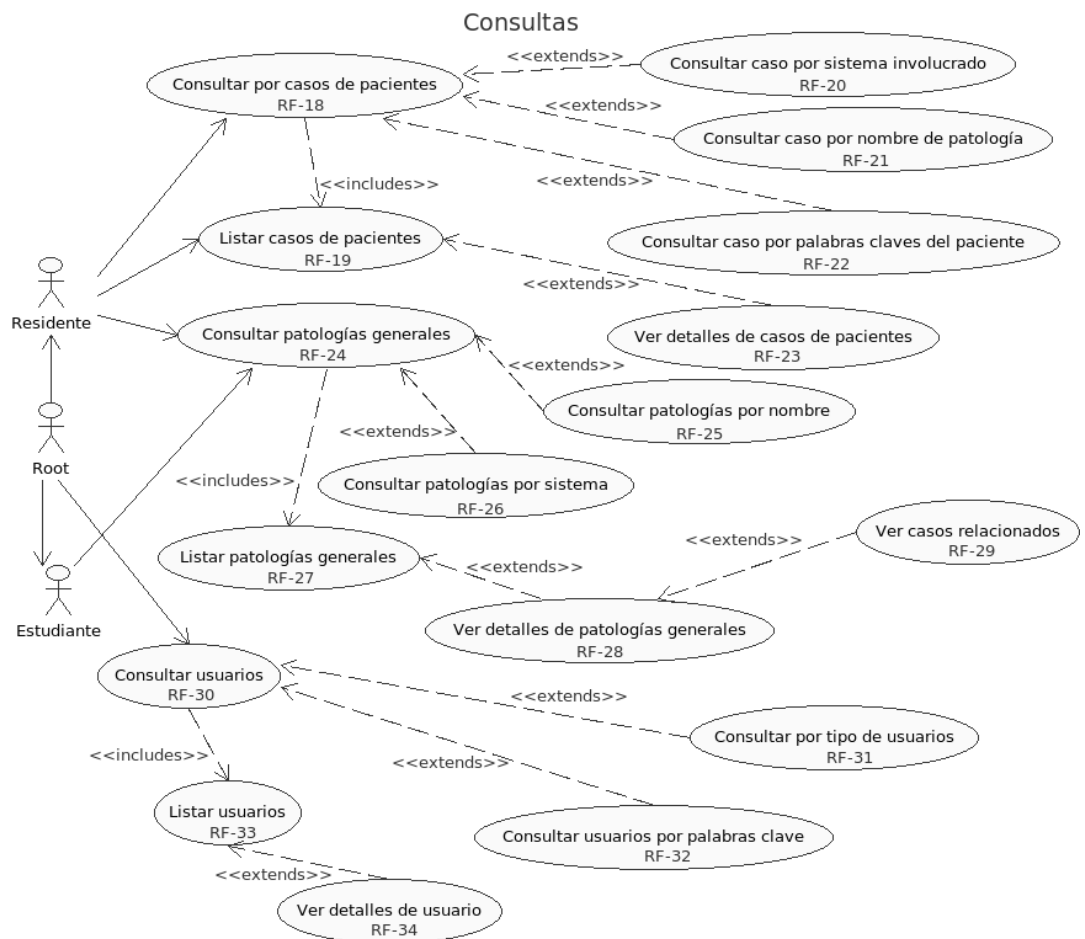


Figura 40. Diagrama de Casos de Uso para el módulo de Consultas.

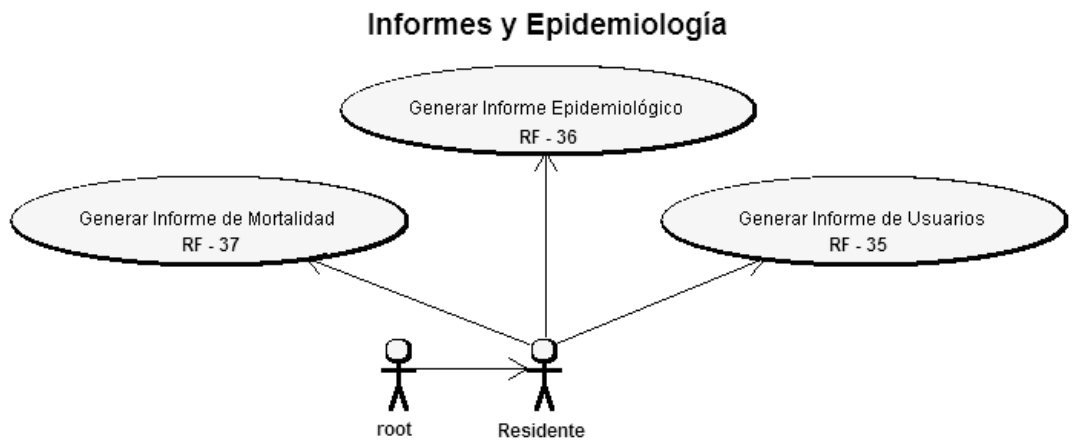


Figura 41. Diagrama de Casos de Uso para la generación de informes.

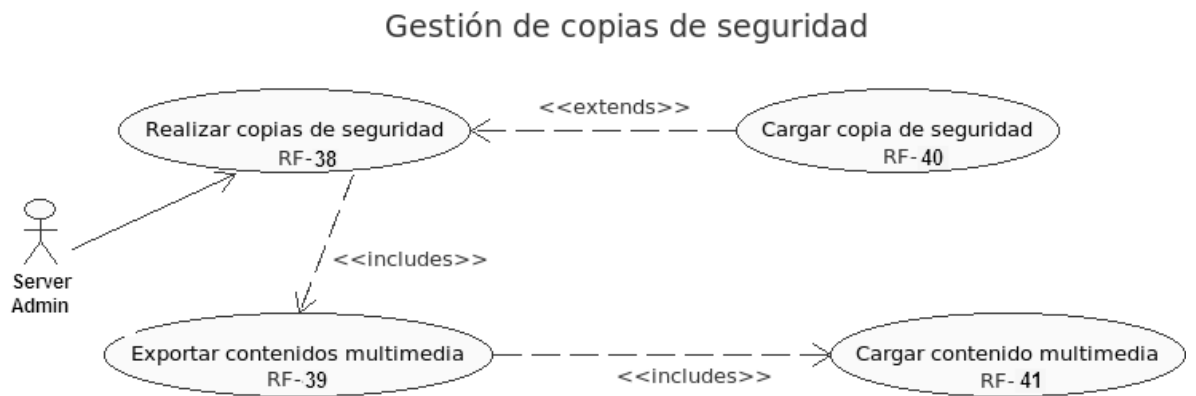


Figura 42. Diagrama de Casos de Uso para la Gestión de Copias de Seguridad.

### **DEFINICIÓN DE ACTORES**

<b>ACT-01</b>	<b>Root</b>
<b>Versión</b>	1.0
<b>Descripción</b>	Este actor representa al usuario del sistema que tiene privilegios para absolutamente todo, desde asignar privilegios y crear cuentas de usuario hasta registrar, modificar, actualizar y eliminar casos de anomalías en la UMMFN, y de patologías generales, generar informes, hacer consultas y copias de respaldo de la base de datos.

<b>ACT-02</b>	<b>Residente</b>
<b>Versión</b>	1.0
<b>Descripción</b>	Este actor representa al usuario del sistema que tiene privilegios para registrar, modificar, actualizar y eliminar casos de anomalías en la UMMFN, y de patologías generales, generar informes y hacer consultas, pero no puede gestionar privilegios ni cuentas de usuario.

<b>ACT-03</b>	<b>Estudiante</b>
<b>Versión</b>	1.0
<b>Descripción</b>	Este actor representa al usuario del sistema que tiene accesos restringidos, únicamente puede consultar casos de patologías generales y casos de anomalías de la UMMFN pero sin visualizar los datos del paciente y los detalles del caso.

<b>ACT-04</b>	<b>Server Admin</b>
<b>Versión</b>	1.0
<b>Descripción</b>	Este actor representa al administrador del servidor en el cual estará alojada la aplicación FETHUS, únicamente se hará cargo de la realización y la carga de las copias de seguridad de la base de datos.

***TABLAS ASOCIADAS A LOS CASOS DE USO:***

<b>RF-01</b>	<b>Crear cuentas de usuario</b>	
<b>Versión</b>	1.0	
<b>Objetivos asociados</b>	OBJ-06 Administrar el acceso y los permisos de los usuarios	
<b>Requisitos asociados</b>	RI-03 Información sobre usuarios del sistema RF-02 Asignar permisos	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el root desee crear una cuenta de usuario.	
<b>Precondición</b>	El usuario no está registrado en el sistema de información. El root del sistema conoce la información personal del usuario.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El root ingresa al menú de administración de cuentas de usuario y selecciona la opción: Crear nueva cuenta de

		Usuario.
	2	El sistema solicita mediante un formulario toda la información referente al usuario. (Nombre, apellidos, login password, pregunta secreta, respuesta pregunta secreta, e-mail, tipo de usuario.)
	3	El root proporciona los datos solicitados.
	4	El sistema valida la información, si existe alguna incongruencia, se vuelve al paso 2.
	5	El sistema muestra un mensaje de confirmación y envía un correo electrónico de notificación al nuevo usuario.
<b>Pos condición</b>	El nuevo usuario esta registrado con determinados privilegios.	
<b>Excepciones</b>	Ninguna	

<b>RF-02</b>	<b>Asignar permisos</b>
<b>Versión</b>	1.0
<b>Objetivos asociados</b>	OBJ-06 Administrar el acceso y los permisos de los usuarios
<b>Requisitos asociados</b>	RF-01 Crear cuentas de usuario RF-03 Modificar cuentas de usuario RI-03 Información sobre usuarios del sistema

<b>Descripción</b>	El sistema debe brindar la opción al root de asignar privilegios o de “estudiante” o de “residente” a los usuarios del sistema. El sistema deberá comportarse tal como se describe en el siguiente caso de uso.	
<b>Precondición</b>	E root desea registrar un nuevo usuario en el sistema o desea modificar alguna cuenta de usuario previamente creada.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El root elige una de las opciones que ofrece el sistema para darle privilegios al usuario en cuestión.
	2	El root acepta los privilegios que asignó.
<b>Pos condición</b>	El usuario posee determinados privilegios con los cuales podrá hacer uso del sistema de información.	
<b>Excepciones</b>	Ninguna	

<b>RF-03</b>	<b>Modificar cuentas de usuario</b>
<b>Versión</b>	1.0
<b>Objetivos asociados</b>	OBJ-06 Administrar el acceso y los permisos de los usuarios
<b>Requisitos asociados</b>	RF-02 Asignar permisos RI-03 Información sobre usuarios del sistema

<b>Descripción</b>	El sistema debe proveer una interfaz en la que el root pueda modificar tanto los datos de los usuarios del sistema como sus privilegios. El sistema deberá comportarse tal como se describe en el siguiente caso de uso.	
<b>Precondición</b>	El usuario a eliminar esta previamente registrado en el sistema.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El root solicita al sistema comenzar el proceso de modificación de los datos de un usuario.
	2	El root selecciona el usuario al cual desea modificar sus datos.
	3	El sistema muestra los datos del usuario seleccionado de manera editable.
	4	El root modifica los datos personales o los privilegios del usuario que desee y le solicita al sistema que los almacene.
	5	El sistema muestra un mensaje de confirmación.
<b>Pos condición</b>	La nueva información de la cuenta del usuario es actualizada	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	4	Si el root solicita cancelar la operación, el sistema cancela la operación, a continuación este caso de uso termina.

<b>RF-04</b>	<b>Eliminar cuentas de usuario</b>	
<b>Versión</b>	1.0	
<b>Objetivos asociados</b>	OBJ-06 Administrar el acceso y los permisos de los usuarios	
<b>Requisitos asociados</b>	RI-03 Información sobre usuarios del sistema	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el root crea necesaria la eliminación de algún usuario del sistema.	
<b>Precondición</b>	El usuario a eliminar esta previamente registrado en el sistema.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El root solicita al sistema comenzar el proceso de eliminación de cuenta de usuarios.
	2	El root selecciona el usuario al cual desea eliminar su cuenta.
	3	El sistema pregunta si en verdad desea eliminar la cuenta de usuario.
	4	El root aprueba la eliminación definitiva de la cuenta de usuario.
	5	El sistema muestra un mensaje de confirmación.
<b>Pos condición</b>		

Excepciones	Paso	Acción
	3	Si el root solicita cancelar la operación, el sistema cancela la operación, a continuación este caso de uso termina.

<b>RF-05</b>	<b>Autenticarse</b>	
<b>Versión</b>	1.0	
<b>Objetivos asociados</b>	OBJ-06 Administrar el acceso y los permisos de los usuarios	
<b>Requisitos asociados</b>	RF-06 Modificar datos personales RI-03 Información sobre usuarios del sistema	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso, cuando cualquier usuario desee autenticar su identidad ante el sistema para de este modo poder acceder a las funciones privilegiadas de éste.	
<b>Precondición</b>	El usuario desea acceder al sistema.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	Una vez que este caso de uso ha sido ejecutado, el sistema muestra un formulario para que el usuario digite su login y su password.
	2	Si la contraseña es correcta, el sistema deja que el usuario acceda a los servicios a los cuales tiene acceso, según su rol.

<b>Pos condición</b>		
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	2	Si la contraseña es errónea, este caso de uso regresa al paso 1.

<b>RF-06</b>	<b>Modificar datos personales</b>	
<b>Versión</b>	1.0	
<b>Objetivos asociados</b>	OBJ-06 Administrar el acceso y los permisos de los usuarios	
<b>Requisitos asociados</b>	RF-05 Autenticarse RI-03 Información sobre usuarios del sistema	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso, cuando cualquier usuario desee modificar sus datos personales.	
<b>Precondición</b>	El usuario que desea modificar su información personal se ha autenticado ante el sistema.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario solicita al sistema comenzar el proceso de modificación de sus datos personales.
	2	El sistema muestra los datos actuales del usuario de manera

		editable.
	3	El usuario modifica los datos que desea y solicita al sistema que los almacene.
	4	El sistema modifica los datos correspondientes que el usuario cambió e informa que el proceso ha terminado con éxito.
<b>Pos condición</b>	La información del usuario está actualizada.	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	2	Si el usuario solicita cancelar la operación, el sistema cancela la operación, a continuación este caso de uso termina.

<b>RF-07</b>	<b>Registro de casos de la UMMFN</b>
<b>Versión</b>	1.0
<b>Objetivos asociados</b>	OBJ-01 Gestionar los casos de anomalías perinatales de la UMMFN
<b>Requisitos asociados</b>	RF-08 Registrar datos del paciente RF-09 Registrar antecedentes del paciente RF-10 Identificar patologías generales asociadas RI-01 Información sobre anomalías perinatales de la UMMFN
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso, cuando el root o el residente desee ingresar un nuevo

	caso de anomalía perinatal.	
<b>Precondición</b>	Se detecta un nuevo caso de malformación fetal en la UMMFN.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El root o el residente solicita al sistema comenzar el proceso de registro de un nuevo caso de malformación fetal.
	2	El sistema muestra el formulario para la inserción de los datos personales del paciente.
	3	El root o el residente ingresa los datos personales del paciente y solicita el proceso de inserción de los antecedentes personales del paciente.
	4	El sistema muestra el formulario para la inserción de los antecedentes personales del paciente.
	5	El root o el residente ingresa los datos personales del paciente y solicita el proceso de inserción de los antecedentes.
<b>Pos condición</b>	El nuevo caso de malformación fetal esta registrado en el sistema.	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	3	Si el usuario solicita cancelar la operación, el sistema cancela la operación, a continuación este caso de uso termina.

<b>RF-10</b>	<b>Identificar patologías generales asociadas</b>
--------------	---

<b>Versión</b>	1.0	
<b>Objetivos asociados</b>	OBJ-01 Gestionar los casos de anomalías perinatales de la UMMFN OBJ-03 Gestionar patologías generales	
<b>Requisitos asociados</b>	RF-07 Registro de casos de la UMMFN RF-12 Actualizar casos de la UMMFN RI-01 Información sobre anomalías perinatales de la UMMFN RI-02 Información sobre patologías generales	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso, cuando el root o el residente desee clasificar la anomalía encontrada en la UMMFN, con una patología.	
<b>Precondición</b>	Se registra un nuevo caso de malformación fetal y se identifica el nombre de la malformación.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El root o el residente despliega una lista de patologías existentes y escoge la que se relaciona con el caso encontrado en la UMMFN.
	2	El root o el residente valida la opción elegida.
<b>Pos condición</b>	El nuevo caso de la UMMFN está asociado a una patología.	
<b>Excepciones</b>	Ninguna	

<b>RF-11</b>	<b>Modificar casos de la UMMFN</b>	
<b>Versión</b>	1.0	
<b>Objetivos asociados</b>	OBJ-01 Gestionar los casos de anomalías perinatales de la UMMFN	
<b>Requisitos asociados</b>	RI-01 Información sobre anomalías perinatales de la UMMFN	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso, cuando el root o el residente desee modificar determinados datos de algún caso de la UMMFN.	
<b>Precondición</b>	El caso a modificar debe estar registrado en el sistema.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El root o el residente solicita al sistema comenzar el proceso de modificación de un caso de malformación fetal de la UMMFN.
	2	El sistema despliega la información correspondiente al caso a modificar de manera editable.
	3	El root o el residente hace las modificaciones necesarias y solicita al sistema actualizar los cambios.
	4	El sistema actualiza los cambios e informa al usuario un aviso sobre los cambios realizados.
<b>Pos condición</b>	El caso de malformación fetal es modificado y actualizado.	

<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	3	Si el usuario solicita cancelar la operación, el sistema cancela la operación, a continuación este caso de uso termina.

<b>RF-12</b>	<b>Actualizar casos de la UMMFN</b>	
<b>Versión</b>	1.0	
<b>Objetivos asociados</b>	OBJ-01 Gestionar los casos de anomalías perinatales de la UMMFN	
<b>Requisitos asociados</b>	RF-10 Identificar patologías generales asociadas RF-13 Registrar información de la sesión ecográfica RI-01 Información sobre anomalías perinatales de la UMMFN	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso, cuando el root o el residente solicite actualizar los datos referentes de algún caso de la UMMFN.	
<b>Precondición</b>	El caso de malformación ha sido registrado en el sistema con anterioridad.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El root o el residente solicita al sistema comenzar el proceso de actualización de un caso de malformación fetal de la UMMFN.

	2	El sistema despliega la información correspondiente al caso solicitado con los nuevos campos a actualizar.
	3	El root o el residente inserta los datos a actualizar y ordena al sistema realizar la actualización.
	4	El sistema realiza la actualización e informa al usuario que el proceso ha finalizado.
<b>Pos condición</b>	El caso de malformación está actualizado.	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	3	Si el usuario solicita cancelar la operación, el sistema cancela la operación, a continuación este caso de uso termina.

<b>RF-13</b>	<b>Registrar información de la sesión ecográfica</b>
<b>Versión</b>	1.0
<b>Objetivos asociados</b>	OBJ-01 Gestionar los casos de anomalías perinatales de la UMMFN
<b>Requisitos asociados</b>	RF-12 Actualizar casos de la UMMFN RI-01 Información sobre anomalías perinatales de la UMMFN
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso, cuando el root o el residente solicite al sistema registrar información referente a la sesión ecográfica del paciente.

<b>Precondición</b>	El caso está registrado en el sistema y se le está haciendo seguimiento por medio de sesiones ecográficas.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El root o el residente solicita al sistema comenzar el proceso de registro de información de la sesión ecográfica del paciente en cuestión.
	2	El sistema despliega la información correspondiente al caso solicitado con los nuevos campos de la sesión a registrar.
	3	El root o el residente ingresa los datos, imágenes, gráficos, etc. de la nueva sesión y ordena al sistema guardar los cambios.
	4	El sistema realiza el registro e informa al usuario que el proceso ha finalizado.
<b>Pos condición</b>	La sesión ecográfica está registrada en el sistema.	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	3	Si el usuario solicita cancelar la operación, el sistema cancela la operación, a continuación este caso de uso termina.

<b>RF-14</b>	<b>Dar de alta casos de la UMMFN</b>
<b>Versión</b>	1.0

<b>Objetivos asociados</b>	OBJ-01 Gestionar los casos de anomalías perinatales de la UMMFN	
<b>Requisitos asociados</b>	RF-15 Registrar conclusiones del caso de la UMMFN RI-01 Información sobre anomalías perinatales de la UMMFN	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso, cuando el root o el residente decida que determinado caso debe ser dado de alta ya sea porque el bebé nació, falleció o se detectó que el caso no es anómalo.	
<b>Precondición</b>	El caso ya no es objeto de estudio por parte de la junta perinatal.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El root o el residente solicita al sistema comenzar el proceso de dar de alta a un caso de la UMMFN.
	2	El sistema pregunta al usuario si está seguro de dar de alta al caso.
	3	El root o el residente ratifica su deseo dar de alta al caso en cuestión.
	4	El sistema realiza el proceso solicitado y muestra un mensaje al usuario donde informa que el proceso ha finalizado con éxito.
<b>Pos condición</b>	El sistema da de alta al caso y guarda la información relevante para el sistema.	
<b>Excepciones</b>	Paso	Acción

	2	Si el usuario solicita cancelar la operación, el sistema cancela la operación, a continuación este caso de uso termina.
--	---	---

<b>RF-15</b>	<b>Registrar conclusiones del caso de la UMMFN</b>	
<b>Versión</b>	1.0	
<b>Objetivos asociados</b>	OBJ-01 Gestionar los casos de anomalías perinatales de la UMMFN	
<b>Requisitos asociados</b>	RF-14 Dar de alta casos de la UMMFN RI-01 Información sobre anomalías perinatales de la UMMFN	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso:	
<b>Precondición</b>	El root o el residente dieron de alta al caso de la UMMFN.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El root o el residente solicita al sistema iniciar el proceso para registrar las conclusiones de casos de la UMMFN.
	2	El sistema despliega el formulario con los campos a llenar sobre las conclusiones del caso.
	3	El root o el residente llena el formulario y solicita al sistema guardar la información suministrada.
	4	El sistema guarda la información ingresada e informa al

	usuario que el proceso ha finalizado con éxito.
<b>Pos condición</b>	El sistema cuenta con las conclusiones del caso al que se dio de alta.
<b>Excepciones</b>	Ninguna

<b>RF-16</b>	<b>Registrar patologías generales</b>	
<b>Versión</b>	1.0	
<b>Objetivos asociados</b>	OBJ-03 Gestionar patologías generales	
<b>Requisitos asociados</b>	RI-02 Información sobre patologías generales	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso:	
<b>Precondición</b>	El sistema no tiene registrada la patología general.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El root o el residente solicita al sistema iniciar el proceso para registrar patologías generales.
	2	El sistema despliega el formulario con los campos a llenar sobre la patología.
	3	El root o el residente llena los campos del formulario y solicita al sistema almacenar los datos ingresados.

	4	El sistema almacena los datos e informa al usuario que la operación ha sido exitosa.
<b>Pos condición</b>	La patología general está almacenada en el sistema de información.	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	3	Si el usuario solicita cancelar la operación, el sistema cancela la operación, a continuación este caso de uso termina.

<b>RF-17</b>	<b>Modificar patologías generales</b>	
<b>Versión</b>	1.0	
<b>Objetivos asociados</b>	OBJ-03 Gestionar patologías generales	
<b>Requisitos asociados</b>	RI-02 Información sobre patologías generales	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso, cuando el root o el residente	
<b>Precondición</b>	La patología general a modificar esta previamente almacenada en el sistema.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El root o el residente solicita al sistema iniciar el proceso para modificar patologías generales.

	2	El sistema muestra los datos correspondientes a la patología general solicitada de manera editable.
	3	El root o el residente modifica los datos que el sistema le permite y solicita al sistema que los almacene.
	4	El sistema modifica los datos de la patología general e informa al usuario que el proceso se realizó exitosamente.
<b>Pos condición</b>	La patología general fue actualizada con éxito.	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	3	Si el usuario solicita cancelar la operación, el sistema cancela la operación, a continuación este caso de uso termina.

<b>RF-18</b>	<b>Consultar por casos de pacientes</b>
<b>Versión</b>	1.0
<b>Objetivos asociados</b>	OBJ-01 Gestionar los casos de anomalías perinatales de la UMMFN OBJ-02 Gestionar los pacientes
<b>Requisitos asociados</b>	RF-19 Listar casos de pacientes RF-20 Consultar caso por sistema involucrado RF-21 Consultar caso por nombre de patología RF-22 Consultar caso por palabras claves del paciente RI-01 Información sobre anomalías perinatales de la UMMFN

<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso, cuando el root, estudiante o el residente solicite al sistema retornar información sobre los casos registrados de la UMMFN	
<b>Precondición</b>	El usuario ya está registrado en el sistema de información.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario ingresa al menú del módulo de consultas selecciona la opción: Consultar casos de Pacientes.
	2	El sistema muestra las diferentes categorías por las cuales puede buscar el usuario de acuerdo a sus privilegios.
	3	El usuario selecciona la categoría mediante la cual desea realizar su consulta.
	4	Se ejecuta el caso de uso correspondiente a la categoría de búsqueda seleccionada.
<b>Pos condición</b>	El Usuario visualiza la información que responde a sus parámetros de búsqueda: RF-21	
<b>Excepciones</b>	Ninguna	

<b>RF-19</b>	<b>Listar casos de pacientes</b>
<b>Versión</b>	1.0

<b>Objetivos asociados</b>	OBJ-01 Gestionar los casos de anomalías perinatales de la UMMFN OBJ-02 Gestionar los pacientes	
<b>Requisitos asociados</b>	RF-18 Consultar por casos de pacientes RF-23 Ver detalles de los casos de pacientes RI-01 Información sobre anomalías perinatales de la UMMFN	
<b>Precondición</b>	El usuario ha ordenado al sistema realizar alguna consulta.	
<b>Descripción</b>	El sistema debe mostrarle los resultados de las consultas al usuario mediante registros ordenados ya sea, alfabéticamente o cronológicamente etc. como se describe en el siguiente caso de uso	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El sistema muestra un mensaje en caso tal que la consulta no arroje ningún resultado.
	2	El sistema imprime una tabla con los resultados y cada uno tiene un vínculo hacia los detalles del caso. RF-23
<b>Pos condición</b>	Ninguno	
<b>Excepciones</b>	Ninguna	

<b>RF-20</b>	<b>Consultar caso por sistema involucrado</b>
<b>Versión</b>	1.0

<b>Objetivos asociados</b>	OBJ-01 Gestionar los casos de anomalías perinatales de la UMMFN OBJ-02 Gestionar los pacientes	
<b>Requisitos asociados</b>	RF-20 Consultar por casos de pacientes RI-01 Información sobre anomalías perinatales de la UMMFN	
<b>Descripción</b>	El sistema debe proveer una interfaz en la que los usuarios puedan consultar casos respecto a los sistemas que se ven afectados. El sistema deberá comportarse tal como se describe en el siguiente caso de uso.	
<b>Precondición</b>	El usuario ha ordenado al sistema realizar alguna consulta.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario selecciona la opción consultar por Sistemas involucrados.
	2	El usuario selecciona el o los sistemas afectados, respecto a los cuales desea elaborar la búsqueda.
	3	El sistema realiza la consulta y retorna los casos que cumplan las condiciones preseleccionadas.
<b>Pos condición</b>	Visualización normal. RF-21	

<b>RF-21</b>	<b>Consultar caso por nombre de patología</b>
<b>Versión</b>	1.0

<b>Objetivos asociados</b>	OBJ-01 Gestionar los casos de anomalías perinatales de la UMMFN OBJ-02 Gestionar los pacientes	
<b>Requisitos asociados</b>	RF-18 Consultar por casos de pacientes RI-01 Información sobre anomalías perinatales de la UMMFN	
<b>Descripción</b>	El sistema debe proveer una interfaz en la que los usuarios puedan consultar casos respecto a las patologías presentadas.. El sistema deberá comportarse tal como se describe en el siguiente caso de uso.	
<b>Precondición</b>	El usuario ha ordenado al sistema realizar alguna consulta.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario selecciona la opción consultar por Patologías del caso.
	2	El usuario selecciona el o las patologías, respecto a los cuales desea elaborar la búsqueda.
	3	El sistema realiza la consulta y retorna los casos que cumplan las condiciones preseleccionadas.
<b>Pos condición</b>	Visualización normal. RF-21	

<b>RF-22</b>	<b>Consultar caso por palabras claves del paciente</b>
<b>Versión</b>	1.0

<b>Objetivos asociados</b>	OBJ-01 Gestionar los casos de anomalías perinatales de la UMMFN OBJ-02 Gestionar los pacientes	
<b>Requisitos asociados</b>	RF-18 Consultar por casos de pacientes RI-01 Información sobre anomalías perinatales de la UMMFN	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el root o el residente desee consultar casos de la UMMFN utilizando como parámetros de búsqueda, información personal del paciente.	
<b>Precondición</b>	Solo el root y el internista pueden realizar este tipo de consulta.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario selecciona la opción consultar por Información del paciente. Ejemplo: antecedentes.
	2	El usuario selecciona en forma dinámica los parámetros que desea tener en cuenta en la búsqueda, El sistema valida todos los datos.
	3	El sistema realiza la consulta y retorna los casos que cumplan las condiciones preseleccionadas.
<b>Pos condición</b>	El sistema muestra el listado de casos. RF-21	
<b>Excepciones</b>	Ninguno	

<b>RF-23</b>	<b>Ver detalles de casos de pacientes</b>	
<b>Versión</b>	1.0	
<b>Objetivos asociados</b>	OBJ-01 Gestionar los casos de anomalías perinatales de la UMMFN OBJ-02 Gestionar los pacientes	
<b>Requisitos asociados</b>	RF-19 Listar casos de pacientes. RI-01 Información sobre anomalías perinatales de la UMMFN	
<b>Descripción</b>	El sistema deberá presentar de manera concisa, completa, y ordenada la información referente a cada caso retornado en una consulta. Mediante una ventana emergente.	
<b>Precondición</b>	Se ha retornado por lo menos un registro en una consulta.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	Una vez que este caso de uso ha sido ejecutado, se desplegará una ventana emergente con toda la información referente al caso.
<b>Pos condición</b>	Ninguna.	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	2	El sistema mostrará la información disponible a la fecha. Independientemente de si se ha dado de alta o no.

<b>RF-24</b>	<b>Consultar patologías generales</b>	
<b>Versión</b>	1.0	
<b>Objetivos asociados</b>	OBJ-03 Gestionar patologías generales	
<b>Requisitos asociados</b>	RF-25 Consultar patología por nombre RF-26 Consultar patología por sistema RF-27 Listar patologías generales RI-02 Información sobre patologías generales	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso, cuando cualquier usuario desee consultar información general acerca de alguna patología perinatal.	
<b>Precondición</b>	Ninguna.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario ingresa al menú del módulo de consultas selecciona la opción: Consultar patologías generales.
	2	El sistema muestra las diferentes categorías por las cuales puede buscar el usuario. Esta opción es independiente de los privilegios.
	3	El usuario selecciona la categoría mediante la cual desea realizar su consulta.

	4	Se ejecuta el caso de uso correspondiente a la categoría de búsqueda seleccionada.
<b>Pos condición</b>	Ninguna	

<b>RF-25</b>	<b>Consultar patología por nombre</b>	
<b>Versión</b>	1.0	
<b>Objetivos asociados</b>	OBJ-03 Gestionar patologías generales	
<b>Requisitos asociados</b>	RF-24 Consultar patologías generales RI-01 Información sobre anomalías perinatales de la UMMFN	
<b>Descripción</b>	El sistema debe proveer una interfaz en la que los usuarios puedan consultar información referente a a las patologías registradas.. El sistema deberá comportarse tal como se describe en el siguiente caso de uso.	
<b>Precondición</b>	El usuario ha ordenado al sistema realizar alguna consulta.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario selecciona la opción consultar Patologías por nombre
	2	El usuario escribe la(s) palabra(s) relacionadas con el nombre de la patología.

	3	El sistema realiza la consulta y retorna los casos que cumplan las condiciones preseleccionadas.
<b>Pos condición</b>	Visualización normal. RF-21	

<b>RF-26</b>	<b>Consultar patología por Sistema</b>	
<b>Versión</b>	1.0	
<b>Objetivos asociados</b>	OBJ-03 Gestionar patologías generales	
<b>Requisitos asociados</b>	RF-24 Consultar patologías generales RI-01 Información sobre anomalías perinatales de la UMMFN	
<b>Descripción</b>	El sistema debe proveer una interfaz en la que los usuarios puedan consultar información referente a a las patologías registradas.. El sistema deberá comportarse tal como se describe en el siguiente caso de uso.	
<b>Precondición</b>	El usuario ha ordenado al sistema realizar alguna consulta.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario selecciona la opción consultar Patologías por Sistema involucrado
	2	El usuario escribe la(s) palabra(s) relacionadas con el sistema involucrado

	3	El sistema realiza la consulta y retorna los casos que cumplan las condiciones preseleccionadas.
<b>Pos condición</b>	Visualización normal. RF-21	

<b>RF-27</b>	<b>Listar Patologías generales</b>	
<b>Versión</b>	1.0	
<b>Objetivos asociados</b>	OBJ-03 Gestionar patologías generales	
<b>Requisitos asociados</b>	RF-24 Consultar patologías generales RF-28 Ver detalles de patologías Generales RI-01 Información sobre anomalías perinatales de la UMMFN	
<b>Precondición</b>	El usuario ha ordenado al sistema realizar alguna consulta.	
<b>Descripción</b>	El sistema debe mostrarle los resultados de las consultas al usuario mediante registros ordenados ya sea, alfabéticamente o por sistema involucrado etc. como se describe en el siguiente caso de uso	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El sistema muestra un mensaje en caso tal que la consulta no arroje ningún resultado.
	2	El sistema imprime una tabla con los resultados y cada uno tiene un vínculo hacia los detalles de la patología. RF-30

<b>Pos condición</b>	Visualización normal. RF-21
<b>Excepciones</b>	Ninguna

<b>RF-28</b>	<b>Ver detalles de patologías generales</b>	
<b>Versión</b>	1.0	
<b>Objetivos asociados</b>	OBJ-03 Gestionar patologías generales	
<b>Requisitos asociados</b>	RF-27 Listar patologías Generales RF-29 Ver casos relacionados RI-01 Información sobre anomalías perinatales de la UMMFN	
<b>Descripción</b>	El sistema deberá presentar de manera concisa, completa, y ordenada la información referente a cada patología retornada en una consulta.	
<b>Precondición</b>	Se ha retornado por lo menos un registro en una consulta.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	Una vez que este caso de uso ha sido ejecutado, se abre una ventana emergente con toda la información referente al caso.
<b>Pos condición</b>	Ninguna.	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>

	2	El sistema mostrará la información disponible a la fecha. Es responsabilidad de los usuarios poblar la base de datos.
--	---	---

<b>RF-29</b>	<b>Ver Casos Relacionados</b>
<b>Versión</b>	1.0
<b>Objetivos asociados</b>	OBJ-03 Gestionar patologías generales
<b>Requisitos asociados</b>	RF-38 Ver detalles de patologías Generales RI-01 Información sobre anomalías perinatales de la UMMFN
<b>Descripción</b>	El sistema deberá permitirle al usuario listar los casos de la UMMFN desde el modulo de patologías generales, si y solo si tiene los privilegios para hacerlo.

<b>RF-30</b>	<b>Consultar Usuario</b>
<b>Versión</b>	1.0
<b>Objetivos asociados</b>	OBJ-06 Administrar el acceso y los privilegios de los usuarios
<b>Requisitos asociados</b>	RF-31 Consultar por tipo de usuario RF-32 Consultar usuarios por palabras clave RF-33 Listar usuarios

	RI-03 Información sobre usuarios del sistema	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso, cuando el usuario root desee consultar información acerca de algún usuario específico.	
<b>Precondición</b>	Ninguna.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario ingresa al menú del módulo de consultas selecciona la opción: Consultar Usuarios.
	2	El sistema muestra las diferentes categorías por las cuales puede buscar el usuario.
	3	El usuario selecciona la categoría mediante la cual desea realizar su consulta.
	4	Se ejecuta el caso de uso correspondiente a la categoría de búsqueda seleccionada.
<b>Pos condición</b>	Ninguna	

<b>RF-31</b>	<b>Consultar por tipo de usuario</b>
<b>Versión</b>	1.0
<b>Objetivos asociados</b>	OBJ-06 Administrar el acceso y los privilegios de los usuarios

<b>Requisitos asociados</b>	RF-30 Consultar usuario RI-03 Información sobre usuarios del sistema	
<b>Descripción</b>	El sistema debe proveer una interfaz en la que los usuarios puedan consultar información referente a las actividades realizadas por los diferentes tipos de usuarios. El sistema deberá comportarse tal como se describe en el siguiente caso de uso.	
<b>Precondición</b>	El usuario ha ordenado al sistema realizar alguna consulta.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario selecciona la opción consultar usuarios por tipo.
	2	El sistema realiza la consulta y retorna los usuarios que cumplan las condiciones preseleccionadas.
<b>Pos condición</b>	Ninguna.	

<b>RF-32</b>	<b>Consultar Usuarios por palabras clave</b>
<b>Versión</b>	1.0
<b>Objetivos asociados</b>	OBJ-06 Administrar el acceso y los privilegios de los usuarios
<b>Requisitos asociados</b>	RF-30 Consultar usuario RI-03 Información sobre usuarios del sistema

<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso, cuando el usuario root desee consultar información acerca de algún usuario específico.	
<b>Precondición</b>	Ninguna.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario ingresa al menú del módulo de consultas selecciona la opción: Consultar Usuarios.
	2	El sistema muestra las diferentes categorías por las cuales puede buscar el usuario.
	3	El usuario selecciona los parámetros de coincidencia mediante los cuales desea realizar su consulta.
4	Se ejecuta el caso de uso correspondiente a la categoría de búsqueda seleccionada.	
<b>Pos condición</b>	Ninguna	

<b>RF-33</b>	<b>Listar Usuarios</b>
<b>Versión</b>	1.0
<b>Objetivos asociados</b>	OBJ-06 Administrar el acceso y los privilegios de los usuarios
<b>Requisitos</b>	RF-30 Consultar usuario

<b>asociados</b>	RF-34 Ver detalles de usuario RI-03 Información sobre usuarios del sistema	
<b>Precondición</b>	El usuario ha ordenado al sistema realizar alguna consulta.	
<b>Descripción</b>	El sistema debe mostrarle los resultados de las consultas al usuario mediante registros ordenados ya sea, alfabéticamente o por tipo de usuario como se describe en el siguiente caso de uso	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El sistema muestra un mensaje en caso tal que la consulta no arroje ningún resultado.
	2	El sistema imprime una tabla con los resultados y cada uno tiene un vínculo hacia los detalles de la patología. RF-30
<b>Pos condición</b>	Visualización normal. RF-21	
<b>Excepciones</b>	Ninguna	

<b>RF-34</b>	<b>Ver detalles de Usuario</b>
<b>Versión</b>	1.0
<b>Objetivos asociados</b>	OBJ-06 Administrar el acceso y los privilegios de los usuarios
<b>Requisitos asociados</b>	RF-33 Listar usuarios

	RI-01 Información sobre anomalías perinatales de la UMMFN	
<b>Descripción</b>	El sistema deberá presentar de manera concisa, completa, y ordenada la información referente al usuario seleccionado, incluyendo últimas transacciones realizadas, última vez en línea etc.	
<b>Precondición</b>	Se ha retornado por lo menos un registro en una consulta.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	Una vez que este caso de uso ha sido ejecutado, se abre una ventana emergente con toda la información referente al usuario.
<b>Pos condición</b>	Ninguna.	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	2	El sistema mostrará la información disponible a la fecha. Es responsabilidad de los usuarios poblar la base de datos.

<b>RF-35</b>	<b>Generar Informe de Usuarios</b>
<b>Versión</b>	1.0
<b>Objetivos asociados</b>	OBJ-06 Administrar el acceso y los privilegios de los usuarios
<b>Requisitos asociados</b>	RI-03 Información sobre usuarios del sistema.

<b>Descripción</b>	El sistema deberá presentar de manera concisa, completa, y ordenada el reporte de los 10 usuarios que más han hecho uso del sistema de información en determinado rango de tiempo.	
<b>Precondición</b>	Existen usuarios que ya se han validado ante el sistema.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	Una vez que este caso de uso ha sido ejecutado, se muestra en pantalla el reporte con dicha información.
<b>Pos condición</b>	Ninguna.	
<b>Excepciones</b>	Ninguna.	

<b>RF-36</b>	<b>Generar Informe Epidemiológico</b>
<b>Versión</b>	1.0
<b>Objetivos asociados</b>	OBJ-01 Gestionar los casos de anomalías perinatales de la UMMFN OBJ-03 Gestionar patologías generales OBJ-05 Gestionar informes estadísticos
<b>Requisitos asociados</b>	RI-01 Información sobre anomalías perinatales de la UMMFN RI-02 Información sobre patologías generales
<b>Descripción</b>	El sistema deberá presentar de manera concisa, completa, y ordenada el reporte en porcentajes de los casos asociados a las

	patologías generales en determinado rango de tiempo.	
<b>Precondición</b>	Existen registros de las diferentes patologías en los casos.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	Una vez que este caso de uso ha sido ejecutado, se muestra en pantalla el reporte con dicha información.
<b>Pos condición</b>	Ninguna.	
<b>Excepciones</b>	Ninguna.	

<b>RF-37</b>	<b>Generar Informe de Mortalidad</b>	
<b>Versión</b>	1.0	
<b>Objetivos asociados</b>	OBJ-01 Gestionar los casos de anomalías perinatales de la UMMFN	
<b>Requisitos asociados</b>	RI-01 Información sobre anomalías perinatales de la UMMFN	
<b>Descripción</b>	El sistema deberá presentar de manera concisa, completa, y ordenada el reporte del número de casos en los cuales los fetos nacieron muertos y vivos.	
<b>Precondición</b>	Se ha dado de alta al menos a un caso de la UMMFN.	
<b>Secuencia</b>	<b>Paso</b>	<b>Acción</b>

<b>normal</b>	1	Una vez que este caso de uso ha sido ejecutado, se muestra en pantalla el reporte con dicha información.
<b>Pos condición</b>	Ninguna.	
<b>Excepciones</b>	Ninguna.	

<b>RF-38</b>	<b>Realizar copia de seguridad</b>	
<b>Versión</b>	1.0	
<b>Objetivos asociados</b>	OBJ-04 Gestionar imágenes	
<b>Requisitos asociados</b>	Ninguno	
<b>Descripción</b>	El administrador del servidor deberá realizar la copia de seguridad manualmente de la base de datos generando un archivo de tipo sql que contenga la información de todos los registros almacenados en la base de datos del sistema.	
<b>Precondición</b>	La base de datos se encuentra mínimamente poblada.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	Una vez que este caso de uso ha sido ejecutado, mysql confirma que la copia de seguridad a la base de datos ha sido exitosa.

<b>Pos condición</b>	Ninguna.
<b>Excepciones</b>	Ninguna.

<b>RF-39</b>	<b>Exportar contenidos multimedia</b>	
<b>Versión</b>	1.0	
<b>Objetivos asociados</b>	OBJ-04 Gestionar imágenes	
<b>Requisitos asociados</b>	RF-38 Realizar copias de seguridad RI-04 Banco de imágenes	
<b>Descripción</b>	El administrador del servidor deberá realizar la copia de seguridad manualmente de la carpeta en la cual se encuentran las imágenes ecográficas del sistema.	
<b>Precondición</b>	Hay imágenes previamente almacenadas en el sistema de información.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	Una vez que este caso de uso ha sido ejecutado, el sistema está totalmente replicado hasta la fecha de la realización de la copia de seguridad.
<b>Pos condición</b>	Ninguna.	
<b>Excepciones</b>	Ninguna.	

<b>RF-40</b>	<b>Cargar copia de seguridad</b>	
<b>Versión</b>	1.0	
<b>Objetivos asociados</b>	OBJ-04 Gestionar imágenes	
<b>Requisitos asociados</b>	RF-38 Realizar copias de seguridad	
<b>Descripción</b>	El administrador del servidor deberá cargar la copia de seguridad de la base de datos manualmente.	
<b>Precondición</b>	Se ha realizado previamente un backup del sistema.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	Una vez que este caso de uso ha sido ejecutado, mysql confirma que la copia de seguridad a la base de datos ha sido cargada exitosamente.
<b>Pos condición</b>	Ninguna.	
<b>Excepciones</b>	Ninguna.	

<b>RF-41</b>	<b>Cargar contenido multimedia</b>	
<b>Versión</b>	1.0	

<b>Objetivos asociados</b>	OBJ-04 Gestionar imágenes	
<b>Requisitos asociados</b>	RF-39 Exportar contenidos multimedia RI-04 Banco de imágenes	
<b>Descripción</b>	El administrador del servidor deberá retornar manualmente la carpeta en la cual se encuentran las imágenes ecográficas del sistema, al directorio de imágenes que se encuentra en la aplicación.	
<b>Precondición</b>	El administrador del servidor ha realizado previamente un backup de las imágenes del sistema.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	Una vez que este caso de uso ha sido ejecutado, las imágenes están a disposición nuevamente de los usuarios del sistema.
<b>Pos condición</b>	Ninguna.	
<b>Excepciones</b>	Ninguna.	

### ***REQUISITOS NO FUNCIONALES***

A continuación se listan todos los requisitos no funcionales que determinarán la construcción y el diseño del sistema.

### ***REQUISITOS DE LA INTERFAZ DE USUARIO***

**Fuentes de entrada de datos:** FETHUS debe permitir acceder a todos los menús del sistema utilizando tanto el mouse como el teclado.

**Destino de salida de datos:** Los destinos primarios de salida de datos serán la pantalla y la impresora.

**Elementos de la interfaz gráfica:** Las interacciones se basarán en acciones físicas sobre elementos visuales (íconos, botones, imágenes, mensajes de texto, barras de desplazamiento y navegación) y en selecciones de tipo menú con sintaxis y órdenes.

**Consistencia en la presentación:** El usuario debe ver la información y los objetos siempre con el mismo aspecto, siguiendo una misma lógica, y ubicados en sitios iguales.

**Consistencia en el comportamiento:** se refiere a que los objetos deben actuar siempre de la misma manera y así no sorprender al usuario.

**Comprensión del texto y los mensajes:** Se deberán presentar textos descriptivos y comprensibles al usuario utilizando un lenguaje fácil y lo menos técnico posible. Esto debe aplicarse tanto a los mensajes del sistema, las etiquetas de las opciones y cualquier texto que se presente al usuario.

**Divulgación progresiva:** Se mostrará al usuario sólo aquellas funciones a las que tenga derecho y a medida que las vaya necesitando, no es preciso ni aconsejable mostrar al usuario todas las funciones que ofrece el programa.

**Efectos inmediatos:** Las operaciones serán rápidas, incrementales, con efectos inmediatos y proporcionarán realimentación visual de las acciones que ejecuta el usuario.

**Operaciones reversibles:** Toda operación realizada por el usuario podrá ser reversible.

**Validación de datos:** La herramienta controlará las entradas para evitar el

ingreso de valores erróneos o fuera del rango especificado para cada tipo de dato.

**Mensajes de error informativos:** Los mensajes de error deben ser informativos y constructivos para que el usuario comprenda su equivocación, pueda remediarla en ese momento y no repetirla en adelante.

**Ayuda:** todos los programas deberán contar con un modulo de ayuda en línea donde se encuentren documentadas e ilustradas todas las funciones que contiene.

**Ayuda contextual:** se proporcionará ayuda referente a la acción que se ejecuta y sin tener que abandonar la pantalla de trabajo.

**Fotografías:** Las fotografías se almacenarán en formato JPG, en resolución máxima de 640\*480.

**Paginación de resultados:** Los resultados de las búsquedas que retornen un listado se mostrarán en varias páginas en caso de ser necesario.

**Alarmas:** El sistema debe notificar a cada usuario las alarmas relevantes a su rol, en forma de textos de un color apropiado.

**Imagen institucional:** La interfaz gráfica utilizará los colores institucionales de la universidad.

**Errores del sistema:** El usuario deberá ser informado acerca de los errores, ocasionados por hardware, software o redes, que detengan la labor que se está ejecutando.

**Perfil de usuario:** Se debe visualizar en todo momento el perfil del usuario (nombre de usuario y su categoría) que haya iniciado sesión en el sistema.

### ***REQUISITOS DE LA INTERFAZ DE HARDWARE***

Debido a que el sistema es vía Web, no se requiere capacidad robusta de la maquina, o hardware externo a la computadora para el uso normal del sistema. Solo se requiere las capacidades mínimas para una conexión a Internet ya sea banda ancha o de acceso conmutado, y la ejecución de un browser o navegador de Internet.

### ***REQUISITOS DE LA INTERFAZ DE SOFTWARE***

Los requisitos de software que se plantean son:

- Un explorador de Internet, por ejemplo Internet Explorer o Mozilla FireFox.
- Activación de JavaScript en el explorador.
- Sistema operativo de cualquier clase (GNU/Linux, Mac OS, Windows, entre otros).

### ***REQUISITOS DE RENDIMIENTO***

El sistema debe soportar la carga pico de actualización de 100 o 200 usuarios, gracias a la escalabilidad del sistema, dado el caso de que algún día se desee implantar el sistema de información en las Unidades MaternoFetales de otros centros médicos, y decidan centralizar su información para la realización de estudios a gran escala, y mejorar la toma de decisiones como un problema de salud pública.

### ***REQUISITOS DE DISEÑO Y DESARROLLO***

**Lenguaje unificado:** Las especificaciones y demás diagramas de diseño deberán hacerse usando UML 2.0

**Uso de herramientas de código libre:** El sistema será construido usando software libre.

**Documentación:** Se debe construir documentación clara, a todo nivel, del sistema para facilitar el desarrollo de nuevas funciones, y las labores de mantenimiento.

## **ANEXO B. ACTA DE ENTREGA DEL SISTEMA.**

### **Acta de entrega del sistema de información para apoyar el control prenatal y epidemiológico en la Unidad de Medicina Maternofetal del Hospital Universitario de Santander, FETHUS 1.0**

Reunidos en las instalaciones del Hospital Universitario de Santander el día lunes 1 de febrero de 2010, los integrantes del Comité de Sistemas de Información para el desarrollo de FETHUS certifican que el sistema:

Se construyó siguiendo estrictamente los requisitos que el comité estableció durante la fase de análisis en las reuniones realizadas con los estudiantes de ingeniería de sistemas autores del presente software.

Está en proceso de alimentación de su base de datos con el material fotográfico y textual (información ecográfica); proceso del cual es enteramente responsable el personal de la UMMFN integrantes del comité.

Ha superado las pruebas efectuadas por los usuarios finales, miembros del comité, y se encuentra completamente operativo para su puesta en funcionamiento.

Con base en lo expuesto anteriormente, el presente comité:

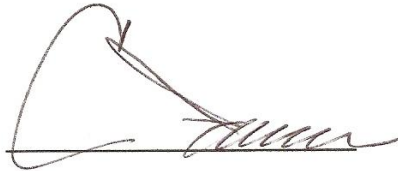
Otorga la calidad de “Terminado” al producto software entregado.

Acepta su plena conformidad con las características del producto entregado.

Acepta que los autores no son en absoluto responsables por la implantación y mantenimiento del producto entregado.

Acepta que la venta, redistribución y modificación del producto entregado debe contar con el consentimiento de los autores, obedeciendo al marco legal brindado por la UIS.

Firmado el día primero del mes de Febrero del año 2010 por:

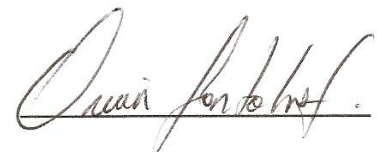


**Dr. Carlos Hernán Becerra.**

Jefe de la UMMFN.



**Dra. Mónica Andrea Beltrán.**  
Médico Especialista UMMFN.

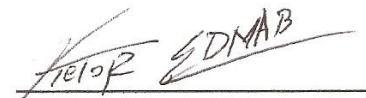


**Dr. Orian Fontalvo F.**

Md Residente de Ginecología II  
año UMMFN - HUS

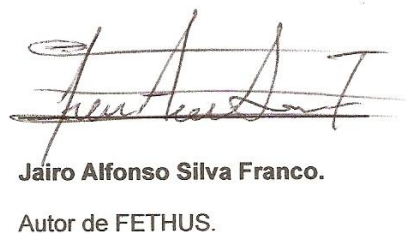


**Dr. Juan Carlos Otero.**  
Médico Especialista UMMFN.



**M.I.I.C.C. Víctor Martínez A.**

Gerente de proyecto.



**Jairo Alfonso Silva Franco.**  
Autor de FETHUS.



**Raúl Fernando Rivero Olarte.**  
Autor de FETHUS.