



UNIVERSIDAD  
NACIONAL  
DE LOJA



*Área de la Energía, las Industrias y los Recursos Naturales No Renovables*

---

CARRERA DE INGENIERÍA EN SISTEMAS

# **“Análisis, Diseño e Implantación del módulo de Fisiatría al sistema web “SISFA” para el Hospital Básico 7 B.I Loja”**

*Tesis previa a la obtención del Título  
de Ingeniero en Sistemas*

**Autor:**

- Alvarado Motoche Leonardo Xavier

**Director:**

- Ing. Marco Augusto Ocampo Carpio. Mg. Sc

**Fecha:**

- 26 de junio del 2015

**LOJA-ECUADOR**

**2015**

## **CERTIFICACIÓN DEL DIRECTOR**

Ing. Marco Augusto Ocampo Carpio. Mg. Sc.

**DOCENTE DE LA CARRERA DE INGENIERÍA EN SISTEMAS DE LA  
UNIVERSIDAD NACIONAL DE LOJA.**

### **CERTIFICA:**

Que la presente tesis con el tema “**Análisis, Diseño e Implantación del módulo de Fisiatría al sistema web “SISFA” para el Hospital Básico 7 B.I Loja**”, previa a la obtención del título de Ingeniero en Sistemas, fue prolijamente dirigido supervisado y revisado en cada uno de sus aspectos; razón por la cual doy validez y justificación, autorizado su presentación, sustentación y publicación.

A handwritten signature in blue ink, reading "Marco Augusto Ocampo Carpio", is written over a horizontal dotted line. The signature is enclosed within a large, loopy blue oval scribble.

Ing. Marco Augusto Ocampo Carpio. Mg. Sc  
**DIRECTOR**

## **AUTORÍA**

Yo **LEONARDO XAVIER ALVARADO MOTOCHÉ** declaro ser el autor del presente trabajo de tesis y eximo expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos de posibles reclamos o acciones legales por el contenido de la misma.

Adicionalmente acepto y autorizo a la Universidad Nacional de Loja, la publicación de mi tesis en el Repositorio – Biblioteca Virtual.

**Firma:**

A handwritten signature in blue ink, appearing to be 'L. Alvarado Motoche', written over a horizontal line.

**Cédula:** 1104594021

**Fecha:** 26 de junio del 2015

# **CARTA DE AUTORIZACIÓN DE TESIS POR PARTE DEL AUTOR, PARA LA CONSULTA, REPRODUCCIÓN PARCIAL O TOTAL Y PUBLICACIÓN DEL TEXTO COMPLETO.**

Yo **LEONARDO XAVIER ALVARADO MOTOCHÉ**, declaro ser autor de la tesis titulada “**ANÁLISIS, DISEÑO E IMPLANTACIÓN DEL MÓDULO DE FISIATRÍA AL SISTEMA WEB “SISFA” PARA EL HOSPITAL BÁSICO 7 B.I LOJA**”, como requisito para optar al grado de: **INGENIERO EN SISTEMAS**; autorizo al Sistema Bibliotecario de la Universidad Nacional de Loja para que con fines académicos; muestre al mundo la producción intelectual de la Universidad, a través de la visibilidad de su contenido de la siguiente manera en el Repositorio Digital Institucional:

Los usuarios pueden consultar el contenido de este trabajo en el RDI, en las redes de información del país y el exterior, con las cuales tenga convenio la Universidad.

La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia de la tesis que realice un tercero.

Para constancia de esta autorización, en la ciudad de Loja, a los veintiséis días del mes de junio del dos mil quince.

**Firma:** 

**Autor:** Leonardo Xavier Alvarado Motoche

**Cedula:** 1104594021

**Dirección:** Loja (Manuel Zambrano y Cuba)

**Correo electrónico:** leo.churon@gmail.com

**Teléfono:** 2587927 **Celular:** 0988139346

## **DATOS COMPLEMENTARIOS**

**Director de tesis:** Ing. Marco Augusto Ocampo Carpio, Mg. Sc.

**Tribunal de Grado:** Ing. Roberto Jácome Galarza, Mg. Sc.

Ing. Iván Siguencia Larreategui, Mg. Sc

Ing. Auliría Virginia Torres Ontaneda, Mg. Sc

## **AGRADECIMIENTO**

Quiero agradecer primeramente a Dios quien es el que me ha permitido llegar hasta este punto de mi vida, agradezco el apoyo incondicional de mis padres: Emigdio Oswaldo Alvarado Cuenca y Orfa María Motoche, quienes siempre han sembrado sus sueños y deseos en mí. A los docentes del Área de la Energía, las Industrias y los Recursos Naturales No Renovables por asesorarme y guiarme en el proceso de desarrollo como estudiante; al Ing. Marco Ocampo por haber contribuido con la dirección, revisión de este PFC. Al personal del Hospital Básico 7 BI Loja quienes me brindaron su apoyo y la información necesaria y así poder culminar exitosamente este PFC.



**Leonardo Xavier Alvarado Motoche**

## **DEDICATORIA**

Dedico esta PFC a mis hermanos y a mis padres: Emigdio Oswaldo Alvarado y Orfa María Motoche, quienes me han sabido guiar durante todas las etapas de mi vida y apoyar durante todo mi proceso de personal y profesional.

A todas las personas que directamente o indirectamente han contribuido a la realización del mismo.

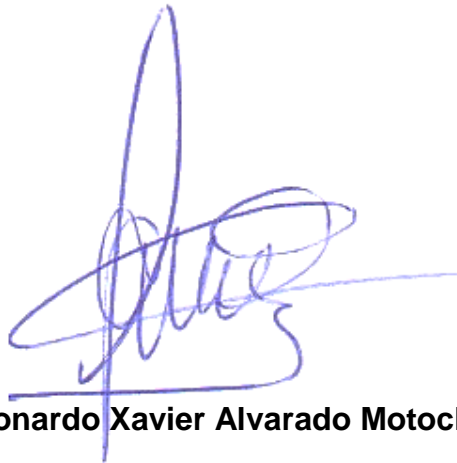
Y sobre todo quiero dedicar este trabajo a mi Dios, quien me ha regalado todos estos años de vida y puso a mi lado a todas estas personas extraordinarias que han sido mi inspiración y apoyo para alcanzar todas mis metas.



**Leonardo Xavier Alvarado Motoche**

## **CESIÓN DE DERECHOS**

El presente PFC queda a disposición de la Universidad Nacional de Loja y del Hospital Básico 7 BI Loja para que pueda determinar su utilización en los fines que considere pertinente.

A handwritten signature in blue ink, appearing to be 'Leonardo', with a horizontal line drawn underneath it.

**Leonardo Xavier Alvarado Motoche**

**C.I: 1104594021**

## **a. TÍTULO**

“ANÁLISIS, DISEÑO E IMPLANTACIÓN DEL MÓDULO DE FISIATRÍA AL SISTEMA WEB “SISFA” PARA EL HOSPITAL BÁSICO 7 B.I LOJA”.



## **b. RESUMEN**

El presente Proyecto de fin de Carrera (PFC), denominado “Análisis, Diseño e Implantación del módulo de Fisiatría al sistema web “SISFA” para el Hospital Básico 7 B.I Loja”, tiene como finalidad que este servicio de Fisiatría en esta unidad de salud, cuente con un sistema integrado el cual brinda información y control oportuno a los servicios de fisiatría.

La metodología de desarrollo aplicada, es XP la cual se inicia con varias reuniones con el personal del Servicio de Fisiatría, en las cuales se da claridad a las necesidades puntuales del sistema, a través de las historias de usuarios, las que sirven de base para crear una metáfora del sistema el cual se tendrá una idea general de la aplicación a implementar.

Para el desarrollo del sistema se utilizó el lenguaje de programación Ruby, con la ayuda del framework Ruby on Rails, el que es de código libre y brinda libertad al desarrollar sitios web, permitiendo hacer uso de HTML, CSS, SASS, JQuery, Coffeescript, los mismo que fueron útiles para la elaboración del Frontend del sistema.

El uso gemas como “sorcery” fue necesaria para realizar la autenticación de usuarios y encriptación de claves, además de “wicked\_pdf” necesaria para la elaboración de reportes e impresión de documentos, Por otra parte se realizaron pruebas como: funcionalidad, seguridad, aceptación, pruebas de carga y estrés para medir el rendimiento del servidor web.

El módulo de fisiatría se lo desarrollo tomando en cuenta el software antes usado, permitiendo llegar a determinar los procesos de registro de pacientes, el cual está relacionado con el módulo de doctores que permite registrar diagnósticos y enviar reportes médicos, en cuanto al departamento de estadística se recopilo toda la información correspondiente a los datos médicos generados y la necesidad de la realización Terapias.

El sistema de Fisiatría hoy en día está integrado con las dependencias de estadística, y doctores, con lo cual se dio solución a varios problemas planteados en este PFC.

## **SUMMARY**

The present PFC, called "Analysis, design and implementation of the module of physiatry to the web system "SISFA" for the basic Hospital 7 B.I Loja", aims that this service of physiatry at this health unit, has an integrated system which provides information and appropriate control to physical medicine services.

The development methodology is XP which begins with meetings with the staff of physiatry, which gives clarity to the specific needs of the system, through the stories of users, which are the basis to create a metaphor for the system, which will have a general idea of the application to be implemented.

The development of the system was the programming language Ruby, with the aid of the framework Ruby on Rails, which is open source and provides freedom to develop web sites, allowing you to make use of HTML, CSS, SASS, JQuery, Coffee script, the same that were useful for the development of the Frontend of the system.

The use of gems as "sorcery" was required to perform user authentication and encryption key, in addition to "wicked pdf" needed for the preparation of reports and document printing, on the other hand were tested as: functionality, security, acceptance, and load and stress tests.

Physiatry module was developed taking into account the software previously used, allowing to determine the processes of registration of patients, which is related to the module from doctors that allows record diagnoses and send medical reports, regarding the Department of statistics has compiled the information corresponding to the medical data generated and the need for therapies.

Physiatry system today is integrated with the statistic and doctors' offices, which gave solutions for several problems in this PFC.

# ÍNDICE DE CONTENIDOS

CERTIFICACIÓN DEL DIRECTOR .....	I
AUTORÍA .....	II
CARTA DE AUTORIZACIÓN DE TESIS POR PARTE DEL AUTOR .....	III
AGRADECIMIENTO .....	IV
DEDICATORIA .....	V
CESIÓN DE DERECHOS .....	VI
A. TÍTULO .....	VII
B. RESUMEN .....	VIII
SUMARY .....	IX
ÍNDICE DE CONTENIDOS .....	X
ÍNDICE DE FIGURAS .....	XVI
ÍNDICE DE TABLAS .....	XVII
GLOSARIO .....	XX
C. INTRODUCCIÓN .....	1
D. REVISIÓN DE LITERATURA O MARCO TEÓRICO .....	3
<b>CAPÍTULO I</b> .....	<b>3</b>
1. RUBY ON RAILS EN EL DESARROLLO DE APLICACIONES WEB .....	3
1.1 <i>Qué es Ruby?</i> .....	3
1.2 <i>Introducción</i> .....	3
1.3 <i>Características de Ruby</i> .....	4
1.4 <i>Por qué usar Ruby?</i> .....	5
1.5 <i>Ventaja y Desventajas Frente a otros Lenguajes</i> .....	5
1.6 <i>Ruby on Rails como herramienta de desarrollo de aplicaciones web</i> .....	6
1.6.1 <i>Introducción</i> .....	6
1.6.2 <i>Arquitectura de Ruby on Rails</i> .....	7
1.7 <i>Casos de éxito de aplicaciones desarrolladas con RoR</i> .....	11
<b>CAPÍTULO II</b> .....	<b>13</b>
2. METODOLOGÍA DE DESARROLLO XP .....	13
2.1 <i>Programación Extrema (Extreme Programming, XP)</i> .....	13
2.2 <i>Roles XP</i> .....	13
2.2.1 <i>Programador</i> .....	13

2.2.2	Cliente .....	14
2.2.3	Encargado de pruebas (Tester) .....	14
2.2.4	Encargado de seguimiento (Tracker).....	14
2.2.5	Entrenador (Coach).....	14
2.2.6	Consultor.....	14
2.2.7	Gestor (Big boss) .....	14
2.2.8	Principios Básicos .....	15
2.3	Retroalimentación a escala fina .....	15
2.3.1.	Proceso continuo en lugar de por lotes. ....	16
2.3.2	Entendimiento compartido. ....	17
2.3.3.	Bienestar del programador.....	19
2.4.	Fases de la Metodología.....	19
2.4.1.	FASE 1: Planificación del Proyecto .....	20
2.4.2.	FASE 2: Diseño.....	22
2.4.3.	FASE 3: Desarrollo. ....	23
2.4.4.	FASE 4: Pruebas. ....	23
<b>CAPÍTULO III.....</b>		<b>24</b>
1.	HERRAMIENTAS Y LENGUAJES DE DESARROLLO. ....	24
3.1	JavaScript.....	24
3.2	JQuery.....	25
3.2.1.	Ventajas de JQuery.....	25
3.3	Coffeescript. ....	26
3.4	Hojas de estilo (CSS).....	27
3.5	Bootstrap Framework Frontend. ....	29
3.5.1.	Introducción.....	29
3.5.2.	Características .....	29
3.5.3.	Estructura y función. ....	29
3.5.3.1.	Sistema de cuadrilla y diseño sensible. ....	30
3.5.3.2.	Entendiendo la hoja de estilo CSS .....	30
3.5.3.3.	Componentes re-usables .....	31
3.6	Gestor de base de datos Mysql .....	31
3.6.2	Seguridad.....	31
3.6.3	Escalabilidad y límites.....	32
3.6.4	Ruby on Rails y Mysql.....	32

<b>E. MATERIALES Y MÉTODOS .....</b>	<b>33</b>
1. <i>MÉTODOS</i> .....	33
2. <i>Técnicas</i> .....	34
3. <i>METODOLOGÍA DE DESARROLLO DE SOFTWARE</i> .....	34
<b>F. RESULTADOS .....</b>	<b>37</b>
<b>FASE 1 .....</b>	<b>37</b>
1. <i>Planificación del Proyecto</i> .....	37
1.1 <i>Introducción</i> .....	37
1.1.1 <i>Propósito</i> .....	37
1.1.3 <i>Definiciones, siglas y abreviaciones.</i> .....	39
1.2. <i>Descripción global</i> .....	40
1.2.1. <i>Perspectiva del producto</i> .....	40
1.2.2. <i>Funciones del producto</i> .....	40
1.2.3. <i>Características del Usuario</i> .....	41
1.2.4. <i>Restricciones</i> .....	42
1.2.5. <i>Atención y dependencias</i> .....	42
1.2.6. <i>Prorratear los requisitos</i> .....	42
1.3. <i>Los requisitos específicos</i> .....	42
1.3.1. <i>Requerimientos Funcionales</i> .....	42
1.3.2. <i>Requerimientos No Funcionales</i> .....	45
1.3.3. <i>Equipo: Integrantes y Roles</i> .....	46
1.3.4. <i>Historias de Usuarios</i> .....	47
1.3.5. <i>Resumen de Historias de Usuario</i> .....	56
1.3.6. <i>Plan de Entregas</i> .....	57
1.3.7. <i>División en iteraciones.</i> .....	58
1.3.9. <i>Entregas Pequeñas</i> .....	58
<b>FASE 2 .....</b>	<b>59</b>
2. <i>Diseño</i> .....	59
2.1. <i>Diseños Simples</i> .....	59
2.2. <i>Glosario de Términos</i> .....	60
2.3. <i>Tarjetas CRC (Clase - Responsabilidad – Colaborador)</i> .....	61
2.4. <i>Modelo del Dominio</i> .....	64
2.5. <i>Modelo de Clases</i> .....	65
2.6 <i>Modelo de la Base de Datos.</i> .....	66

<b>FASE 3</b> .....	<b>67</b>
3. Desarrollo.....	67
3.1. Cliente Siempre Presente.....	67
3.2. Pruebas Unitarias.....	67
3.3. Programación por parejas.....	71
3.4. Integración.....	72
<b>FASE 4</b> .....	<b>72</b>
4. Pruebas.....	72
4.1. Implantación.....	72
4.2. Pruebas de Aceptación.....	72
4.3. Plan de Validación.....	101
4.3.1. Propósito.....	101
4.3.2. Introducción.....	101
4.3.2.1. Objetivos.....	101
4.3.2.2. Antecedentes.....	101
4.3.2.3. Alcances.....	101
4.3.2.4. Referencias.....	102
4.3.3. Elementos software a probar.....	102
4.3.4. Características a probar.....	103
4.3.5. Enfoque general de la prueba.....	103
4.3.5.1. Aspectos generales.....	103
4.3.5.1.1. En lo relativo a la documentación.....	103
4.3.5.1.2. En lo relativo a los encargados del desarrollo.....	103
4.3.5.1.3. En lo relativo a la confidencialidad.....	103
4.3.5.2. Enfoques.....	103
4.3.5.2.1. Enfoque para probar los accesos al sistema.....	103
4.3.5.2.2. Enfoque para probar registro de turnos de terapia.....	104
4.3.5.2.3. Enfoque para probar administración de terapeutas.....	104
4.3.5.2.4. Enfoque para probar administración de tratamientos y terapias.....	104
4.3.5.2.5. Enfoque para probar registro de diagnósticos.....	104
4.3.5.2.6. Enfoque para probar envío de pacientes a fisioterapia.....	104
4.3.5.2.7. Enfoque para probar generación de reportes por: terapeuta, certificado/paciente, factura.....	104
4.3.5.3. Amplitud de la prueba.....	104

4.3.6. Criterios de paso/fallo para cada elemento .....	104
4.3.7. Criterios de suspensión y requisitos de reanudación .....	104
4.3.8. Productos entregables .....	105
4.3.9. Necesidades de entorno .....	105
4.3.9.1. Hardware.....	105
4.3.9.1.1. Características del servidor .....	105
4.3.9.1.2. Características del computador .....	105
4.3.9.2. Software .....	105
4.3.9.3. Documentos .....	105
4.3.10. Responsabilidades en la organización y realización de pruebas .....	106
4.3.10.1. Responsable del proyecto .....	106
4.3.10.2. Grupo de usuarios del sistema del Hospital Básico 7 B.I Loja. ....	106
4.3.11. Pruebas de Funcionalidad: .....	106
4.3.12. Pruebas de Usabilidad.....	107
4.3.13. Pruebas de Seguridad: .....	108
4.3.14. Pruebas de Carga y Estrés.....	108
4.3.15. Resultados .....	108
<b>G. DISCUSIÓN .....</b>	<b>115</b>
1. Desarrollo de la Propuesta Alternativa .....	115
2. Valoración Técnica Económica Ambiental.....	118
2.1 Académica .....	118
2.2 Técnica .....	118
2.3 Económica .....	118
2.4 Valoración Ambiental .....	121
<b>H. CONCLUSIONES.....</b>	<b>122</b>
<b>I. RECOMENDACIONES .....</b>	<b>123</b>
<b>J. BIBLIOGRAFÍA .....</b>	<b>124</b>
<b>K. ANEXOS.....</b>	<b>126</b>
ANEXO 1: ENCUESTA REALIZADA AL SOBRE LA INTERFAZ DEL SISTEMA. ....	127
ANEXO 2: ENCUESTA REALIZADA SOBRE PROBLEMAS EN PROCESOS DEL SISTEMA. ....	129
ANEXO 3: ENCUESTA REALIZADA SOBRE DIFICULTAD EN PROCESOS DEL SISTEMA. ....	131
ANEXO 4: FOTOGRAFÍAS DEL PERSONAL DEL HOSPITAL UTILIZANDO EL SISTEMA IMPLANTANDO....	133
ANEXO 5: VALIDACIÓN HTML. ....	134
ANEXO 6: PRUEBAS DE CARGA Y ESTRÉS CON JMETER. ....	135

<b>1.1</b>	<b>PRUEBA DE CARGA.....</b>	<b>135</b>
<b>1.2</b>	<b>PRUEBA DE RENDIMIENTO.....</b>	<b>136</b>
<b>1.3</b>	<b>PRUEBA DE ESTRÉS.....</b>	<b>137</b>
	ANEXO 7: CERTIFICACIÓN DE TRADUCCIÓN DEL RESUMEN. ....	139
	ANEXO 8: CERTIFICACIÓN DE HABER IMPLANTADO EL SISTEMA EN EL HOSPITAL BÁSICO 7BI LOJA. ....	140
	ANEXO 9: LICENCIAS. ....	141
	ANEXO 10: ANTEPROYECTO.....	<b>¡ERROR! MARCADOR NO DEFINIDO.</b>



## ÍNDICE DE FIGURAS

Figura 1. Arquitectura de una aplicación desarrollada con RoR [4] .....	8
Figura 2. Componente de Rails y MVC [5] .....	8
Figura 3. Fases de la Metodología XP [10]. .....	19
Figura 4 . Código para cargar librerías JS al proyecto .....	25
Figura 5. Ejemplo de sintaxis Coffeescript .....	26
Figura 6. Código para activación de tooltip Bootstrap. ....	27
Figura 7. Tooltip Bootstrap.....	27
Figura 8. Código css aplicado a tablas de la aplicación. ....	28
Figura 9. Estructura de archivos de Bootstrap. ....	30
Figura 10. Componentes gráficos de Bootstrap. ....	31
Figura 11. Conexión a la base de datos desde RoR. ....	32
Figura 12. Análisis de encuesta de interfaz.....	110
Figura 13. Análisis de problemas del módulo de fisioterapia. ....	112
Figura 14. Análisis de dificultad módulo de central de enfermería .....	114

## ÍNDICE DE TABLAS

TABLA I. COMPARATIVA ENTRE FRAMEWORKS DE DESARROLLO WEB RUBY ..	6
TABLA II: PERFILES DE USUARIOS.....	41
TABLA III: PERMISOS A TODOS LOS USUARIOS .....	42
TABLA IV: PERMISOS DEL ADMINISTRADOR.....	43
TABLA V: PERMISOS HA ADMINISTRADOR DE FISIATRÍA.....	43
TABLA VII: PERMISOS HA ADMINISTRADOR DE ESTADÍSTICA.....	44
TABLA VIII: PERMISOS DEL AUXILIAR DE ESTADÍSTICA. ....	44
TABLA VIII: PERMISOS DEL FISIATRA.....	44
TABLA IX: PERMISOS DEL DOCTOR.....	45
TABLA X: REQUERIMIENTOS NO FUNCIONALES .....	45
TABLA XI: EQUIPO, INTEGRANTES Y ROLES DEL PROYECTO .....	46
TABLA XII: HISTORIA DE USUARIO: ADMINISTRACIÓN USUARIOS .....	47
TABLA XIII: HISTORIA DE USUARIO REGISTRO TURNO DE TERAPIA. ....	48
TABLA XIV: HISTORIA DE USUARIO: CALENDARIO DE TURNOS .....	49
TABLA XV: HISTORIA DE USUARIO: ADMINISTRACIÓN DE TERAPISTAS .....	50
TABLA XVI: HISTORIA DE USUARIO: REPORTES DE FISIATRÍA.....	51
TABLA XVII: HISTORIA DE USUARIO ADMINISTRACIÓN DE TRATAMIENTOS Y TERAPIAS.....	53
TABLA XVIII: HISTORIA DE USUARIO: ADMINISTRACIÓN DE HORARIOS.....	54
TABLA XIX: HISTORIA DE USUARIO: ENVIAR A FISIATRÍA PACIENTE .....	55
TABLA XX: RESUMEN DE HISTORIAS DE USUARIO .....	56
TABLA XXI: PLAN DE ENTREGAS.....	57
TABLA XXII: VELOCIDAD DEL PROYECTO .....	58
TABLA XXIII: ITERACIONES.....	59
TABLA XXIV: GLOSARIO DE TÉRMINOS .....	60
TABLA XXV: TARJETAS CRC PERSONAL .....	61
TABLA XXVI: TARJETAS CRC TURNO.....	61
TABLA XXVII: TARJETAS CRC NECESITA_TERAPIA.....	61
TABLA XXVIII: TARJETAS CRC CONSULTA_EXTERNA_MORBILIDAD.....	61
TABLA XXIX: TARJETAS CRC ASIGNAR_HORARIO .....	62

TABLA XXX: TARJETAS CRC HORARIO .....	62
TABLA XXXI: TARJETAS CRC TRATAMIENTO_REGISTRO .....	62
TABLA XXXII: TARJETAS CRC RESULTADO_TRATAMIENTO.....	63
TABLA XXXIII: TARJETAS CRC TRATAMIENTO .....	63
TABLA XXXIV: PRUEBA UNITARIA 1 .....	68
TABLA XXXV: PRUEBA UNITARIA 2.....	68
TABLA XXXVI: PRUEBA UNITARIA 3.....	69
TABLA XXXVII: PRUEBA UNITARIA 4.....	69
TABLA XXXVIII: PRUEBA UNITARIA 5.....	70
TABLA XXXIX: PRUEBA UNITARIA 6.....	70
TABLA XL: PRUEBA UNITARIA 7.....	71
TABLA XLI: PRUEBA UNITARIA 10.....	71
TABLA XLII: CASO DE PRUEBA 1: ADMINISTRACIÓN USUARIOS.....	73
TABLA XLIII: CASO DE PRUEBA 2: ADMINISTRACIÓN USUARIOS.....	74
TABLA XLIV: CASO DE PRUEBA 3: ADMINISTRACIÓN USUARIOS .....	75
TABLA XLV: CASO DE PRUEBA 4: ADMINISTRACIÓN USUARIOS .....	76
TABLA XLVI: CASO DE PRUEBA 5: ADMINISTRACIÓN USUARIOS .....	77
TABLA XLVII: CASO DE PRUEBA 6: REGISTRO DE TURNOS DE TERAPIA .....	78
TABLA XLVIII: CASO DE PRUEBA 7: REGISTRO DE TURNOS DE TERAPIA .....	79
TABLA XLIX: CASO DE PRUEBA 8: CALENDARIO DE TURNOS .....	80
TABLA L: CASO DE PRUEBA 9: CALENDARIO DE TURNOS .....	81
TABLA LI: CASO DE PRUEBA 10: ADMINISTRADOR DE TERAPISTAS.....	82
TABLA LII: CASO DE PRUEBA 11: ADMINISTRADOR DE TERAPISTAS.....	83
TABLA LIII: CASO DE PRUEBA 12: ADMINISTRADOR DE TERAPISTAS.....	84
TABLA LIV: CASO DE PRUEBA 13: ADMINISTRADOR DE TERAPISTAS .....	85
TABLA LV: CASO DE PRUEBA 14: ADMINISTRADOR DE TERAPISTAS .....	86
TABLA LVI: CASO DE PRUEBA 15: REPORTES DE FISIATRÍA .....	87
TABLA LVII: CASO DE PRUEBA 16: REPORTES DE FISIATRÍA .....	88
TABLA LVIII: CASO DE PRUEBA 17: REPORTES DE FISIATRÍA.....	89
TABLA LIX: CASO DE PRUEBA 18: ADMINISTRACIÓN DE TRATAMIENTOS Y TERAPIAS.....	90

TABLA LX: CASO DE PRUEBA 19: ADMINISTRACIÓN DE TRATAMIENTOS Y TERAPIAS.....	91
TABLA LXI: CASO DE PRUEBA 20: ADMINISTRACIÓN DE TRATAMIENTOS Y TERAPIAS.....	92
TABLA LXII: CASO DE PRUEBA 21: ADMINISTRACIÓN DE TRATAMIENTOS Y TERAPIAS.....	93
TABLA LXIII: CASO DE PRUEBA 22: ADMINISTRACIÓN DE TRATAMIENTOS Y TERAPIAS.....	94
TABLA LXIV: CASO DE PRUEBA 23: ADMINISTRACIÓN DE HORARIOS.....	95
TABLA LXV: CASO DE PRUEBA 24: ADMINISTRACIÓN DE HORARIOS.....	96
TABLA LXVI: CASO DE PRUEBA 25: ADMINISTRACIÓN DE HORARIOS.....	97
TABLA LXVII: CASO DE PRUEBA 26: ADMINISTRACIÓN DE HORARIOS.....	98
TABLA LXVIII: CASO DE PRUEBA 27: ADMINISTRACIÓN DE HORARIOS.....	99
TABLA LXIX: CASO DE PRUEBA 28: ENVIAR A FISIATRÍA PACIENTE.....	100
TABLA LXX: FUNCIONALIDAD POR USUARIOS.....	102
TABLA LXXI: TIEMPO Y COSTO DEL PROYECTO.....	119
TABLA LXXII. PRESUPUESTO TECNOLÓGICO.....	119
TABLA LXXIII: PRESUPUESTO DE MATERIALES.....	120
TABLA LXXIV: PRESUPUESTO SERVICIOS.....	120
TABLA LXXV: RESUMEN DE PRESUPUESTO.....	121

## GLOSARIO

Siglas	Significado
PFC	Proyecto de Fin de carrera
RoR	Ruby on Rails
CTEXM	Consulta Externa por morbilidad
CTEXP	Consulta Externa preventiva
HTML	Lenguaje de marcas de hipertexto en inglés HyperText Markup Language
CSS	Hojas de estilo en cascada o en inglés Cascading Style Sheets
AJAX	Es un acrónimo de Asynchronous JavaScript + XML, que se puede traducir como "JavaScript asíncrono + XML"
XML	Lenguaje de marcas extensible, en inglés Extensible Markup Language
DOM	Modelo de Objetos de Documento, en inglés Document Object Model
XHTML	HTML expresado como XML, en inglés extensible HyperText Markup Language
JSON	Acrónimo de JavaScript Object Notation
SHA	Algoritmo de Hash Seguro, en inglés Secure Hash Algorithm
ERS	Especificación de Requisitos Software

## **c. Introducción**

Las unidades de salud son un eje primordial para el desarrollo de una nación, las mismas que se destacan en diferentes campos ya sean a nivel privado o dependiente del Estado, brindando servicios y atención a la colectividad, muchas de las cuales no aprovechan el uso de las nuevas tecnologías que día a día avanzan de manera escalar, dentro de estas unidades de salud se encuentra el Hospital Básico 7 BI Loja, regentado por el Estado y con cobertura a nivel Nacional, brindando servicios a la provincia de Loja.

Es así, que la creciente afluencia de personas a servicios médicos, conlleva a mejorar tanto la calidad de atención al cliente, y el manejo de registros médicos que se generan, por lo cual la ausencia de un sistema computacional para el servicio de Fisiatría en el Hospital Básico 7 BI Loja, no permite que se lleve con claridad el manejo de horarios, la emisión de turnos y registros que se realizan en una terapia, por ende es la base para desarrollar un sistema computacional para la institución y que se integre con el sistema web con el que cuenta hospital, con la finalidad de llevar registro de pacientes, terapeutas y procesos necesarios para realizar una terapia.

Basado en esto y en avances tecnológicos, como lo son los sistemas integrados, que ofrecen el manejo de información de manera ordenada e inmediata, se tiene como objetivo el desarrollo: “Análisis, Diseño e Implantación del módulo de Fisiatría al sistema web “SISFA” para el Hospital Básico 7 B.I Loja”.

En la sección INTRODUCCIÓN, se explica la importancia y estructura del PFC, en la METODOLOGÍA, se detalla cada uno de los métodos de investigación tanto científicos, experimentales y ciertas técnicas investigativas, además de las fases que contempla la metodología.

En la REVISIÓN DE LA LITERATURA, se presenta información de los diferentes aspectos que se tomaron en cuenta para poder llevarlo a cabo, se dividió en tres capítulos: En el primero se encuentra información referente al lenguaje de programación y framework que fueron la base para el desarrollo del software. En el segundo capítulo se describe la metodología utilizada XP (Extreme-Programming

Programación-Extrema), la cual permite una planificación adecuada de las tareas a ejecutarse. En el tercer capítulo encontramos una descripción de las herramientas utilizadas para la codificación del software.

RESULTADOS, contiene el desarrollo de la metodología aplicada la cual representa el trabajo realizado en la elaboración del proyecto, en la DISCUSIÓN, se describe el cumplimiento de los objetivos planteados, así también una evaluación técnica, económica, ambiental sobre el trabajo realizado, para terminar con las CONCLUSIONES Y RECOMENDACIONES que describe los puntos más relevantes basado en las experiencias del autor.

## **d. REVISIÓN DE LITERATURA O MARCO TEÓRICO**

### **CAPÍTULO I**

#### **1. Ruby on Rails en el desarrollo de Aplicaciones Web.**

##### **1.1 Qué es Ruby?**

Ruby es un Lenguaje de Programación de alto nivel orientado a objetos en el cual hasta los números y los caracteres literales son objetos y tienen métodos de su clase, que pueden llamarse normalmente, este es rápido y sencillo, por lo cual su línea de aprendizaje no es tan extensa a diferencia de otros lenguajes como: java, python, php, etc.

##### **1.2 Introducción**

Ruby, es un lenguaje creado por Yukihiro "Matz" Matsumoto, quien empezó a trabajar en Ruby el 24 de febrero de 1993, y lo presentó al público en el año 1995.

El cual ha manifestado que está “tratando de hacer que Ruby sea natural, no simple”, de una forma que se asemeje a la vida real, ruby es simple en apariencia, pero complejo por dentro, como el cuerpo humano.

Además que está diseñado para la productividad y la diversión del desarrollador, siguiendo los principios de una buena interfaz de usuario, el diseño de sistemas necesita enfatizar las necesidades humanas más que las de la máquina.

En Ruby, todo es un objeto. Se le puede asignar propiedades y acciones a toda información y código. La programación orientada a objetos llama a las propiedades *variables de instancia* y las acciones son conocidas como *métodos*. La orientación a objetos pura de Ruby se suele demostrar con un simple código que aplica una acción a un número. En muchos lenguajes, los números y otros tipos primitivos no son objetos. Ruby sigue la influencia del lenguaje Smalltalk pudiendo



poner métodos y variables de instancia a todos sus tipos de datos. Esto facilita el uso de Ruby, porque las reglas que se aplican a los objetos son aplicables a todo Ruby [1].

La última versión estable de la rama 1.8 es la 1.8.7\_p248, de la rama 1.9 es la 1.9.1\_p378. La versión en 1.9 que incorpora mejoras sustanciales en el rendimiento del lenguaje, que se espera queden reflejadas en la próxima versión estable de producción del lenguaje, Ruby 1.9.0.1 Diferencias en rendimiento entre la actual implementación de Ruby (1.8.6) y otros lenguajes de programación más arraigados han llevado al desarrollo de varias máquinas virtuales para Ruby. Entre éstas se encuentra JRuby, un intento de llevar Ruby a la plataforma Java, y Rubinius, un intérprete modelado basado en las máquinas virtuales de Smalltalk. Los principales desarrolladores han apoyado la máquina virtual proporcionada por el proyecto YARV, que se fusionó en el árbol de código fuente de Ruby el 31 de diciembre de 2006, y se dio a conocer como Ruby 1.9.

### **1.3 Características de Ruby**

Al ser Ruby un lenguaje orientado a objetos y bastante flexivo posee muchas características las cuales se mencionan a continuación:

#### **Lenguaje de guiones interpretado:**

- Posibilidad de realizar directamente llamadas al sistema operativo
- Potentes operaciones sobre cadenas de caracteres y expresiones regulares
- Retroalimentación inmediata durante el proceso de desarrollo.

#### **Programación orientada a objetos:**

- Todo es un objeto
- Clases, herencia, métodos.
- Mixins por módulos.
- Iteradores y cierres.

#### **Rápido y sencillo:**

- Son innecesarias las declaraciones de variables.

- Las variables no tienen tipo
- La sintaxis es simple y consistente.
- La gestión de la memoria es automática.

#### **1.4 Por qué usar Ruby?**

Ruby es diferente a los demás lenguajes orientados a objetos que conocemos, combina con éxito la elegancia conceptual del Smalltalk, la facilidad de uso y aprendizaje de Python, el pragmatismo de Perl, y las características y funcionalidades de muchos otros lenguajes. Con esto podemos decir que el uso de este lenguaje esta con lo mejor de cada uno de ellos, haciendo que las fortalezas de estos lenguajes sean las fortalezas de Ruby, para el desarrollo de este proyecto se hizo uso de la versión 2.0.

Además podemos añadir métodos, y parámetros a cualquier clase que queramos incluso a las clases básicas de este lenguaje, haciendo que Ruby sea escalable a un nivel muy superior de los otros lenguajes.

El lenguaje contempla los errores que se puede tener con un sistema completo de excepciones que permite lanzar, capturar, preparar, y ejecutar cualquier programa con errores.

Podemos hablar que ya es un lenguaje con extensa documentación ya sea en su página Web, o en otras páginas externas, cuenta con innumerables libros (desde cookbooks), libros de referencia y guías completas incluso escritas por el propio autor de Ruby. Y para nuestra comunidad de habla hispana podemos decir que ya existe un grupo llamado rubies, que está ayudando a la traducción de varios documentos para aprender este lenguaje.

Por último, Ruby tiene bastantes paquetes disponibles, como para las interfaces gráficas, funciones de red, para utilizar hilos, inclusive existe una para cargar paquetes de otros lenguajes como Python [2].

#### **1.5 Ventaja y Desventajas Frente a otros Lenguajes**

Como todo lenguaje este tienes sus puntos fuertes y débiles frente a otros lenguajes, los cuales se ven reflejados en la siguiente tabla comparativa:

TABLA I. COMPARATIVA ENTRE FRAMEWORKS DE DESARROLLO WEB RUBY

Características	RUBY	PYTHON	PHP
Línea de aprendizaje	Corta de 2 a 3 meses	Corta de 2 a 3 meses	Media de 4 a 6 meses
Propósito	Fue diseñado para hacer la programación fácil y flexible	Fue diseñado para aumentar la productividad	Fue diseñado para desarrollo web de páginas dinámicas
Influencia de lenguajes	ADA EIFFEL C++ LISP CLU PERL DYLAN PHYTON	ABC, ICON ALGOL68, JAVA C++ PERL	C, PERL, TCL JAVA, C++
Año de lanzamiento	1995	1991	1991
Framework de desarrollo	Ruby on Rails, Sinatra, Maverick	Django, Pylons, Turbo Gears	Symfony, ZendFramework, CakePHP

A pesar que Ruby es un lenguaje joven frente a otros, cuenta con la influencia de varios lenguajes y ha cumplido con el propósito por el cual fue realizado ya que su línea de aprendizaje es corta y además cuenta con una gran comunidad de desarrolladores.

## 1.6 Ruby on Rails como herramienta de desarrollo de aplicaciones web

### 1.6.1 Introducción

Ruby on Rails, también conocido como RoR o Rails es un framework de aplicaciones web de código abierto escrito en el lenguaje de programación Ruby, siguiendo el paradigma de la arquitectura Modelo Vista Controlador (MVC). Trata de combinar la simplicidad con la posibilidad de desarrollar aplicaciones del mundo real escribiendo menos código que con otros frameworks y con un mínimo de configuración. El

lenguaje de programación Ruby permite la meta programación, de la cual Rails hace uso, lo que resulta en una sintaxis que muchos de sus usuarios encuentran muy legible. Rails se distribuye a través de RubyGems, que es el formato oficial de paquete y canal de distribución de bibliotecas y aplicaciones Ruby [3].

## **Filosofía**

Los principios fundamentales de Ruby on Rails incluyen No te repitas (del inglés Don't repeat yourself, DRY) y Convención sobre configuración.

No te repitas significa que las definiciones deberían hacerse una sola vez. Dado que Ruby on Rails es un framework de pila completa, los componentes están integrados de manera que no hace falta establecer puentes entre ellos. Por ejemplo, en ActiveRecord, las definiciones de las clases no necesitan especificar los nombres de las columnas; Ruby puede averiguarlos a partir de la propia base de datos, de forma que definirlos tanto en el código como en el programa sería redundante.

Así, cuando se diseña una aplicación partiendo de cero sin una base de datos pre existente, el seguir las convenciones de Rails significa usar menos código (aunque el comportamiento puede ser configurado si el sistema debe ser compatible con un sistema heredado anterior).

### **1.6.2 Arquitectura de Ruby on Rails**

La Figura 1 muestra la arquitectura de una aplicación desarrollada en Ruby on Rails en la que se puede observar los módulos que se manejan y el flujo de operaciones que conllevan a la ejecución de este tipo de aplicaciones.

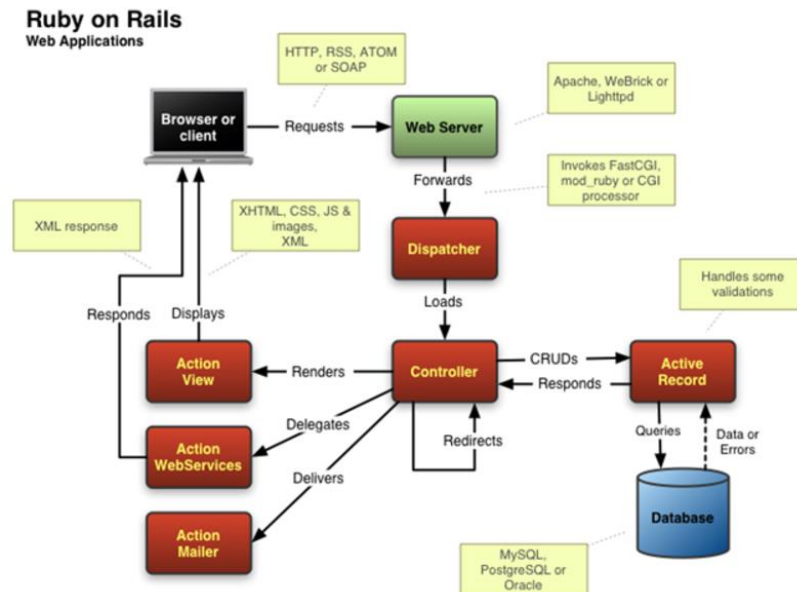


Figura 1. Arquitectura de una aplicación desarrollada con RoR [4]

El modelo que maneja es Modelo, Vista, Controlador, en el cual Rails realiza las acciones AccionView, AccionRecord, AccionController cada uno correspondiente al MCV como lo indica la Figura 2:

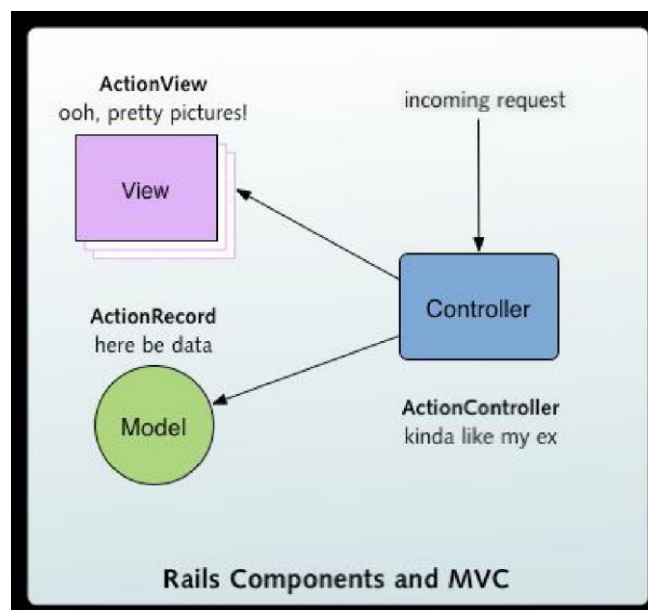


Figura 2. Componente de Rails y MVC [5]

Se realiza una breve descripción de cada uno de estos módulos y de las operaciones que realiza cada uno.

## **1. Modelo**

Las aplicaciones web orientadas a objetos sobre bases de datos, el Modelo consiste en las clases que representan a las tablas de la base de datos.

En Ruby on Rails, las clases del Modelo son gestionadas por ActiveRecord. Por lo general, lo único que tiene que hacer el programador es heredar de la clase ActiveRecord: Base, y el programa averiguará automáticamente qué tabla usar y qué columnas tiene [6].

Las definiciones de las clases también detallan las relaciones entre clases con sentencias de mapeo objeto relacional. Por ejemplo, si la clase "Persona tiene una definición has\_many: facturas, y existe una instancia de persona llamada a entonces p.facturas devolverá un array con todos los objetos Factura cuya columna persona\_id (en la tabla comentarios) sea igual a p.id.

Las rutinas de validación de datos (p.e. validates\_uniqueness\_of: checksum) y las rutinas relacionadas con la actualización (p.e. after\_destroy:borrar\_archivo, before\_update:actualizar\_detalle) también se especifican e implementan en la clase del modelo.

Lo que realiza el modelo al final no es más que las tablas de la base de datos y sus relaciones, además de las migraciones que son los cambios que se realizan en la BD.

## **2. Vista**

La Vista es la lógica de visualización, o cómo se muestran los datos de las clases del Controlador. Con frecuencia en las aplicaciones web la vista consiste en una cantidad mínima de código incluido en HTML.

El método que se emplea en Rails por defecto es usar Ruby Embebido (archivos.rhtml, desde la versión 2.x en adelante de RoRarchivos.html.erb), que son básicamente fragmentos de código HTML con algo de código en Ruby, siguiendo una sintaxis

similar a JSP. También pueden construirse vistas en HTML y XML con Builder o usando el sistema de plantillas Liquid.

Es necesario escribir un pequeño fragmento de código en HTML para cada método del controlador que necesita mostrar información al usuario. El "maquetado" o distribución de los elementos de la página se describe separadamente de la acción del controlador y los fragmentos pueden invocarse unos a otros [6].

### **3. Controlador**

En MVC, las clases del Controlador responden a la interacción del usuario e invocan a la lógica de la aplicación, que a su vez manipula los datos de las clases del Modelo y muestra los resultados usando las Vistas. En las aplicaciones web basadas en MVC, los métodos del controlador son invocados por el usuario usando el navegador web.

La implementación del Controlador es manejada por el ActionPack de Rails, que contiene la clase ApplicationController. Una aplicación Rails simplemente hereda de esta clase y define las acciones necesarias como métodos, que pueden ser invocados desde la web, por lo general en la forma:

`http://aplicacion/ejemplo/metodo`

Esta invoca a `EjemploController#método`, y presenta los datos usando el archivo de plantilla `/app/views/ejemplo/método.html.erb`, a no ser que el método redirija a algún otro lugar.

Rails también proporciona andamiaje, que puede construir rápidamente la mayor parte de la lógica y vistas necesarias para realizar las operaciones más frecuentes [8].

### **4. Otros módulos**

Además, Rails ofrece otros módulos, como Action Mailer (para enviar correo electrónico) o Active Resource que proporciona la infraestructura necesaria para crear de manera sencilla recursos REST, algo por lo que apuesta claramente Rails en sus últimas versiones desplazando así a otros modelos como SOAP y XML-RPC a los que se les daba soporte en versiones anteriores mediante Action Web Service.

## **1.7 Casos de éxito de aplicaciones desarrolladas con RoR.**

Actualmente el creciente desarrollo de aplicaciones web y móviles, ha obligado a las empresas a migrar sus sistemas a nuevas tecnologías, buscando una plataforma que no sea muy compleja, pero lo suficientemente segura y flexible, es así que muchas de estas empresas han optado por Ruby on Rails como es el caso de:

### **Twitter:**

Una de las redes sociales más grande del mundo, actualmente se encuentra funcionando bajo el uso de muchos lenguajes pero el core o corazón de su estructura está desarrollada en RoR.

### **Github:**

Es un servicio de creación de repositorios que ayuda alojar proyectos de forma pública y privada además de brinda control de versiones, es una de las herramientas más utilizadas por los desarrollares del mundo, creado por la empresa hitgub Inc., lanzado en el año 2008.

### **Shopify:**

Es una solución de sitio web de comercio electrónico que le permite vender en línea al darle todo lo que necesita para crear su tienda online, creado por la empresa shopify. Inc. en el año 2006.

### **Hospitium:**

Software para la gestión de refugios de animales, rescates, creado por travis Berry y Jinna Adansom en el año de 2010.

### **How are you?:**

Es una red social por la salud que se maneja en ciertos hospitales que pone a los pacientes en conocimiento de sus exámenes, medicamentos, dosis, citas médicas, además de poner en contacto con su médico en tiempo real, creada por la empresa Cambridge healthcare en el año 2013.



Existen muchas más aplicaciones web que han sido desarrollados en Ruby, pero se cree conveniente mencionar algunos de los que más sobresaliente.

# **CAPÍTULO II**

## **2. Metodología de desarrollo XP**

### **2.1 Programación Extrema (Extreme Programming, XP)**

Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo.

XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

Los principios y prácticas son de sentido común pero llevadas al extremo, de ahí proviene su nombre. Kent Beck, el padre de XP, describe la filosofía de XP en [7] sin cubrir los detalles técnicos y de implantación de las prácticas. Posteriormente, otras publicaciones de experiencias se han encargado de dicha tarea. A continuación presentaremos las características esenciales de XP organizadas en los tres apartados siguientes: historias de usuario, roles, proceso y prácticas.

### **2.2 Roles XP.**

Aunque en otras fuentes de información aparecen algunas variaciones y extensiones de roles XP, en este apartado describiremos los roles de acuerdo con la propuesta original de Beck.

#### **2.2.1 Programador**

El programador escribe las pruebas unitarias y produce el código del sistema. Debe existir una comunicación y coordinación adecuada entre los programadores y otros miembros del equipo.

### **2.2.2 Cliente**

El cliente escribe las historias de usuario y las pruebas funcionales para validar su implementación. Además, asigna la prioridad a las historias de usuario y decide cuáles se implementan en cada iteración centrándose en aportar mayor valor al negocio. El cliente es sólo uno dentro del proyecto pero puede corresponder a un interlocutor que está representando a varias personas que se verán afectadas por el sistema.

### **2.2.3 Encargado de pruebas (Tester)**

El encargado de pruebas ayuda al cliente a escribir las pruebas funcionales. Ejecuta las pruebas regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas.

### **2.2.4 Encargado de seguimiento (Tracker)**

El encargado de seguimiento proporciona realimentación al equipo en el proceso XP. Su responsabilidad es verificar el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado, comunicando los resultados para mejorar futuras estimaciones.

También realiza el seguimiento del progreso de cada iteración y evalúa si los objetivos son alcanzables con las restricciones de tiempo y recursos presentes. Determina cuándo es necesario realizar algún cambio para lograr los objetivos de cada iteración.

### **2.2.5 Entrenador (Coach)**

Es responsable del proceso global. Es necesario que conozca a fondo el proceso XP para proveer guías a los miembros del equipo de forma que se apliquen las prácticas XP y se siga el proceso correctamente.

### **2.2.6 Consultor**

Es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto. Guía al equipo para resolver un problema específico.

### **2.2.7 Gestor (Big boss)**

Es el vínculo entre clientes y programadores, ayuda a que el equipo trabaje efectivamente creando las condiciones adecuadas. Su labor esencial es de coordinación.

### 2.2.8 Principios Básicos

La Programación Extrema se basa en 12 principios básicos agrupados en cuatro categorías, a continuación se presenta cada una de ellas, detallada de cómo se llevó a cabo su cumplimiento en el desarrollo del PFC:

### 2.3 Retroalimentación a escala fina

La primera de las cuatro categorías principales en las que se agrupan los 12 principios básicos de la programación extrema, la que consiste en cuatro secciones:

1. **El principio de pruebas:** Se tiene que establecer un período de pruebas de aceptación del programa. Dentro del desarrollo del sistema se utilizó la herramienta Unit\_Test del framework Codelgniter, en la fase 3 de resultados se encuentra el detalle.
2. **Proceso de planificación:** Es un espacio frecuente de comunicación entre el cliente y los programadores. El equipo técnico realiza una estimación del esfuerzo requerido para la implementación de las historias de usuario y los clientes deciden sobre el ámbito y tiempo de las entregas y de cada iteración. Esta práctica se puede ilustrar como un juego, donde existen dos tipos de jugadores: Cliente y Programador. El cliente establece la prioridad de cada historia de usuario, de acuerdo con el valor que aporta para el negocio. Los programadores estiman el esfuerzo asociado a cada historia de usuario. Se ordenan las historias de usuario según prioridad y esfuerzo, y se define el contenido de la entrega y/o iteración, apostando por enfrentar lo de más valor y riesgo cuanto antes. Este juego se realiza durante la planificación de la entrega, en la planificación de cada iteración y cuando sea necesario reconducir el proyecto.
3. **El cliente en el sitio:** Durante el desarrollo del sistema se mantuvo la comunicación con el usuario permitiendo determinar los requerimientos, definir la funcionalidad, señalar las prioridades. El cliente tiene que estar presente y disponible todo el tiempo para el equipo. Gran parte del éxito del proyecto XP se debe a que es el cliente quien conduce constantemente el trabajo hacia lo que aportará mayor valor de negocio y los programadores pueden resolver de manera inmediata cualquier duda asociada. La comunicación oral es más

efectiva que la escrita, ya que esta última toma mucho tiempo en generarse y puede tener más riesgo de ser mal interpretada. En [8] Jeffries indica que se debe pagar un precio por perder la oportunidad de un cliente con alta disponibilidad. Algunas recomendaciones propuestas para dicha situación son las siguientes: intentar conseguir un representante que pueda estar siempre disponible y que actúe como interlocutor del cliente, contar con el cliente al menos en las reuniones de planificación, establecer visitas frecuentes de los programadores al cliente para validar el sistema, anticiparse a los problemas asociados estableciendo llamadas telefónicas frecuentes y conferencias, reforzando el compromiso de trabajo en equipo.

- 4. Programación en parejas:** Toda la producción de código debe realizarse con trabajo en parejas de programadores. Según Cockburn y Williams en un estudio realizado para identificar los costos y beneficios de la programación en parejas [7], las principales ventajas de introducir este estilo de programación son: muchos errores son detectados conforme son introducidos en el código (inspecciones de código continuas), por consiguiente la tasa de errores del producto final es más baja, los diseños son mejores y el tamaño del código menor (continúa discusión de ideas de los programadores), los problemas de programación se resuelven más rápido, se posibilita la transferencia de conocimientos de programación entre los miembros del equipo, varias personas entienden las diferentes partes sistema, los programadores conversan mejorando así el flujo de información y la dinámica del equipo, y finalmente, los programadores disfrutan más su trabajo. Dichos beneficios se consiguen después de varios meses de practicar la programación en parejas. En los estudios realizados por Cockburn y Williams este lapso de tiempo varía de 3 a 4 meses.

### **2.3.1. Proceso continuo en lugar de por lotes.**

Integración continua: Cada pieza de código es integrada en el sistema una vez que esté lista. Así, el sistema puede llegar a ser integrado y construido varias veces en un mismo día. Todas las pruebas son ejecutadas y tienen que ser aprobadas para que el nuevo código sea incorporado definitivamente. La integración continua a menudo reduce la fragmentación de los esfuerzos de los desarrolladores por falta de comunicación sobre lo que puede ser reutilizado o compartido. Martin Fowler en [8]

afirma que el desarrollo de un proceso disciplinado y automatizado es esencial para un proyecto controlado, el equipo de desarrollo está más preparado para modificar el código cuando sea necesario, debido a la confianza en la identificación y corrección de los errores de integración.

**1. Refactorización:** La refactorización es una actividad constante de reestructuración del código con el objetivo de remover duplicación de código, mejorar su legibilidad, simplificarlo y hacerlo más flexible para facilitar los posteriores cambios. La refactorización mejora la estructura interna del código sin alterar su comportamiento externo [8]. Robert Martin señala que el diseño del sistema de software es una cosa viviente. No se puede imponer todo en un inicio, pero en el transcurso del tiempo este diseño evoluciona conforme cambia la funcionalidad del sistema. Para mantener un diseño apropiado, es necesario realizar actividades de cuidado continuo durante el ciclo de vida del proyecto.

De hecho, este cuidado continuo sobre el diseño es incluso más importante que el diseño inicial. Un concepto pobre al inicio puede ser corregido con esta actividad continua, pero sin ella, un buen diseño inicial se degradará.

**2. Entregas pequeñas:** La idea es producir rápidamente versiones del sistema que sean operativas, aunque obviamente no cuenten con toda la funcionalidad pretendida para el sistema pero sí que constituyan un resultado de valor para el negocio. Una entrega no debería tardar más 3 meses.

### **2.3.2 Entendimiento compartido.**

**1. Diseño simple:** Se debe diseñar la solución más simple que pueda funcionar y ser implementada en un momento determinado del proyecto. La complejidad innecesaria y el código extra debe ser removido inmediatamente. Kent Beck dice que en cualquier momento el diseño adecuado para el software es aquel que: supera con éxito todas las pruebas, no tiene lógica duplicada, refleja claramente la intención de implementación de los programadores y tiene el menor número posible de clases y métodos.

**2. Metáfora:** El sistema fue desarrollado por módulos, cada módulo comprende

algunas funciones, de tal forma que se permita acoplar a cada tipo de usuario dentro del sistema. Estas funciones aparecieron al ir elaborando las historias de usuario, tomando en cuenta las tarjetas CRC, las mismas que ayudarán a definir las actividades durante el diseño del sistema. En XP no se enfatiza la definición temprana de una arquitectura estable para el sistema. Dicha arquitectura se asume evolutiva y los posibles inconvenientes que se generarían por no contar con ella explícitamente en el comienzo del proyecto se solventan con la existencia de una metáfora. El sistema es definido mediante una metáfora o un conjunto de metáforas compartidas por el cliente y el equipo de desarrollo. Una metáfora es una historia compartida que describe cómo debería funcionar el sistema. Martin Fowler en [6] explica que la práctica de la metáfora consiste en formar un conjunto de nombres que actúen como vocabulario para hablar sobre el dominio del problema. Este conjunto de nombres ayuda a la nomenclatura de clases y métodos del sistema.

3. **Propiedad colectiva del código:** un código con propiedad compartida. Nadie es el propietario de nada, todos son el propietario de todo. [9]. Cualquier programador puede cambiar cualquier parte del código en cualquier momento. Esta práctica motiva a todos a contribuir con nuevas ideas en todos los segmentos del sistema, evitando a la vez que algún programador sea imprescindible para realizar cambios en alguna porción de código.
4. **Estándar de codificación:** define la propiedad del código compartido así como las reglas para escribir y documentar el código y la comunicación entre diferentes piezas de código desarrolladas por diferentes equipos. Respecto a este punto no se encontró ningún problema, además fue desarrollado por una sola persona.

XP enfatiza la comunicación de los programadores a través del código, con lo cual es indispensable que se sigan ciertos estándares de programación (del equipo, de la organización u otros estándares reconocidos para los lenguajes de programación utilizados). Los estándares de programación mantienen el código legible para los miembros del equipo, facilitando los cambios.

### 2.3.3. Bienestar del programador.

La semana de 40 horas: Se debe trabajar un máximo de 40 horas por semana. No se trabajan horas extras en dos semanas seguidas. Si esto ocurre, probablemente está ocurriendo un problema que debe corregirse. El trabajo extra desmotiva al equipo. Los proyectos que requieren trabajo extra para intentar cumplir con los plazos suelen al final ser entregados con retraso. En lugar de esto se puede realizar el juego de la planificación para cambiar el ámbito del proyecto o la fecha de entrega.

### 2.4. Fases de la Metodología

La metodología XP consta de cuatro fases tal como lo indica la figura 2, para el desarrollo de la aplicación se utilizó cada una de ellas.

La relación entre el equipo de diseño, los que desarrollan el software y clientes es totalmente diferente al que se produce en las metodologías tradicionales, que se basaba en una fase de captura de los requisitos previa al desarrollo, y de una fase de validación posterior al mismo [10].

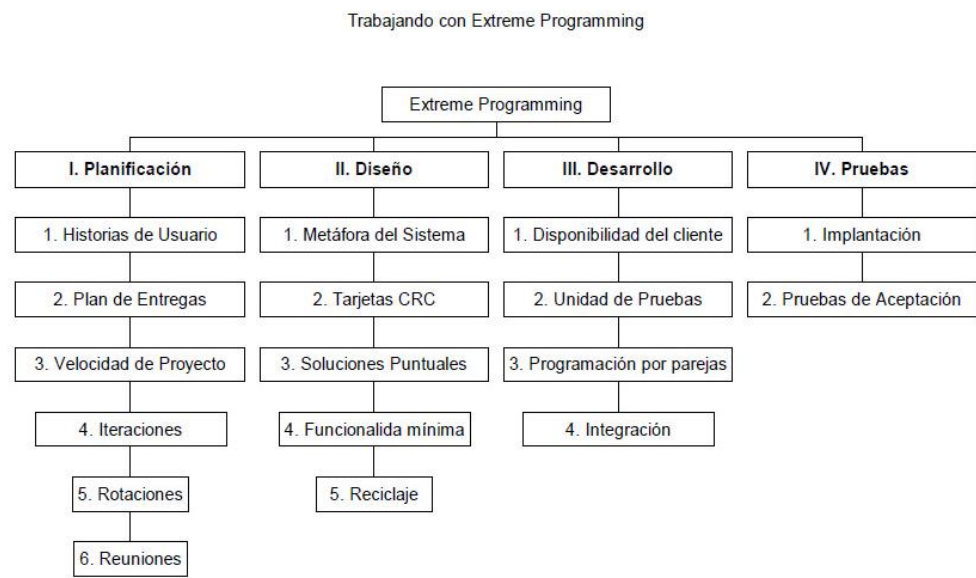


Figura 3. Fases de la Metodología XP [10].



## **2.4.1. FASE 1: Planificación del Proyecto**

### **1. Historias de Usuario:**

Las historias de usuario son la técnica utilizada en XP para especificar los requisitos del software. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales.

El tratamiento de las historias de usuario es muy dinámico y flexible, en cualquier momento historias de usuario pueden romperse, reemplazarse por otras más específicas o generales, añadirse nuevas o ser modificadas. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas [9].

Respecto de la información contenida en la historia de usuario, existen varias plantillas sugeridas pero no existe un consenso al respecto. En muchos casos sólo se propone utilizar un nombre y una descripción o sólo una descripción, más quizás una estimación de esfuerzo en días. Beck en su libro presenta un ejemplo de ficha (customer story and task card) en la cual pueden reconocerse los siguientes contenidos:

Fecha, tipo de actividad (nueva, corrección, mejora), prueba funcional, número de historia, prioridad técnica y del cliente, referencia a otra historia previa, riesgo, estimación técnica, descripción, notas y una lista de seguimiento con la fecha, estado cosas por terminar y comentarios.

Una de las interrogantes (que también se presenta cuando se utilizan casos de uso) es ¿cuál es el nivel de granularidad adecuado para una historia de usuario? La respuesta no es tajante. Jeffries en dice que depende de la complejidad del sistema, debe haber al menos una historia por cada característica importante, y propone realizar una o dos historias por programador por mes. Si se tienen menos, probablemente sea conveniente dividir las historias, si se tienen más lo mejor es disminuir el detalle y agruparlas. Para efectos de planificación, las historias pueden ser de una a tres semanas de tiempo de programación (para no superar el tamaño de una iteración).

No hay que preocuparse si en un principio no se identifican todas las historias de usuario. Al comienzo de cada iteración estarán registrados los cambios en las historias de usuario y según eso se planificará la siguiente iteración.

Las historias de usuario son descompuestas en tareas de programación y asignadas a los programadores para ser implementadas durante una iteración.

## **2. Planes de Entrega:**

En esta fase el cliente establece la prioridad de cada historia de usuario, y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Una entrega debería obtenerse en no más de tres meses. Esta fase dura unos pocos días.

Las estimaciones de esfuerzo asociado a la implementación de las historias la establecen los programadores utilizando como medida el punto. Un punto, equivale a una semana ideal de programación. Las historias generalmente valen de 1 a 3 puntos. Por otra parte,

El equipo de desarrollo mantiene un registro de la “velocidad” de desarrollo, establecida en puntos por iteración, basándose principalmente en la suma de puntos correspondientes a las historias de usuario que fueron terminadas en la última iteración.

La planificación se puede realizar basándose en el tiempo o el alcance. La velocidad del proyecto es utilizada para establecer cuántas historias se pueden implementar antes de una fecha determinada o cuánto tiempo tomará implementar un conjunto de historias. Al planificar por tiempo, se multiplica el número de iteraciones por la velocidad del proyecto, determinándose cuántos puntos se pueden completar. Al planificar según alcance del sistema, se divide la suma de puntos de las historias de usuario seleccionadas entre la velocidad del proyecto, obteniendo el número de iteraciones necesarias para su implementación.

### **3. Iteraciones:**

Esta fase incluye varias iteraciones sobre el sistema antes de ser entregado. El Plan de Entrega está compuesto por iteraciones de no más de tres semanas. En la primera iteración se puede intentar establecer una arquitectura del sistema que pueda ser utilizada durante el resto del proyecto. Esto se logra escogiendo las historias que fueren la creación de esta arquitectura, sin embargo, esto no siempre es posible ya que es el cliente quien decide qué historias se implementarán en cada iteración (para maximizar el valor de negocio). Al final de la última iteración el sistema estará listo para entrar en producción.

Los elementos que deben tomarse en cuenta durante la elaboración del Plan de la Iteración son: historias de usuario no abordadas, velocidad del proyecto, pruebas de aceptación no superadas en la iteración anterior y tareas no terminadas en la iteración anterior. Todo el trabajo de la iteración es expresado en tareas de programación, cada una de ellas es asignada a un programador como responsable, pero llevadas a cabo por parejas de programadores. Wake en proporciona algunas guías útiles para realizar la planificación de la entrega y de cada iteración.

### **4. La Velocidad del Proyecto:**

Medida que representa la rapidez con la que se desarrolla el proyecto; estimarla es muy sencillo, basta con contar el número de historias de usuario que se pueden implementar en una iteración; de esta forma, se sabrá el cupo de historias que se pueden desarrollar en las distintas iteraciones. Usando la velocidad del proyecto controlaremos que todas las tareas se puedan desarrollar en el tiempo del que dispone la iteración. Es conveniente reevaluar esta medida cada 3 o 4 iteraciones y si se aprecia que no es adecuada hay que negociar con el cliente un nuevo "Release Plan" [10].

#### **2.4.2. FASE 2: Diseño.**

##### **1. Diseños Simples**

Se debe diseñar la solución más simple que pueda funcionar y ser implementada en un momento determinado del proyecto. La complejidad innecesaria y el código extra debe ser removido inmediatamente. Kent Beck dice que en cualquier momento el diseño adecuado para el software es aquel que: supera con éxito todas las pruebas,

no tiene lógica duplicada, refleja claramente la intención de implementación de los programadores y tiene el menor número posible de clases y métodos.

## **2. Glosarios de Términos**

Usar glosarios de términos y una correcta especificación de los nombres de métodos y clases ayudará a comprender el diseño y facilitará sus posteriores ampliaciones y la reutilización del código.

## **3. Refactorizar**

Refactorizar es mejorar y modificar la estructura y codificación de códigos ya creados sin alterar su funcionalidad. Refactorizar supone revisar de nuevo estos códigos para procurar optimizar su funcionamiento. Es muy común rehusar códigos ya creados que contienen funcionalidades que no serán usadas y diseños obsoletos.

### **2.4.3. FASE 3: Desarrollo.**

Antes del desarrollo de cada historia de usuario el cliente debe especificar detalladamente lo que ésta hará y también tendrá que estar presente cuando se realicen los test que verifiquen que la historia implementada cumple la funcionalidad especificada. La codificación debe hacerse atendiendo a estándares de codificación ya creados.

### **2.4.4. FASE 4: Pruebas.**

La producción de código está dirigida por las pruebas unitarias. Las pruebas unitarias son establecidas antes de escribir el código y son ejecutadas constantemente ante cada modificación del sistema. Los clientes escriben las pruebas funcionales para cada historia de usuario que deba validarse. En este contexto de desarrollo evolutivo y de énfasis en pruebas constantes, la automatización para apoyar esta actividad es crucial.

## **CAPÍTULO III**

### **1. Herramientas y Lenguajes de desarrollo.**

#### **3.1 JavaScript.**

JavaScript (abreviado comúnmente "JS") es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas aunque existe una forma de JavaScript del lado del servidor (Server-side JavaScript o SSJS). Su uso en aplicaciones externas a la web, por ejemplo en documentos PDF, aplicaciones de escritorio (mayoritariamente widgets) es también significativo. [11]

JavaScript se diseñó con una sintaxis similar a C, aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo Java y JavaScript no están relacionados y tienen semánticas y propósitos diferentes.

Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del Document Object Model (DOM).

Tradicionalmente se venía utilizando en páginas web HTML para realizar operaciones y únicamente en el marco de la aplicación cliente, sin acceso a funciones del servidor. JavaScript se interpreta en el agente de usuario, al mismo tiempo que las sentencias van descargándose junto con el código HTML.

En la figura 4 se muestra el código escrito en JS necesario para cargar las librerías al proyecto de desarrollo como: JQuery,

```
13  //= require jquery
14  //= require jquery_ujs
15  //= require turbolinks
16  //= require bootstrap
17  //= require_tree .
18  //= require jquery_nested_form
19  //= require nested
20  //= require nested2
21  //= require alertify
22  //= require chosen.jquery
23  //= require moment
24  //= require jquery.dataTables
25  //= require bootstrap-datepicker
26  //= require jquery-ui
27  //= require best_in_place
28  //= require timepicker
```

Figura 4 . Código para cargar librerías JS al proyecto

### 3.2 JQuery.

JQuery es un framework JavaScript libre y open source, del lado del cliente; el objetivo de esta librería de java script es simplificar los comandos comunes de JavaScript. En pocas palabras escribir menos para hacer más.

Se podría decir que sirve como base para la programación avanzada de aplicaciones, que aporta una serie de funciones o códigos para realizar tareas habituales.

Nos ofrece una infraestructura con la que tendremos mucha mayor facilidad para la creación de aplicaciones complejas del lado del cliente. Por ejemplo, con JQuery obtendremos ayuda en la creación de interfaces de usuario, efectos dinámicos, aplicaciones que hacen uso de Ajax, etc. Cuando programemos JavaScript con JQuery tendremos a nuestra disposición una interfaz para programación que nos permitirá hacer cosas con el navegador que estemos seguros que funcionarán para todos nuestros visitantes. Simplemente debemos conocer las librerías del framework y programar utilizando las clases, sus propiedades y métodos para la consecución de nuestros objetivos [12].

#### 3.2.1. Ventajas de JQuery

Es importante comentar que JQuery no es el único framework que existe en el mercado. Existen varias soluciones similares que también funcionan muy bien, que básicamente nos sirven para hacer lo mismo. Como es normal, cada uno de los frameworks tiene sus ventajas e inconvenientes, pero JQuery es un producto con una aceptación por parte de los programadores muy buena y un grado de penetración en

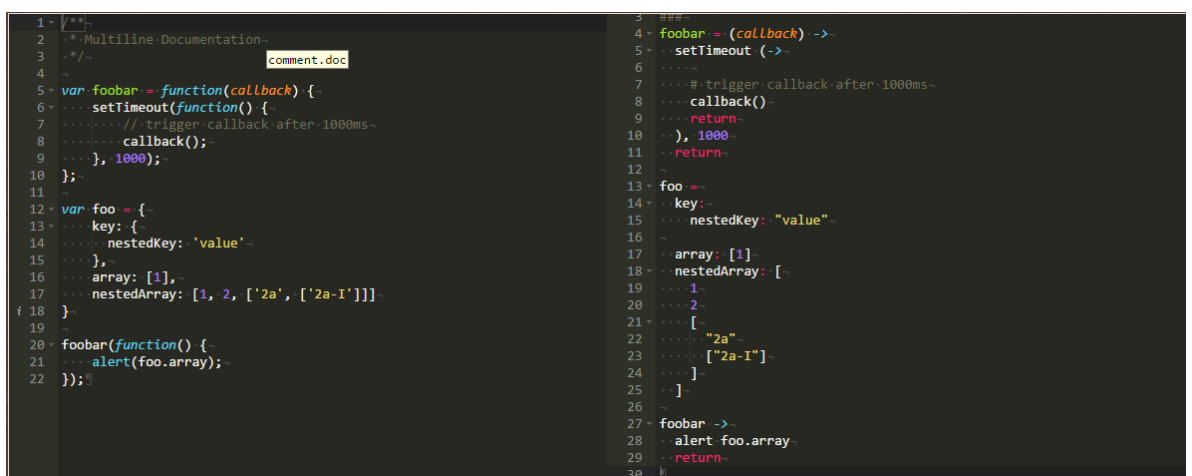
el mercado muy amplio, lo que hace suponer que es una de las mejores opciones. Además, es un producto serio, estable, bien documentado y con un gran equipo de desarrolladores a cargo de la mejora y actualización del framework. Otra cosa muy interesante es la dilatada comunidad de creadores de plugins o componentes, lo que hace fácil encontrar soluciones ya creadas en jQuery para implementar asuntos como interfaces de usuario, galerías, votaciones, efectos diversos, etc. [12]

### 3.3 Coffeescript.

Coffeescript es un poco de lenguaje que compila a JavaScript. La sintaxis es inspirada en Ruby y Python, e implementa muchas características de ambos lenguajes. La sintaxis es a menudo idéntica de JavaScript, pero este no es un súper conjunto, y por lo tanto, algunas palabras clave de JavaScript, como la función y var no están permitidas, los cuales arrojará errores de sintaxis. Si se está escribiendo un archivo Coffeescript, que tiene que ser puro no se puede entremezclar las dos lenguas [13].

Este añade características más sofisticadas, como la comprensión de listas y la coincidencia de patrones. Coffeescript compila previsiblemente a JavaScript y los programas se pueden escribir por lo general en 1/3 menos de líneas de código sin ningún efecto en el rendimiento ni en el tiempo de ejecución. A partir del 16 de marzo de 2011, Coffeescript ha estado en la lista de los proyectos más visitados en GitHub, y al 27 de julio 2012 es el undécimo lenguaje más popular en GitHub.

En la figura se puede apreciar un ejemplo de la sintaxis, en la parte izquierda: Javascript y en la parte derecha el resultado en Coffeescript.



```
1  /**
2  * Multiline Documentation
3  */
4
5  var foobar = function(callback) {
6    setTimeout(function() {
7      // trigger callback after 1000ms
8      callback();
9    }, 1000);
10 };
11
12 var foo = {
13   key: {
14     nestedKey: 'value'
15   },
16   array: [1],
17   nestedArray: [1, 2, ['2a', ['2a-I']]]
18 };
19
20 foobar(function() {
21   alert(foo.array);
22 });
```

```
1  /**
2  * Multiline Documentation ->
3  */
4  foobar = (callback) ->
5    setTimeout (->
6      ->
7      # trigger callback after 1000ms
8      callback()
9    ), 1000
10   return
11
12
13  foo =
14    key:
15      nestedKey: "value"
16
17    array: [1]
18    nestedArray: [
19      1
20      2
21      ["2a"
22        ["2a-I"]
23      ]
24    ]
25
26
27  foobar ->
28    alert foo.array
29    return
30
```

Figura 5. Ejemplo de sintaxis Coffeescript

El lenguaje tiene un seguimiento relativamente grande por parte de la comunidad Ruby. El soporte a Coffeescript está incluido en Ruby on Rails desde su versión 3.1.

En la figura 5 podemos observar el código desarrollado en la sintaxis Coffeescript necesario para activar los tooltip de Bootstrap, el cual hace uso de la librería JQuery. Los resultados se pueden apreciar en la figura 6. Donde se coloca el puntero encima del icono indicando que acción se llevara a cabo "Suspende".

```
1 window.Helpers ||= {}
2
3 window.Helpers.TooltipHelper = {
4   init: ->
5     $(".ttip").tooltip placement: "top"
6     # $(".mostrar").each ->
7     # console.Log $(this)
8     # $(this).insertBefore "<div class='tooltip' title='Mostrar'><div class='tooltip-inner'>"
9     # $(this).insertAfter "</div><div class='tooltip-arrow'></div></div>"
10    # $(this).tooltip()
11  }
12
13 jQuery window.Helpers.TooltipHelper.init
14 $(document).on "page:load", window.Helpers.TooltipHelper.init
```

Figura 6. Código para activación de tooltip Bootstrap.

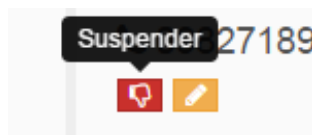


Figura 7. Tooltip Bootstrap.

### 3.4 Hojas de estilo (CSS).

Hojas de Estilo en Cascada (Cascading Style Sheets), es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre estilo y formato de sus documentos



CSS permite separar los contenidos de la página y la información sobre su aspecto. En el ejemplo anterior, dentro de la propia página HTML se crea una zona especial en la que se incluye toda la información relacionada con los estilos de la página.

Utilizando CSS, se pueden establecer los mismos estilos con menos esfuerzo y sin *ensuciar* el código HTML de los contenidos con etiquetas <font>, la etiqueta <style> crea una zona especial donde se incluyen todas las reglas CSS que se aplican en la página [14].

### ¿Para qué sirve?

CSS se utiliza para dar estilo a documentos HTML y XML, separando el contenido de la presentación. Los Estilos definen la forma de mostrar los elementos HTML y XML. CSS permite a los desarrolladores Web controlar el estilo y el formato de múltiples páginas Web al mismo tiempo. Cualquier cambio en el estilo marcado para un elemento en la CSS afectará a todas las páginas vinculadas a esa CSS en las que aparezca ese elemento.

Mediante etiquetas HTML en una página Web se define la estructura del documento y con las definiciones de estilos se describe la apariencia. Las hojas de estilo en cascada, definen el efecto visual que tendrán las distintas etiquetas HTML. CSS3 es el último estándar de CSS [14].

En la figura 8. Se puede apreciar el código necesario para aplicar un fondo blanco con borde de 1px para las tablas de la aplicación.

```
table.dataTable td.sorting_1{ background-color: white; border:1px lightgrey; }
table.dataTable td{ background-color: white; border:1px lightgrey;}
table.dataTable tr.odd { background-color: white; border:1px lightgrey;}
table.dataTable tr.even{ background-color: white; border:1px lightgrey; }
```

Figura 8. Código css aplicado a tablas de la aplicación.

## **3.5 Bootstrap Framework Frontend.**

### **3.5.1. Introducción**

Bootstrap es un framework o conjunto de herramientas de software libre para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como, extensiones de JavaScript opcionales adicionales.

### **3.5.2. Características**

Bootstrap tiene un soporte relativamente incompleto para HTML5 y CSS 3, pero es compatible con la mayoría de los navegadores web. La información básica de compatibilidad de sitios web o aplicaciones está disponible para todos los dispositivos móviles y navegadores. Existe un concepto de compatibilidad parcial que hace disponible la información básica de un sitio web para todos los dispositivos móviles y navegadores. Por ejemplo, las propiedades introducidas en CSS3 para las esquinas redondeadas, gradientes y sombras son usadas por Bootstrap a pesar de la falta de soporte de navegadores antiguos. Esto extiende la funcionalidad de la herramienta, pero no es requerida para su uso [15].

Desde la versión 2.0 también soporta diseños sensibles. Esto significa que el diseño gráfico de la página se ajusta dinámicamente, tomando en cuenta las características del dispositivo usado (Computadoras, tabletas, teléfonos móviles). Bootstrap es de código abierto y está disponible en GitHub.

### **3.5.3. Estructura y función.**

Bootstrap es modular y consiste esencialmente en una serie de hojas de estilo LESS que implementan la variedad de componentes de la herramienta. Una hoja de estilo llamada bootstrap.less incluye los componentes de las hojas de estilo. Los desarrolladores pueden adaptar el mismo archivo de Bootstrap, seleccionando los componentes que deseen usar en su proyecto.

Los ajustes son posibles en una medida limitada a través de una hoja de estilo de configuración central. Los cambios más profundos son posibles mediante las declaraciones LESS [15].

El uso del lenguaje de hojas de estilo LESS permite el uso de variables, funciones y operadores, selectores anidados, así como clases mixin.

Desde la versión 2.0, la configuración de Bootstrap también tiene una opción especial de "Personalizar" en la documentación. Por otra parte, los desarrolladores eligen en un formulario los componentes y ajustes deseados, y de ser necesario, los valores de varias opciones a sus necesidades. El paquete consecuentemente generado ya incluye la hoja de estilo CSS pre-compilada.

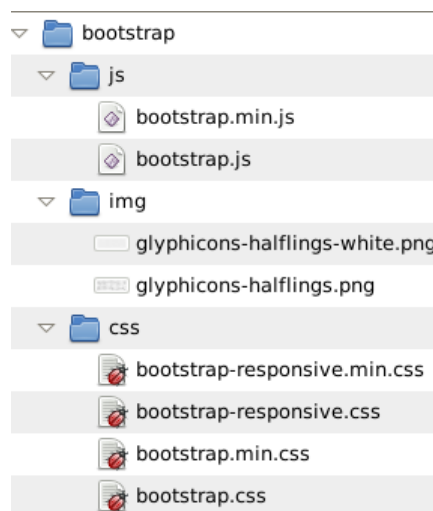


Figura 9. Estructura de archivos de Bootstrap.

### 3.5.3.1. Sistema de cuadrilla y diseño sensible.

Bootstrap viene con una disposición de cuadrilla estándar de 940 píxeles de ancho. Alternativamente, el desarrollador puede usar un diseño de ancho-variable. Para ambos casos, la herramienta tiene cuatro variaciones para hacer uso de distintas resoluciones y tipos de dispositivos: teléfonos móviles, formato de retrato y paisaje, tabletas y computadoras con baja y alta resolución (pantalla ancha). Esto ajusta el ancho de las columnas automáticamente.

### 3.5.3.2. Entendiendo la hoja de estilo CSS

Bootstrap proporciona un conjunto de hojas de estilo que proveen definiciones básicas de estilo para todos los componentes de HTML. Esto otorga una uniformidad al navegador y al sistema de anchura, da una apariencia moderna para el formateo de los elementos de texto, tablas y formularios.

### 3.5.3.3. Componentes re-usables

En adición a los elementos regulares de HTML, Bootstrap contiene otra interfaz de elementos comúnmente usados. Ésta incluye botones con características avanzadas (e.g grupo de botones o botones con opción de menú desplegable, listas de navegación, etiquetas horizontales y verticales, ruta de navegación, paginación, etc.), etiquetas, capacidades avanzadas de miniaturas tipográficas, formatos para mensajes de alerta y barras de progreso

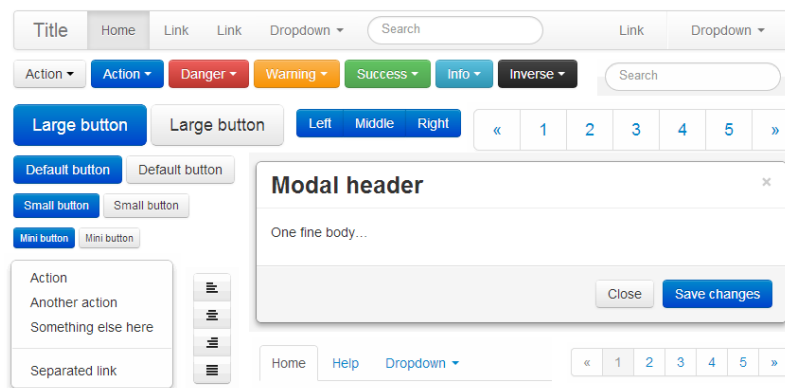


Figura 10. Componentes gráficos de Bootstrap.

## 3.6 Gestor de base de datos Mysql

### 3.6.1 ¿Qué es Mysql?

MySQL es un sistema de gestión de bases de datos relacional, licenciado bajo la GPL de la GNU. Su diseño multi hilo le permite soportar una gran carga de forma muy eficiente. Fue creada por la empresa sueca MySQL AB, que mantiene el copyright del código fuente del servidor SQL, así como también de la marca.

Aunque es software libre, MySQL AB distribuye una versión comercial, que no se diferencia de la versión libre más que en el soporte técnico que se ofrece, y la posibilidad de integrar este gestor en un software propietario, ya que de no ser así, se vulneraría la licencia GPL.

### 3.6.2 Seguridad

Un sistema de privilegios y contraseñas que es muy flexible y seguro, y que permite verificación basada en el host. Las contraseñas son seguras porque todo el tráfico de contraseñas está cifrado cuando se conecta con un servidor.

### 3.6.3 Escalabilidad y límites

Soporte a grandes bases de datos. Usamos MySQL Server con bases de datos que contienen 50 millones de registros. También conocemos a usuarios que usan MySQL Server con 60.000 tablas y cerca de 5.000.000.000.000 de registros.

Se permiten hasta 64 índices por tabla (32 antes de MySQL 4.1.2). Cada índice puede consistir desde 1 hasta 16 columnas o partes de columnas. El máximo ancho de límite son 1000 bytes (500 antes de MySQL 4.1.2). Un índice puede usar prefijos de una columna para los tipos de columna CHAR, VARCHAR, BLOB, o TEXT.

### 3.6.4 Ruby on Rails y Mysql

Ruby on Rails posee un módulo llamado Active records, quien se encarga de gestionar las consultas y creación de tablas y filas de la bd, Rails gestiona los accesos a la base de datos automáticamente (aunque, si se necesita, se pueden hacer consultas directas en SQL) Rails intenta mantener la neutralidad con respecto a la base de datos, la portabilidad de la aplicación a diferentes sistemas de base de datos y la reutilización de bases de datos preexistentes.

```
1 # MySQL. Versions 4.1 and 5.0 are recommended.
2 #
3 # Install the MySQL driver:
4 #   gem install mysql2
5 #
6 # And be sure to use new-style password hashing:
7 #   http://dev.mysql.com/doc/refman/5.0/en/old-client.html
8 development:
9   adapter: mysql2
10  encoding: utf8
11  reconnect: false
12  database: sisfa_development
13  pool: 5
14  username: root
15  password: root
16  socket: /tmp/mysql.sock
17
18 # Warning: The database defined as "test" will be erased and
19 # re-generated from your development database when you run "rake".
20 # Do not set this db to the same as development or production.
21 test:
22   adapter: mysql2
23   encoding: utf8
24   reconnect: false
25   database: sisfa_test
26   pool: 5
27   username: root
28   password: root
29   socket: /var/run/mysqld/mysqld.sock
30
31 production:
32   adapter: mysql2
33   encoding: utf8
34   reconnect: false
35   database: sisfa_production
36   pool: 5
37   username: root
38   password: root
39   socket: /tmp/mysql.sock
```

Figura 11. Conexión a la base de datos desde RoR.

## **e. Materiales y Métodos**

### **1. MÉTODOS**

El desarrollo de este software requiere seguir los lineamientos de ciertos métodos, así como de técnicas e instrumentos que permitan la recopilación y análisis de la información necesaria para la presentación del PFC, tales como:

**1. Método Deductivo:** Para el desarrollo de este software este método permitió recolectar la información sobre conceptos acorde al tema, principios, normas y más temas relacionados al manejo y organización de la información de los procesos que se efectúan en el Hospital Básico 7 B.I Loja, para con ello determinar los problemas que se presentan tanto en el desenvolvimiento correcto del servicio de fisioterapia y en la recepción de los datos requeridos en los departamentos de estadística y doctores con los que se relaciona. Los pasos secuenciales e interactuantes que se siguieron en el desarrollo del Método Deductivo son:

1. Aplicar la Metodología de desarrollo XP.
2. Utilizar Herramientas y lenguajes de desarrollo: Ruby, y framework Ruby on Rails y el gestor de base de datos Mysql.
3. Desarrollo y ejecución del sistema de acuerdo a los requerimientos del usuario

**2. Método Inductivo:** Permitted analizar las particularidades de la realidad actual, ayuda a profundizar la indagación de funcionalidades y ventajas del sistema con lo cual se llega al problema general de investigación, por medio de la extracción de información que facilitaron las personas encargadas de los servicios antes mencionados, para llegar a la conclusión del problema global de esta entidad. Este método funciona desde el punto de vista modificador o auxiliador al Método Deductivo ya que todo proyecto de esta clase es vivencial e interactuante en el medio social – salud – técnico – económico, etc.

**3. Método del Enfoque de Sistemas:** Este método a determinado organizar en forma sintética el Método Deductivo y Método Inductivo a través del desarrollo del módulo de Fisioterapia el cual siguió las etapas de; Estudio Preliminar, Análisis, Diseño e Implementación y que para su explotación racional se lo realizara a través de los manuales: del programador y de usuario durante los 3 años de explotación del sistema dentro de la oficina de Fisioterapia. Con el desarrollo y

ejecución del sistema de acuerdo a los requerimientos de usuario el hospital podrá brindar un mejor servicio al usuario y por ende retroalimentar a la institución y a la vez desarrollarse en una forma armónica e interactuando con la sociedad.

## 2. Técnicas

Las Técnicas que se utilizaron para la recopilación de la información son las siguientes:

- **Lectura comprensiva:** El uso de este tipo de lectura nos permitió obtener un conocimiento ordenado y sistemático de la realidad o de los acontecimientos hecho o ideas relacionadas con el tema específico.
- **La Observación directa:** Esta técnica permitió apreciar los problemas encontrados respecto a la falta de comunicación entre departamentos y consultorios médicos.
- **Entrevista:** Esta técnica se utilizó para determinar problemas específicos ya que se la realizo con todas las personas encargadas de los servicios de Fisiatría, doctores de la institución y el departamento de estadística; que colaboraron con las respectivas preguntas planteadas por el investigador para así llegar a formular y determinar el problema general del Proyecto de fin de carrera (PFC).
- **Encuesta:** Esta técnica se realizó al personal que labora en los respectivos departamentos luego de su interacción con el módulo de fisiatría del sistema "SISFA" ya implementado, permite determinar si el sistema cumple con lo solicitado por la institución, es decir permite realizar la validación de la aplicación, lo cual se puede corroborar en la parte de resultados fase 4 y en el anexo 1, 2 y 3 donde se encuentran adjuntadas las encuestas.

## 3. METODOLOGÍA DE DESARROLLO DE SOFTWARE

Para el desarrollo del proyecto, se empleó la metodología XP (Programación Extrema), la cual tiene muchas ventajas como calidad, flexibilidad y un diseño simple. Además es una metodología ligera que trata de buscar métodos sencillos de obtener software de calidad.

Para suplir la falta de requisitos, casos de usos y demás herramientas utilizamos historias de usuarios tal como exige XP, además para el desarrollo del proyecto se debe considerar la implementación de módulos los mismos que deben contemplar las cuatro fases de la metodología XP (Planificación, Diseño, Desarrollo y Pruebas).

Además se cumple con las recomendaciones para emplear XP en este proyecto como son:

- ✓ Interés sincero por la parte involucrada.
- ✓ El equipo de trabajo pequeño, el cual consta de 1 persona.
- ✓ Se dispone de una formación elevada y capacidad de aprender.

### **3.1 Fases de la Metodología XP**

#### **3.1.1 Fase 1: Planificación del proyecto.**

**Historias De Usuario:** El primer paso es definir las historias de usuario con el cliente. Las cuales son usadas para estimar tiempos de desarrollo de la parte de la aplicación que describen. La descripción de estas se encuentra en el apartado 1.3.4. de la fase 1 de Resultados, son en total 22 historias de usuario, además existe un resumen de las mismas en el apartado 1.3.5.

**Plan de Entregas:** Después de tener ya definidas las historias de usuario es necesario crear un plan de publicaciones, en inglés "Release planning" donde los desarrolladores y clientes establecen los tiempos de implementación ideales de las historias de usuario, la prioridad con la que serán implementadas y las historias que serán implementadas en cada versión del programa, el mismo se lo puede evidenciar en el apartado 1.3.6. Fase 1 de Resultados.

**Iteraciones:** Al comienzo de cada iteración los clientes deben seleccionar las historias de usuario definidas en el "Release planning" que serán implementadas por iteraciones. En el apartado 1.3.7. Fase 1 de Resultados se encuentra la división de iteraciones planteada.



**Velocidad del proyecto:** Representa la rapidez con la que se desarrolla el proyecto; basta con contar el número de historias de usuario que se pueden implementar en una iteración, ver apartado 1.3.8. Fase 1 Resultados.

### **3.1.2 Fase 2: Diseño**

**Diseños simples:** Realizar los diseños de los módulos fácilmente entendibles, ver apartado 2.1 de la fase 2 de Resultados.

Glosarios de términos: Usar glosarios de términos y una correcta especificación de los nombres de métodos y clases ayudará a comprender el diseño y facilitará sus posteriores ampliaciones y la reutilización del código.

**Refactorizar:** Supone revisar de nuevo estos códigos para procurar optimizar su funcionamiento.

**Tarjetas C.R.C.:** El uso de las tarjetas C.R.C (Class, Responsibilities and Collaboration) permiten centrarse y apreciar el desarrollo orientado a objetos. Las mismas que se pueden evidenciar en el apartado 2.3 de la fase 2 de Resultados.

### **3.1.3 Fase 3: Codificación**

Programar bajo estándares mantiene el código consistente y facilita su comprensión y escalabilidad. En esta fase también encontramos las pruebas unitarias, las cuales están descritas en el apartado 3.2 de la fase 3 de Resultados.

### **3.1.4 Fase 4: Pruebas**

Se hace uso de test para comprobar el funcionamiento de los códigos que vayamos implementando.

En la fase 4 de Resultados se encuentra detallado todos los apartados aplicados. Se realizó una serie de pruebas como: pruebas de aceptación (4.2 de fase 4 de Resultados), pruebas funcionales (4.3.12 de fase 4 de Resultados), pruebas de Usabilidad (4.3.13 de fase 4 de Resultados), pruebas de seguridad (4.3.14 de fase 4 de Resultados) y pruebas de carga y estrés (4.3.15 de fase 4 de Resultados).

## **f. Resultados**

La presentación de resultados se encuentra dividida de acuerdo a las fases de la metodología XP.

### **FASE 1**

#### **1. Planificación del Proyecto**

En esta primera fase se realizó un análisis de requisitos, con la finalidad de descubrir, modelar y delimitar los requisitos que han de satisfacer al sistema; es decir identificar de forma clara las funcionalidades que debe llevar a cabo el software.

##### **1.1 Introducción**

Esta sección recoge la Especificación de Requisitos Software (ERS), de la aplicación desarrollada, siguiendo el estándar IEEE 830-1998.

###### **1.1.1 Propósito**

Una de las fases más relevantes en el desarrollo de software es el levantamiento de requerimientos; a través de él, se permite entender mejor el problema, organizando y estructurando la información que el cliente haya brindado, se puede identificar las necesidades del negocio, los objetivos de la aplicación y los usuarios que tendrán interacción con ella.

Esta fase pretende orientar el trabajo inicial del desarrollo de software, dando descripciones preliminares de la aplicación, considerando algunos detalles de las secciones claves que la identifican.

Está dirigido a todas aquellas personas interesadas en el proyecto, principalmente al Hospital Básico 7 BI Loja y a las personas que hacen uso de sus servicios.

###### **1.1.2 Alcance**

El producto de este proyecto a desarrollar se implementará dentro del software ya vigente en la institución llamado "SISFA" relevante al departamento de Fisiatría, el cual se caracteriza por ser una aplicación web y multiplataforma. A continuación se presenta las especificaciones que el sistema debe cumplir.

Dentro de lo correspondiente a administraciones:

- **Administrar Terapias y tratamientos:** Ingresar, modificar, suspender, activar terapias y tratamientos. Los datos necesarios son: código único y nombre de la terapia.
- **Administrar Fisiatra:** Ingresar, modificar, suspender, eliminar. Los datos necesarios son: nombres, número de identificación, dirección, teléfono, correo electrónico, especialidad, cantidad de turnos.
- **Administrar Turnos:** Crear turno día, Ingresar turno mañana, Impresión de listado de turnos. Los datos necesarios son: paciente, doctor, fecha, hora.
- **Administrar Cuenta:** Crear, cambiar de clave, cambiar información de su cuenta. Los datos necesarios son: nombre de usuario, clave.
- **Administrar Usuarios:** Crear, modificar clave, suspender cuenta, activar cuenta. Los datos necesarios son: nombre de usuario, clave, nombres, número de identificación.
- **Administrar Horarios:** Crear, suspender, activar y eliminar. Los datos necesarios son: hora inicial y hora final.

El sistema también:

- Registra diagnósticos de pacientes.
- Permite la descarga de reportes en formato excel.
- Informar del número de pacientes atendidos por turnos.
- Informa a los fisiatras de sus pacientes que han solicitado un turno.
- Informa a estadística si un paciente ha sido registrado en fisioterapia.

El objetivo es “Desarrollar un módulo de gestión de los servicios de Fisiatría para el Sistema Web SISFA del Hospital Básico 7 B.I Loja”, con el fin que la institución cuente con un software que permite la integración de sus servicios, brindando al personal una herramienta fácil de usar, buscando como metas:

- Flujo de información correcto y ordenado de procesos internos de la institución.
- Facilitar el acceso a información compartida entre departamentos.
- Mejorar el desenvolvimiento laboral del personal.
- Brindar un mejor servicio a la ciudadanía tanto en atención como en seguridad de registros médicos.

### 1.1.3 Definiciones, siglas y abreviaciones.

1. **Usuario:** Persona(s) que hace uso del software, en la aplicación se encuentran diferentes tipos de usuario como: Administrador, Administrador Fisiatría, Fisiatra.
  - **Administrador:** Usuario único que puede crear y asignar cuentas.
  - **Administrador Fisiatría:** Usuario que se encuentra a cargo de todo el departamento de fisiatría.
  - **Fisiatra:** Usuario que se encuentra a cargo de diagnosticar a un paciente.
2. **Paciente:** Persona que debe haber sido previamente registrada en la institución y posee una historia clínica, la cual puede ser de tres tipos: Militar, Familiar-Militar o Civil:
  - **Militar:** Pertenece al Ejército de las Fuerzas Armadas del Ecuador y está afiliado(a) al ISSFA.
  - **Familiar-Militar:** Una persona dentro de su núcleo familiar pertenece al Ejército de las Fuerzas Armadas del Ecuador y está afiliado(a) al ISSFA.
  - **Civil:** Persona particular que hace uso de los servicios de la institución.
3. **Terapias:** Hace referencia a los diferentes tratamientos a los que se puede someter un paciente dependiendo de su lesión.
4. **Diagnóstico:** análisis de enfermedad, al cual se llega a partir de hallazgos, síntomas, signos de un paciente.
5. **Turno:** Apertura de tiempo para una consulta médica, por parte de un paciente a un doctor.

## **1.2. Descripción global**

### **1.2.1. Perspectiva del producto**

Este proyecto de ingeniería de software está basado en lenguajes web, por lo cual la arquitectura es multiplataforma, siendo independiente del sistema operativo, por tal motivo no es necesario establecer relación con otros sistemas.

El diseño de la interfaz “UI” se ha realizado de manera clara y sencilla, la cual es intuitiva permitiendo que no sea confusa en la parte de Experiencia Usuario “UX”.

### **1.2.2. Funciones del producto**

A continuación se enmarca las funciones que se cumplen dentro del módulo de fisioterapia:

- Administrar la información de los terapeutas.
- Suspensión de doctores.
- Administración de terapias y tratamientos.
- Impresión de historial clínico de registros médicos.
- Visualización y emisión de turnos en el día o para al día siguiente.
- Generación y registro de sesiones de terapias.
- Generación automática de reportes diarios o mensuales por terapeuta.
- Generación automática de reportes diaria o mensuales por paciente.
- Generación automática de reportes diaria o mensuales por financiero.
- Generación automática de certificado médico.

Para describir de forma más precisa las funcionalidades identificadas para el sistema se ha organizado de la siguiente manera, la tabla II describe los perfiles de usuarios de la aplicación.

TABLA II: PERFILES DE USUARIOS

Actores del Sistema	Funciones
Administrador	<ul style="list-style-type: none"> <li>• Crear usuario administrador de fisiatría</li> <li>• Suspender usuarios</li> </ul>
Administrador Fisiatría	<ul style="list-style-type: none"> <li>• Registrar terapistas</li> <li>• Suspender terapistas</li> <li>• Administrar terapias y tratamientos</li> <li>• Generación e impresión de reportes médicos</li> <li>• Generación e impresión de certificados médicos</li> <li>• Crear horarios</li> <li>• Suspender horarios</li> <li>• Emisión de turnos</li> </ul>
Fisiatra	<ul style="list-style-type: none"> <li>• Registrar diagnósticos de paciente</li> </ul>
Administrador Estadística	<ul style="list-style-type: none"> <li>• Emisión de turnos</li> <li>• Emisión de reportes fisiatría</li> </ul>
Auxiliar Estadística	<ul style="list-style-type: none"> <li>• Emisión de turnos</li> </ul>
Doctor	<ul style="list-style-type: none"> <li>• Remitir pacientes a fisiatría</li> </ul>

### 1.2.3. Características del Usuario

Los usuarios del sistema son los funcionarios que trabajan en los diferentes departamentos del hospital, los cuales están categorizados como:

- Administrador: jefe del departamento de sistemas del hospital.
- Administrador fisiatría: responsable del departamento de fisiatría.
- Administrador estadística: responsable del departamento de estadística.
- Auxiliar de estadística: ayudante de estadística.
- Fisiatra: doctor especializado en terapias y tratamientos.
- Doctor: doctores que laboran en la institución.

Dependiendo de su perfil, el usuario podrá ejercer las funciones descritas en el apartado 1.2.2., Tabla II de perfiles de usuario.

#### **1.2.4. Restricciones**

Debido a que se ha hecho uso de nuevas tecnologías como CSS3, el navegador debe tener habilitado el uso de JS, pero el mismo sufrirá ciertos cambios en la parte visual, al abrirlo en versiones inferiores a IE8, pero no existe ningún problema en cuanto a las operaciones.

#### **1.2.5. Atención y dependencias**

Algunos factores que pueden afectar los requerimientos del sistema son:

- Agregar nuevos roles de usuarios los cuales ya están definidos.
- Cambiar el diseño de los reportes impresos.
- Modificar el curso de los eventos en las acciones de los usuarios con el sistema.

#### **1.2.6. Prorratear los requisitos**

Algunas mejoras que pueden plantearse para un futuro son:

- Agregar nuevos formularios médicos.
- Permitir a la aplicación conectarse con el sistema del ejercito ISSFA, para validar y autocompletar la información de pacientes militares.
- Emisión de turnos por internet.

### **1.3. Los requisitos específicos**

#### **1.3.1. Requerimientos Funcionales**

El sistema permitirá a todos los usuarios:

TABLA III: PERMISOS A TODOS LOS USUARIOS

<b>REF.</b>	<b>DESCRIPCIÓN</b>	<b>CATEGORÍA</b>
RF-1.01	Acceder al sistema mediante el inicio de sesión con la clave correspondiente.	Oculto
RF-1.02	Cambiar su clave.	Evidente

El sistema permitirá al administrador:

TABLA IV: PERMISOS DEL ADMINISTRADOR

<b>REF.</b>	<b>DESCRIPCIÓN</b>	<b>CATEGORÍA</b>
RF-1.03	Anular cuentas de usuarios	Evidente
RF-1.04	Acceso a todas las funciones de los departamentos	Ocultas

El sistema permitirá al administrador fisiatría:

TABLA V: PERMISOS HA ADMINISTRADOR DE FISIATRÍA

<b>REF.</b>	<b>DESCRIPCIÓN</b>	<b>CATEGORÍA</b>
RF-1.05	Administrar terapias (crear, editar, suspender)	Evidente
RF-1.06	Administra horarios (crear, suspender)	Evidente
RF-1.07	Administrar fisiatras (crear, editar, suspender)	Evidente
RF-1.08	Emitir turnos (diarios y para mañana )	Evidente
RF-1.09	Generar reportes por terapeuta	Evidente
RF-1.10	Generar reportes por paciente/certificados	Evidente
RF-1.11	Generar reportes por factura	Evidente
RF-1.12	Editar registros	Evidente
RF-1.13	Fijar turnos por hora	Evidente



El sistema permitirá al administrador estadística:

TABLA VII: PERMISOS HA ADMINISTRADOR DE ESTADÍSTICA

<b>REF.</b>	<b>DESCRIPCIÓN</b>	<b>CATEGORÍA</b>
RF-1.14	Administrar historias clinicas (crear, editar, eliminar)	Evidente
RF-1.15	Administrar doctores (crear, editar, suspender, eliminar)	Evidente
RF-1.16	Emitir turnos (diarios y para mañana )	Evidente
RF-1.17	Generar reportes mensuales de fisioterapia	Evidente

El sistema permitirá al auxiliar de estadística:

TABLA VIII: PERMISOS DEL AUXILIAR DE ESTADÍSTICA.

<b>REF.</b>	<b>DESCRIPCIÓN</b>	<b>CATEGORÍA</b>
RF-1.18	Emitir turnos (diarios y para mañana )	Evidente

El sistema permitirá al fisiatra:

TABLA VIII: PERMISOS DEL FISIATRA.

<b>REF.</b>	<b>DESCRIPCIÓN</b>	<b>CATEGORÍA</b>
RF-1.19	Diagnosticar pacientes	Evidente
RF-1.20	Diagnosticar pacientes hospitalizados	Evidente
RF-1.21	Visualizar historia clinica de pacientes	Evidente

El sistema permitirá al doctor:

TABLA IXX: PERMISOS DEL DOCTOR.

REF.	DESCRIPCIÓN	CATEGORÍA
RF-1.22	Diagnosticar pacientes por CTEXM	Evidente
RF-1.23	Diagnosticar pacientes por CTEXP	Evidente
RF-1.24	Diagnosticar pacientes hospitalizados	Evidente
RF-1.25	Diagnosticar pacientes por emergencia	Evidente
RF-1.26	Visualizar historia clinica de pacientes	Evidente
RF-1.27	Editar información de pacientes	Evidente
RF-1.28	Remitir pacientes a fisioterapia	Evidente

### 1.3.2. Requerimientos No Funcionales

TABLA X: REQUERIMIENTOS NO FUNCIONALES

REF.	DESCRIPCIÓN
RNF-01	El sistema deberá validar número de identificación.
RNF-02	El sistema deberá mostrar notificaciones ante errores.
RNF-03	El sistema deberá pedir confirmación antes de eliminar algún registro.
RNF-04	El sistema deberá ser intuitivo.
RNF-05	El sistema deberá brindar tiempos de repuestas accesibles en las operaciones.
RNF-06	El sistema deberá mostrar etiquetas claras para campos, botones, menús.

### 1.3.3. Equipo: Integrantes y Roles

TABLA XI: EQUIPO, INTEGRANTES Y ROLES DEL PROYECTO

<b>Miembro</b>	<b>Roles XP</b>	<b>Metodología</b>
Ing. Marco Ocampo	Tutor encargado de seguimiento	<b>XP</b>
Leonardo Alvarado	Programador	

### 1.3.4. Historias de Usuarios

TABLA XII: HISTORIA DE USUARIO: ADMINISTRACIÓN USUARIOS

Historia de Usuario	
<b>Número:</b> 1	<b>Nombre:</b> Administración usuarios
<b>Usuario:</b> administrador	
<b>Iteración asignada:</b> 1	
<b>Prioridad en Negocio:</b> Media (Alta / Media / Baja)	<b>Puntos Estimados:</b> 1
<b>Riesgo en Desarrollo:</b> Media (Alto / Medio / Bajo)	<b>Puntos Estimados:</b> 1.5
<p><b>Descripción:</b></p> <p>El administrador puede realizar ingresos de nuevos usuarios, modificaciones, anular o activar. Los tipos de usuario son: Administrador, Administrador fisiatría, Administrador estadística, Auxiliar_estadística, Fisiatra, Doctor.</p>	
<p><b>Observaciones:</b></p> <div style="display: flex; justify-content: space-around; align-items: flex-start;"> <div style="border: 1px solid #ccc; padding: 10px; width: 45%;"> <p style="text-align: center;"><b>Crear Usuarios</b></p> <p>Nombres Apellidos <input type="text"/></p> <p>C.I O RUC <input type="text"/></p> <p>Nombre de Usuario <input type="text"/></p> <p>Auxiliar_Estadística <input type="text"/></p> <p>Clave <input type="text"/></p> <p>Confirmar Clave <input type="text"/></p> <p style="text-align: center; background-color: #28a745; color: white; padding: 5px;">Guardar</p> </div> <div style="width: 45%;"> <div style="border: 1px solid #ccc; padding: 10px; margin-bottom: 10px;"> <div style="display: flex; align-items: center;"> <div> <p><b>Amanda Diaz Orezco</b></p> <p>✉ amandal7@hotmail.com</p> <p>📍 La Argelia 34-98</p> <p>☎ 095643890</p> <p>🏠  </p> <p>— Enfermera</p> </div> </div> </div> <div style="border: 1px solid #ccc; padding: 10px;"> <div style="display: flex; align-items: center;"> <div> <p><b>Gonzalez Cano Edwin Alberto</b></p> <p>✉ </p> <p>📍 La Tebaida</p> <p>☎ 0956432589</p> <p>🏠  </p> <p>— Doctor</p> </div> </div> </div> </div> </div>	

TABLA XIII: HISTORIA DE USUARIO REGISTRO TURNO DE TERAPIA.

<b>Historia de Usuario</b>	
<b>Número:</b> 2	<b>Nombre:</b> Registro turno de terapia
<b>Usuario:</b> Administrador fisioterapia, fisiatra	
<b>Iteración asignada:</b> 1	
<b>Prioridad en Negocio:</b> Media (Alta / Media / Baja)	<b>Puntos Estimados:</b> 1
<b>Riesgo en Desarrollo:</b> Alta (Alto / Medio / Bajo)	<b>Puntos Estimados:</b> 2
<b>Descripción:</b> El administrador de fisioterapia y el fisiatra pueden registrar un turno, con terapias y horarios a cada paciente.	
<b>Observaciones:</b>	
	

TABLA XIV: HISTORIA DE USUARIO: CALENDARIO DE TURNOS

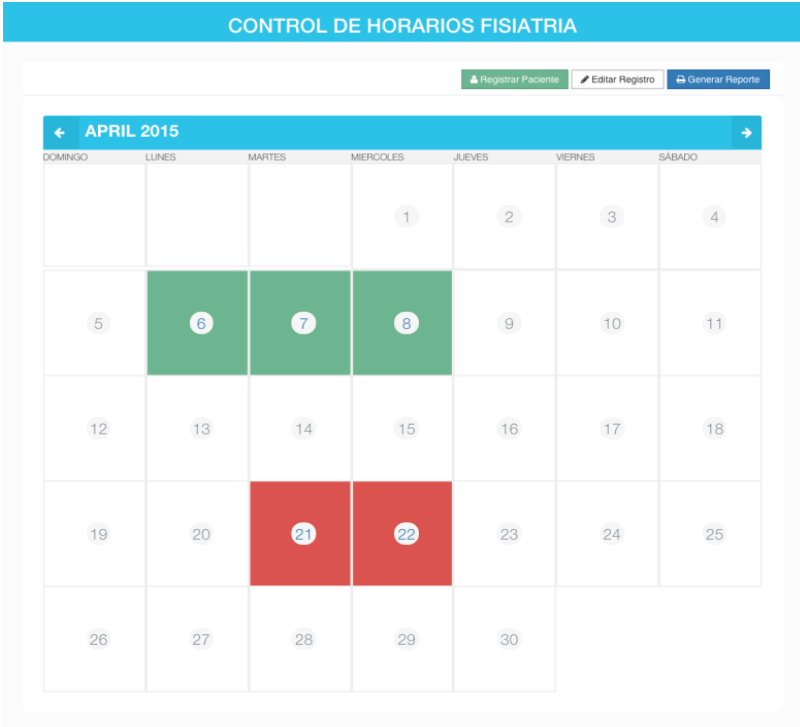
<b>Historia de Usuario</b>	
<b>Número:</b> 3	<b>Nombre:</b> Calendario de turnos
<b>Usuario:</b> Administrador fisiatría y fisiatra	
<b>Iteración asignada:</b> 1	
<b>Prioridad en Negocio:</b> Media (Alta / Media / Baja)	<b>Puntos Estimados:</b> 0.5
<b>Riesgo en Desarrollo:</b> Alto (Alto / Medio / Bajo)	<b>Puntos Estimados:</b> 1.5
<p><b>Descripción:</b></p> <p>El administrador de fisiatría y fisiatra pueden visualizar los días que estén disponibles los turnos basada en colores: rojo ocupado, verde disponible.</p>	
<p><b>Observaciones:</b></p> 	

TABLA XV: HISTORIA DE USUARIO: ADMINISTRACIÓN DE TERAPISTAS

Historia de Usuario	
<b>Número:</b> 4	<b>Nombre:</b> Administrador de terapistas
<b>Usuario:</b> Administrador fisiatría	
<b>Iteración asignada:</b> 1	
<b>Prioridad en Negocio:</b> Media (Alta / Media / Baja)	<b>Puntos Estimados:</b> 0.5
<b>Riesgo en Desarrollo:</b> Alto (Alto / Medio / Bajo)	<b>Puntos Estimados:</b> 1
<p><b>Descripción:</b></p> <p>El administrador de fisiatría puede ingresar, editar, suspender o activar nuevos fisiatras.</p>	
<p><b>Observaciones:</b></p> <div style="text-align: center;">  <p>The image shows two screenshots from a web application. The top screenshot is a form titled 'Crear Terapista' with fields for 'Ingrese Nombres y Apellidos', 'Ingrese Cedula', 'Numero Telefonico', and 'Direccion', followed by a green 'Guardar' button. The bottom screenshot shows the 'Administración de Terapistas' interface with a search bar and a list of four therapist profiles, each with a name, email, location, phone number, and status icons.</p> </div>	

TABLA XVI: HISTORIA DE USUARIO: REPORTES DE FISIATRÍA

<b>Historia de Usuario</b>																					
<b>Número:</b> 5	<b>Nombre:</b> Reportes de fisiatría																				
<b>Usuario:</b> Administrador fisiatría y fisiatra																					
<b>Iteración asignada:</b> 1																					
Prioridad en Negocio: Media (Alta / Media / Baja)	Puntos Estimados: 1																				
Riesgo en Desarrollo: Media (Alto / Medio / Bajo)	Puntos Estimados: 1.5																				
<p><b>Descripción:</b></p> <p>El administrador de fisiatría puede generar diferentes reportes: certificado de terapias, pacientes atendidos por fisiatra, terapias recibidas por paciente y reportes financieros.</p>																					
<p><b>Observaciones:</b></p> <div data-bbox="331 1272 1401 1800" style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p>Reporte por Terapeuta</p> <p>Seleccione Personal: <input type="text" value="Calvache Carrion Leonardo Favio"/>      Fecha Inicial: <input type="text" value="2015-02-24"/>      Fecha Final: <input type="text" value="2015-05-07"/>      <input type="button" value="Buscar"/></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Hora</th> <th>Fecha</th> <th>Paciente</th> <th>Resultado</th> </tr> </thead> <tbody> <tr> <td>7:00 - 8:30</td> <td>2015-03-24</td> <td>Arteaga Zambrano Lenin Francisco</td> <td>primera terapia aplicada</td> </tr> <tr> <td>7:00 - 8:30</td> <td>2015-03-25</td> <td>Arteaga Zambrano Lenin Francisco</td> <td>segunda terapia aplicada</td> </tr> <tr> <td>7:00 - 8:30</td> <td>2015-03-26</td> <td>Arteaga Zambrano Lenin Francisco</td> <td>tercera terapia aplicada</td> </tr> <tr> <td>7:00 - 8:30</td> <td>2015-04-06</td> <td>Adelaida Guaman Urgiles</td> <td>la paciente se muestra favorable al tratamiento recibido</td> </tr> </tbody> </table> <p><input type="button" value="Imprimir PDF"/></p> </div>		Hora	Fecha	Paciente	Resultado	7:00 - 8:30	2015-03-24	Arteaga Zambrano Lenin Francisco	primera terapia aplicada	7:00 - 8:30	2015-03-25	Arteaga Zambrano Lenin Francisco	segunda terapia aplicada	7:00 - 8:30	2015-03-26	Arteaga Zambrano Lenin Francisco	tercera terapia aplicada	7:00 - 8:30	2015-04-06	Adelaida Guaman Urgiles	la paciente se muestra favorable al tratamiento recibido
Hora	Fecha	Paciente	Resultado																		
7:00 - 8:30	2015-03-24	Arteaga Zambrano Lenin Francisco	primera terapia aplicada																		
7:00 - 8:30	2015-03-25	Arteaga Zambrano Lenin Francisco	segunda terapia aplicada																		
7:00 - 8:30	2015-03-26	Arteaga Zambrano Lenin Francisco	tercera terapia aplicada																		
7:00 - 8:30	2015-04-06	Adelaida Guaman Urgiles	la paciente se muestra favorable al tratamiento recibido																		




## Reporte de Pacientes / Certificado

Paciente H.C:

 Arteaga Zambrano Lenin Francisco

Fecha Inicial

 2015-02-23

Fecha Final

 2015-05-01

Buscar

Hora	Fecha	Terapista	Resultado
7:00 - 8:30	2015-03-24	Calvache Carrion Leonardo Favio	primera terapia aplicada
7:00 - 8:30	2015-03-25	Calvache Carrion Leonardo Favio	segunda terapia aplicada
7:00 - 8:30	2015-03-26	Calvache Carrion Leonardo Favio	tercera terapia aplicada

Reporte PDF

Certificado PDF

## Reporte de Factura


Tipo de busqueda

Fecha

Fecha Inicial

 2015-03-01

Fecha Final

 2015-04-30

Buscar

# Factura	Fecha	Total
23	2015-03-20	\$ 10.0
233	2015-03-20	\$ 123.0
223	2015-04-01	\$ 105.0
100	2015-04-15	\$ 22.0
	TOTAL	\$ 260.0

Imprimir PDF

TABLA XVII: HISTORIA DE USUARIO ADMINISTRACIÓN DE TRATAMIENTOS Y TERAPIAS.

<b>Historia de Usuario</b>					
<b>Número:</b> 6	<b>Nombre:</b> Administración de tratamientos y terapias				
<b>Usuario:</b> Administrador fisiatría					
<b>Iteración asignada:</b> 1					
<b>Prioridad en Negocio:</b> Media (Alta / Media / Baja)	<b>Puntos Estimados:</b> 1.5				
<b>Riesgo en Desarrollo:</b> Alta (Alto / Medio / Bajo)	<b>Puntos Estimados:</b> 2				
<p><b>Descripción:</b></p> <p>El Administrador de fisiatría puede: crear, editar tratamientos y terapias, así como también anular y suspender cada tratamiento.</p>					
<p><b>Observaciones:</b></p> <div style="border: 1px solid #ccc; padding: 10px; margin-bottom: 10px;"> <p>Lista de Tratamientos <span style="float: right;">^</span></p> <div style="border: 1px solid #ccc; padding: 5px; margin: 5px 0;"> <p style="text-align: center;">Registro de Tratamientos</p> <p><b>Ingrese Terapia</b></p> <p>Ejm: ELECTROTERAPIA <input style="width: 100%;" type="text"/></p> <p><b>Tratamientos</b></p> <table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; border-bottom: 1px solid #ccc;"># Codigo</th> <th style="text-align: left; border-bottom: 1px solid #ccc;">Nombre de Tratamiento</th> </tr> </thead> <tbody> <tr> <td style="border-bottom: 1px solid #ccc;">Ejm: 291327</td> <td style="border-bottom: 1px solid #ccc;">Ejm: LASER TERAPIA <span style="float: right; color: red; font-size: 0.8em;">x</span></td> </tr> </tbody> </table> <p style="text-align: center; margin-top: 5px;"><span style="border: 1px solid #ccc; padding: 2px 5px; background-color: #f0f0f0;">+ Tratamiento</span></p> <p style="text-align: center; margin-top: 5px;"><span style="background-color: #28a745; color: white; padding: 5px 10px; border: 1px solid #28a745;">GUARDAR</span></p> </div> </div> <div style="margin-top: 10px;"> <p><b>RADIOTERAPIA</b> <span style="float: right; color: #007bff; font-weight: bold;">+</span></p> <div style="display: flex; align-items: center; margin-bottom: 5px;"> <input style="width: 60px; border: 1px solid #ccc; margin-right: 10px;" type="text" value="Codigo"/> <input style="width: 150px; border: 1px solid #ccc; margin-right: 10px;" type="text" value="Ingrese nombre"/> <span style="background-color: #007bff; color: white; padding: 5px 10px; border: 1px solid #007bff; margin-right: 10px;">+</span> </div> <p><b>29342.</b> prueba desde lista tratamiento <span style="float: right; color: red; font-size: 0.8em;">x</span></p> <p><b>23122.</b> prueba 23 <span style="float: right; color: red; font-size: 0.8em;">x</span></p> </div>		# Codigo	Nombre de Tratamiento	Ejm: 291327	Ejm: LASER TERAPIA <span style="float: right; color: red; font-size: 0.8em;">x</span>
# Codigo	Nombre de Tratamiento				
Ejm: 291327	Ejm: LASER TERAPIA <span style="float: right; color: red; font-size: 0.8em;">x</span>				

TABLA XVIII: HISTORIA DE USUARIO: ADMINISTRACIÓN DE HORARIOS


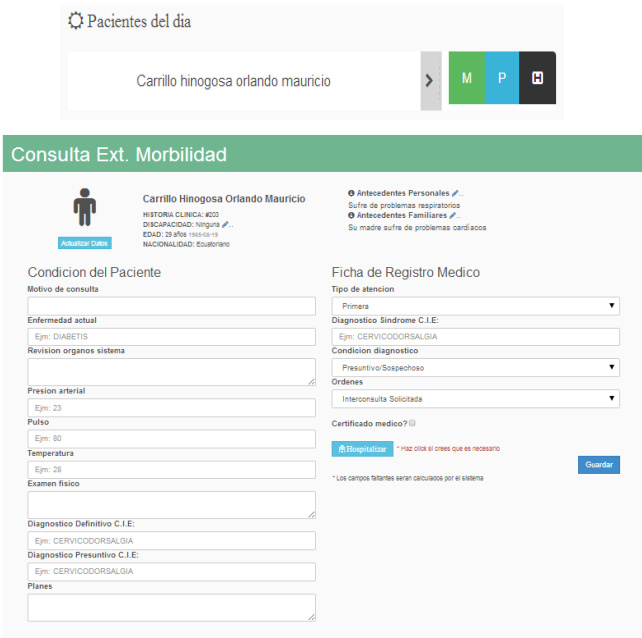
<b>Historia de Usuario</b>	
<b>Número:</b> 7	<b>Nombre:</b> Administración de horarios.
<b>Usuario:</b> Administrador fisiatría	
<b>Iteración asignada:</b> 1	
<b>Prioridad en Negocio:</b> Media (Alta / Media / Baja)	<b>Puntos Estimados:</b> 0.5
<b>Riesgo en Desarrollo:</b> Media (Alto / Medio / Bajo)	<b>Puntos Estimados:</b> 1.5
<p><b>Descripción:</b></p> <p>El Administrador de fisiatría puede: generar nuevos horarios, anularlos o volverlos a activar, además de controlar el número de turnos por hora-pacientes.</p>	
<p><b>Observaciones:</b></p> 	

TABLA XIX: HISTORIA DE USUARIO: ENVIAR A FISIATRÍA PACIENTE

<b>Historia de Usuario</b>	
<b>Número:</b> 8	<b>Nombre:</b> Enviar a fisiatría paciente
<b>Usuario:</b> Doctor	
<b>Iteración asignada:</b> 1	
<b>Prioridad en Negocio:</b> Media (Alta / Media / Baja)	<b>Puntos Estimados:</b> 1
<b>Riesgo en Desarrollo:</b> Media (Alto / Medio / Bajo)	<b>Puntos Estimados:</b> 1.5
<p><b>Descripción:</b></p> <p>El doctor puede atender o emitir un diagnóstico a su paciente y enviarlo al departamento de fisiatría, previamente el paciente debe obtener un turno.</p>	
<p><b>Observaciones:</b></p>  <p>The screenshot shows a web application interface for patient management. At the top, there's a header 'Pacientes del día' with a search bar containing 'Carrillo hinogosa orlando mauricio' and navigation buttons 'M', 'P', and a calendar icon. Below this is a green banner for 'Consulta Ext. Morbilidad'. The main content area is divided into two columns. The left column, 'Condicion del Paciente', includes fields for 'Motivo de consulta', 'Enfermedad actual' (with example 'DIABETIS'), 'Revision organos sistema', 'Presion arterial' (with example '23'), 'Pulso' (with example '80'), 'Temperatura' (with example '28'), 'Examen fisico', 'Diagnostico Definitivo C.I.E.', 'Diagnostico Presuntivo C.I.E.', and 'Planes'. The right column, 'Ficha de Registro Medico', includes 'Tipo de atencion' (dropdown), 'Diagnostico Sindrome C.I.E.' (with example 'CERVICODORSALGIA'), 'Condicion diagnostico' (dropdown), 'Ordenes' (dropdown), and 'Certificado medico?'. There are also links for 'Antecedentes Personales' and 'Antecedentes Familiares'. A 'Guardar' button is at the bottom right.</p>	

### 1.3.5. Resumen de Historias de Usuario

TABLA XX: RESUMEN DE HISTORIAS DE USUARIO

N	Nombre	Prioridad	Riesgo	Iteración
1	Administrar usuarios	Media	Media	1
2	Registro turno de terapia	Media	Alta	1
3	Calendario de turnos	Media	Alta	1
4	Administrador de terapeutas	Media	Alta	1
5	Reportes de fisioterapia	Media	Media	1
6	Administración de tratamientos y terapias	Media	Alta	1
7	Administración de horarios	Media	Media	1
8	Enviar a fisioterapia paciente	Media	Media	1

### 1.3.6. Plan de Entregas

TABLA XXI: PLAN DE ENTREGAS.

N	HISTORIA DE USUARIO	TIEMPO ESTIMADO (CALENDARIO ESTIMADO)			ITERACION ASIGNADA			ENTREGA ASIGNADA		
		SEMANAS ESTIMADAS	DIAS ESTIMADOS	HORAS ESTIMADAS	1	2	3	1	2	3
1	Administrar usuarios	2	10	20	X			X		
2	Registro turno de terapia	2	10	30	X			X		
3	Calendario de turnos	4	20	60	X			X		
4	Administración de terapistas	2	10	30	X			X		
5	Reportes de fisioterapia	2	10	30	X			X		
6	Administración de tratamientos y terapias	1	5	15	X			X		
7	Administración de horarios	1	6	18	X			X		
8	Enviar a fisioterapia paciente	2	10	20	X			X		
<b>TIEMPO TOTAL ESTIMADO</b>		16	81	223						

### 1.3.7. División en iteraciones.

El proyecto se dividió en dos iteraciones, por consiguiente con un total de dos entregas, se desarrolló partes de la aplicación completamente funcionales.

En la primera iteración se refiere a ingreso, control y emisión de turnos de fisioterapia, consta de las siguientes partes: administración de usuarios, registro de turnos de terapia, construcción del calendario de turnos, administración de terapeutas, reportes de fisioterapia; en la segunda iteración se realiza la integración con el módulo de estadística y controles del software que está formado por: administración de tratamientos y terapias, administración de horarios y el envío de pacientes a fisioterapia.

El periodo tomado para la planeación de la primera iteración, fue de 10 semanas, debido a la complejidad de las historias de usuario a tratar, para definir la interfaz y el desarrollo, ya que se realizó un prototipo y la funcionalidad de sus historias de usuario, al igual que para la segunda iteración se tomó 6 semanas, además por ser la interacción de integración con los departamentos de estadística y doctores.

### 1.3.8. Velocidad del proyecto

TABLA XXII: VELOCIDAD DEL PROYECTO

	<b>Iteración 1</b>	<b>Iteración 2</b>
<b>Horas</b>	400	180
<b>Semanas</b>	10	6
<b>Horas semanales</b>	40	30
<b>Historias de usuario (velocidad del proyecto)</b>	5	3

### 1.3.9. Entregas Pequeñas

Se procedió a liberar nuevas versiones según la funcionalidad que se fue implementando en el proyecto:

TABLA XXIII: ITERACIONES

Iteración	Fecha de entrega
1ra	27 de Enero del 2015
2da	31 de Marzo del 2015

## **FASE 2**

### **2. Diseño**

Con la metodología XP el diseño se realiza durante todo el tiempo de vida del proyecto, siendo constantemente revisado y muy probablemente modificado debido a cambios presentados durante el desarrollo.

#### **2.1. Diseños Simples**

En lo que respecta a "UI" del diseño, se invirtió el tiempo exclusivamente necesario en elaboración de diagramas y diseño de interfaz gráfica, elaborando una aplicación sencilla e intuitiva, pero que cumpla los requerimientos del usuario, desde el punto de vista de las interfaces, se invirtió el tiempo necesario en su diseño, se prestó mucha atención al elaborar formularios, efectos, ubicación de los elementos tal y como el cliente las había solicitado, obteniendo respuesta positiva del cliente, manifestando conformidad con la apariencia visual de la aplicación. (Ver Manual de usuario que viene junto con este informe)



## 2.2. Glosario de Términos

TABLA XXIV: GLOSARIO DE TÉRMINOS

<b>Termino</b>	<b>Definición</b>
Administrador	Usuario que se encuentra encargado de la administración del software.
Administrador estadística.	Usuario que se encuentra encargado de la administración del departamento de estadística.
Auxiliar estadística.	Usuario que ayuda al administrador de estadística, en la emisión de turnos y apertura de historias clínicas.
Doctor	Usuario que se encuentra a cargo de diagnosticar a un paciente.
Administrador fisiatría	Usuario que se encuentra encargado de la administración del departamento de fisiatría.
Reportes Médicos	Reportes que contienen información detallada de un paciente con diagnóstico de ingreso y egreso.
Diagnostico Paciente	Registro de notas medicas por parte de los fisiatras que son de relevancia para el doctor.
Calendario de turnos	Son los registros de todos los turnos necesarios para los fisiatras el cual contiene la información sobre: los pacientes y las terapias o tratamientos a los que van a ser sometidos, además del horario de cada sesión.

### 2.3. Tarjetas CRC (Clase - Responsabilidad – Colaborador)

TABLA XXV: TARJETAS CRC PERSONAL

PERSONAL	
Activar personal	Cliente
Suspender personal	Doctor

TABLA XXVI: TARJETAS CRC TURNO

TURNO	
Ingresar turno	Paciente
Eliminar turno	Doctor

TABLA XXVII: TARJETAS CRC NECESITA\_TERAPIA

NECESITA_TERAPIA	
Remite a fisioterapia	Paciente
	Doctor
	Consulta_Externa_Morbilidad

TABLA XXVIII: TARJETAS CRC CONSULTA\_EXTERNA\_MORBILIDAD

CONSULTA_EXTERNA_MORBILIDAD	
Ingresar registro	Turno
Guardar registro	Paciente
Enviar registro	Doctor

TABLA XXIX: TARJETAS CRC ASIGNAR\_HORARIO

ASIGNAR_HORARIO	
Ingresar registro	Turno
Guardar registro	Paciente
	Doctor
	Cliente
	Factura

TABLA XXX: TARJETAS CRC HORARIO

HORARIO	
Crear horario	Turno
Suspender Horario	Paciente
	Doctor

TABLA XXXI: TARJETAS CRC TRATAMIENTO\_REGISTRO

TRATAMIENTO_REGISTRO	
Agregar tratamiento	Tratamiento
Eliminar tratamiento	Asignar_Horario
	Paciente
	Doctor

TABLA XXXII: TARJETAS CRC RESULTADO\_TRATAMIENTO

RESULTADO_TRATAMIENTO	
Agregar diagnostico	Paciente
Editar diagnostico	Doctor
	Asignar_Horario
	Horario
	User

TABLA XXXIII: TARJETAS CRC TRATAMIENTO

TRATAMIENTO	
Agregar ítem	User
Suspender ítem	

## **2.4. Modelo del Dominio**

## **2.5. Modelo de Clases**

## **2.6 Modelo de la Base de Datos.**

## **FASE 3**

### **3. Desarrollo**

Mediante esta metodología se realiza la codificación desde el principio, lo que beneficia al logro de los objetivos a través de entregas frecuentes al cliente. Los elementos más importantes en cuanto a la codificación son: el cliente siempre debe estar presente, se debe trabajar en parejas y debe haber una propiedad colectiva del código.

Como se menciona en el capítulo I y III la aplicación es desarrollada bajo Ruby, usando el framework Ruby on Rails, librerías de jQuery como datatables, jquery-ui-1.8.21.custom, datepicker, wicked\_pdf gema para la generación de reportes en pdf, y haciendo uso del framework fronted Bootstrap.

#### **3.1. Cliente Siempre Presente**

Al mantener constante comunicación con el cliente mientras se desarrolla la aplicación, es un punto algo complejo de asimilar si se consideran los costos que esto representa, el desarrollador debía de acudir al hospital, podía llamar vía telefónica al cliente o mediante el internet, en el momento que requieran solucionar cualquier duda en el proceso de implementación. De esta forma se logra una buena comunicación con el cliente.

#### **3.2. Pruebas Unitarias**

Para las pruebas se utilizó "rspec", una herramienta del framework Ruby on Rails especializada para hacer testing del código, donde se probó ciertos métodos que forman el sistema, que cumple las especificaciones y tienen el comportamiento esperado.

Estas pruebas se las puede correr en modo desarrollo, las cuales se encuentra en el repositorio de la aplicación. A continuación se presenta en tablas el detalle de las mismas.



TABLA XXXIV: PRUEBA UNITARIA 1

<b>Numero de Caso de Prueba:</b> 01
<b>Caso de Prueba:</b> Número de identificación es único
<b>Objetivo de la prueba:</b> Verificar el número de identificación se encuentra registrado en el sistema y que no es posible registrarse nuevamente.
<b>Condiciones:</b> Estar registrado en el sistema.
<b>Datos de Entrada:</b> Datos del usuario específicamente número de identificación.
<b>Salida Esperada:</b> Informar que se encuentra en el sistema.
<b>Salida Obtenida:</b> El usuario existe en el sistema
<b>Evaluación:</b> Prueba satisfactoria

TABLA XXXV: PRUEBA UNITARIA 2

<b>Numero de Caso de Prueba:</b> 02
<b>Caso de Prueba:</b> Código de tratamientos y terapias es único.
<b>Objetivo de la prueba:</b> Verificar que un tratamiento posea un único código de identificación y que no pueda repetirse su código.
<b>Condiciones:</b> Estar creado y habilitado en el sistema.
<b>Datos de Entrada:</b> Nombre del tratamiento y código.
<b>Salida Esperada:</b> Informar que se encuentra en el sistema.
<b>Salida Obtenida:</b> El tratamiento existe en el sistema
<b>Evaluación:</b> Prueba satisfactoria

TABLA XXXVI: PRUEBA UNITARIA 3

<b>Numero de Caso de Prueba:</b> 03
<b>Caso de Prueba:</b> Horario creado una sola vez
<b>Objetivo de la prueba:</b> Verificar que los horarios de atención de fisioterapia estén creados una sola vez.
<b>Condiciones:</b> Estar creado en el sistema y habilitado.
<b>Datos de Entrada:</b> hora inicial y hora final
<b>Salida Esperada:</b> Horario almacenado.
<b>Salida Obtenida:</b> Horario habilitado.
<b>Evaluación:</b> Prueba satisfactoria

TABLA XXXVII: PRUEBA UNITARIA 4

<b>Numero de Caso de Prueba:</b> 04
<b>Caso de Prueba:</b> no sobrepasar los turnos por hora establecidos
<b>Objetivo de la prueba:</b> Verificar que los turnos por hora establecidos para cada fisiatra se cumplan y no se almacenen más turnos dentro del sistema.
<b>Condiciones:</b> Estar registrado en el sistema turnos y doctor.
<b>Datos de Entrada:</b> Asignar horario
<b>Salida Esperada:</b> Informar que el límite de turnos por día y por horario establecidos en configuración por el administrador de fisiatría.
<b>Salida Obtenida:</b> Turnos totales por horario disponible u ocupado.
<b>Evaluación:</b> Prueba satisfactoria

TABLA XXXVIII: PRUEBA UNITARIA 5

<b>Numero de Caso de Prueba:</b> 05
<b>Caso de Prueba:</b> Verificar que los terapistas no estén duplicados en el sistema
<b>Objetivo de la prueba:</b> Verificar que un terapeuta solo puede estar registrado dentro del sistema una vez.
<b>Condiciones:</b> Estar registrado en el sistema doctor.
<b>Datos de Entrada:</b> Datos del doctor, específicamente el nombre, número de cedula, número telefónico y dirección.
<b>Salida Esperada:</b> Informar que ya está registrado.
<b>Salida Obtenida:</b> Doctor registrado.
<b>Evaluación:</b> Prueba satisfactoria

TABLA XXXIX: PRUEBA UNITARIA 6

<b>Numero de Caso de Prueba:</b> 06
<b>Caso de Prueba:</b> Turno a doctor suspendido
<b>Objetivo de la prueba:</b> Verificar que un doctor que ha sido suspendido no pueda asignársele turnos hasta que sea de nuevo activado.
<b>Condiciones:</b> Estar registrado en el sistema el doctor.
<b>Datos de Entrada:</b> Datos del doctor.
<b>Salida Esperada:</b> Informar que el doctor está suspendido.
<b>Salida Obtenida:</b> No está disponible.
<b>Evaluación:</b> Prueba satisfactoria

TABLA XL: PRUEBA UNITARIA 7

<b>Numero de Caso de Prueba:</b> 09
<b>Caso de Prueba:</b> Verificar turno asignado.
<b>Objetivo de la prueba:</b> Verificar que solo puede asignar turnos que estén disponibles.
<b>Condiciones:</b> Estar registrado en el sistema como usuario: administrador fisiatría.
<b>Datos de Entrada:</b> Estar registrado el paciente y doctor responsable.
<b>Salida Esperada:</b> Informar que el turno está ocupado.
<b>Salida Obtenida:</b> Turno ocupado.
<b>Evaluación:</b> Prueba satisfactoria

TABLA XLI: PRUEBA UNITARIA 10

<b>Numero de Caso de Prueba:</b> 10
<b>Caso de Prueba:</b> Administrador puede modificar o eliminar diagnósticos de pacientes.
<b>Objetivo de la prueba:</b> Verificar que solo el administrador puede modificarla o eliminarla los diagnósticos generados por los fisiatras.
<b>Condiciones:</b> Estar registrado en el sistema como usuario: administrador o administrador_fisiatria.
<b>Datos de Entrada:</b> Usuario id.
<b>Salida Esperada:</b> Informar que no puede modificar o eliminar diagnóstico.
<b>Salida Obtenida:</b> No puedes modificar o eliminar estos diagnósticos.
<b>Evaluación:</b> Prueba satisfactoria

### 3.3. Programación por parejas

Con el uso de la metodología XP se resalta que toda la realización del código debe ser hecha en parejas, obteniendo un diseño de mejor calidad y un código más organizado

y en el que se puede solucionar los problemas más fácilmente. En el desarrollo de este proyecto se utilizó un “fork” del repositorio original llamado “sisfa” en el cual se adaptó el módulo de fisiatría aquí propuesto, donde se muestra: historial de cambios, tareas y bugs del sistema, llevando de manera ordenada el código.

### **3.4. Integración**

En este punto se hace mención que se deben hacer integraciones cada pocas horas o en lo posible no tardar más de un día entre una y otra integración, entre más se tarda en encontrar un problema, resultará más costoso resolverlo, e integrar frecuentemente evita problemas como el trabajar sobre una clase obsoleta. En cuanto a esta parte no existió problema en realizarlo, debido a que se lleva un control de versiones y una vez aprobadas se realizaba la integración.

## **FASE 4**

### **4. Pruebas**

#### **4.1. Implantación**

Una vez finalizado el desarrollo de la aplicación se la acoplo al sistema SISFA e implantó, en un servidor de la Institución para pruebas locales y en la nube para su implantación final, verificando su funcionamiento y realizando las respectivas explicaciones al Ing. Juan Pablo Sánchez, quien será el encargado de su manejo. De esta manera se llevó a cabo con el cumplimiento del objetivo principal del proyecto de fin de carrera.

#### **4.2. Pruebas de Aceptación**

Las pruebas de aceptación, son diseñadas en base a las historias de usuario, es una prueba formal conducida para determinar si un sistema satisface los criterios de aceptación y permite al cliente determinar si acepta el sistema. Para lo cual se elaboró las siguientes tablas.

TABLA XLII: CASO DE PRUEBA 1: ADMINISTRACIÓN USUARIOS

<b>Numero de Caso de Prueba: 01</b>	<b>Numero de Historia de Usuario: 01</b>
<b>Caso de Prueba:</b> Administración usuarios	
<b>Nombre de Caso de Prueba:</b> Ingresar Usuarios Correctamente	
<b>Descripción:</b> Verificará que el administrador puede realizar ingresos de nuevos usuarios.	
<b>Condiciones de Ejecución:</b> El administrador ha ingresado correctamente los datos requeridos para ingresar.	
<b>Entrada:</b> <ol style="list-style-type: none"> <li>1. Seleccionar el menú Usuarios.</li> <li>2. Seleccionar el Nuevo Usuario.</li> <li>3. Llenar correctamente todos campos solicitados.</li> <li>4. Seleccionar el botón Guardar.</li> </ol>	
<b>Resultado Esperado:</b> Se actualiza la información correspondiente al nuevo usuario.	
<b>Evaluación:</b> La evaluación fue aprobada, el ingreso se ejecutó correctamente.	

TABLA XLIII: CASO DE PRUEBA 2: ADMINISTRACIÓN USUARIOS

<b>Numero de Caso de Prueba: 02</b>	<b>Numero de Historia de Usuario: 01</b>
<b>Caso de Prueba:</b> Administración usuarios	
<b>Nombre de Caso de Prueba:</b> Ingresar Usuarios Incorrectamente	
<b>Descripción:</b> Verificará que si los datos que ingresa el administrador, no son correctos o no son llenados no se realiza el ingreso del mismo.	
<b>Condiciones de Ejecución:</b> El administrador ha ingresado incorrectamente los datos o tiene campos en blanco.	
<b>Entrada:</b> <ol style="list-style-type: none"> <li>1. Seleccionar el menú Usuarios.</li> <li>2. Seleccionar el Nuevo Usuario.</li> <li>3. Llenar incorrectamente los campos solicitados o dejarlos en blanco.</li> <li>4. Seleccionar el botón Guardar.</li> </ol>	
<b>Resultado Esperado:</b> No se guarda la información correspondiente al nuevo usuario. Se presenta mensajes indicando que campo no ha sido llenado o si el tipo de dato es incorrecto.	
<b>Evaluación:</b> La evaluación fue aprobada, el ingreso no se ejecutó.	

TABLA XLIV: CASO DE PRUEBA 3: ADMINISTRACIÓN USUARIOS

<b>Numero de Caso de Prueba: 03</b>	<b>Numero de Historia de Usuario: 01</b>
<b>Caso de Prueba:</b> Administración usuarios	
<b>Nombre de Caso de Prueba:</b> Suspende usuario	
<b>Descripción:</b> Verificará que el administrador puede suspender usuarios.	
<b>Condiciones de Ejecución:</b> El usuario ha sido registrado previamente.	
<b>Entrada:</b> <ol style="list-style-type: none"> <li>1. Seleccionar el menú Usuarios.</li> <li>2. Buscar usuario.</li> <li>3. Seleccionar icono suspender.</li> <li>4. Seleccionar el botón OK.</li> </ol>	
<b>Resultado Esperado:</b> Se actualiza la información correspondiente al usuario.	
<b>Evaluación:</b> El usuario se suspendió correctamente.	



TABLA XLV: CASO DE PRUEBA 4: ADMINISTRACIÓN USUARIOS

<b>Numero de Caso de Prueba: 04</b>	<b>Numero de Historia de Usuario: 01</b>
<b>Caso de Prueba:</b> Administración usuarios	
<b>Nombre de Caso de Prueba:</b> Activar usuario	
<b>Descripción:</b> Verificará que el administrador puede activar usuarios.	
<b>Condiciones de Ejecución:</b> El usuario ha sido registrado previamente.	
<b>Entrada:</b> <ol style="list-style-type: none"> <li>1. Seleccionar el menú Usuarios.</li> <li>2. Buscar usuario.</li> <li>3. Seleccionar icono activar.</li> <li>4. Seleccionar el botón OK.</li> </ol>	
<b>Resultado Esperado:</b> Se actualiza la información correspondiente al usuario.	
<b>Evaluación:</b> El usuario se activó correctamente.	

TABLA XLVI: CASO DE PRUEBA 5: ADMINISTRACIÓN USUARIOS

<b>Numero de Caso de Prueba: 05</b>	<b>Numero de Historia de Usuario: 01</b>
<b>Caso de Prueba:</b> Administración usuarios	
<b>Nombre de Caso de Prueba:</b> Nueva clave	
<b>Descripción:</b> Verificará que el administrador puede dar nueva clave a usuarios.	
<b>Condiciones de Ejecución:</b> El usuario ha sido registrado previamente.	
<b>Entrada:</b> <ol style="list-style-type: none"> <li>1. Seleccionar el menú Usuarios.</li> <li>2. Buscar usuario.</li> <li>3. Seleccionar icono nueva clave.</li> <li>4. Seleccionar el botón Guardar.</li> </ol>	
<b>Resultado Esperado:</b> Se actualiza la información correspondiente al usuario.	
<b>Evaluación:</b> Nueva clave ingresada correctamente.	

TABLA XLVII: CASO DE PRUEBA 6: REGISTRO DE TURNOS DE TERAPIA

<b>Numero de Caso de Prueba: 06</b>	<b>Numero de Historia de Usuario: 02</b>
<b>Caso de Prueba:</b> Registro de turnos de terapia	
<b>Nombre de Caso de Prueba:</b> Registrar turno de terapia correctamente	
<b>Descripción:</b> Verificará que el administrador de fisioterapia: puede realizar registros de turnos de terapia.	
<b>Condiciones de Ejecución:</b> El administrador de fisioterapia ha ingresado correctamente los datos requeridos para ingresar.	
<b>Entrada:</b> <ol style="list-style-type: none"> <li>1. Seleccionar el menú Calendario.</li> <li>2. Seleccionar el menú Registrar paciente</li> <li>3. Seleccionar la pestaña Datos del paciente: Paciente, Doctor remitente, Factura, Valor factura y Diagnóstico del paciente.</li> <li>4. Llenar correctamente todos campos solicitados.</li> <li>5. Seleccionar la pestaña Terapias</li> <li>6. Ingresar las terapias a las que va a ser sometido el paciente; presiona Agregar si es más de una.</li> <li>7. Llena correctamente todos los campos solicitados.</li> <li>8. Seleccionar la pestaña Horarios.</li> <li>9. Ingresar los datos solicitados: fecha y horario; presiona Agregar en el caso de ser más de una sesión.</li> <li>10. Llena correctamente los campos solicitados.</li> <li>11. Seleccionar el botón Guardar.</li> </ol>	
<b>Resultado Esperado:</b> Se registra el turno correspondiente al paciente.	
<b>Evaluación:</b> La evaluación fue aprobada, el registro se ejecutó correctamente.	

TABLA XLVIII: CASO DE PRUEBA 7: REGISTRO DE TURNOS DE TERAPIA

<b>Numero de Caso de Prueba: 07</b>	<b>Numero de Historia de Usuario: 02</b>
<b>Caso de Prueba:</b> Registro de turnos de terapia	
<b>Nombre de Caso de Prueba:</b> Registrar turnos de terapia incorrectamente	
<b>Descripción:</b> Verificará que si el administrador de fisioterapia ingresa mal los datos, no se registra un turno de terapia.	
<b>Condiciones de Ejecución:</b> El administrador ha ingresado incorrectamente los datos o tiene campos obligatorios en blanco.	
<b>Entrada:</b> <ol style="list-style-type: none"> <li>1. Seleccionar el menú Calendario.</li> <li>2. Seleccionar el menú Registrar paciente</li> <li>3. Seleccionar la pestaña Datos del paciente: Paciente, Doctor remitente, Factura, Valor factura y Diagnóstico del paciente.</li> <li>4. Llenar incorrectamente los campos solicitados o deja en blanco alguno.</li> <li>5. Seleccionar la pestaña Terapias</li> <li>6. Ingresar las terapias a las que va a ser sometido el paciente; presiona Agregar si es más de una.</li> <li>7. Llena incorrectamente los campos solicitados o deja en blanco.</li> <li>8. Seleccionar la pestaña Horarios.</li> <li>9. Ingresar los datos solicitados: fecha y horario; presiona Agregar en el caso de ser más de una sesión.</li> <li>10. Llena incorrectamente los campos solicitados o deja en blanco.</li> <li>11. Seleccionar el botón Guardar.</li> </ol>	
<b>Resultado Esperado:</b> No se guarda la información correspondiente al paciente y al turno para terapia. Se presenta mensajes indicando que campo o campos no ha sido llenado o si el tipo de dato es incorrecto.	
<b>Evaluación:</b> La evaluación fue aprobada, el registro no se ejecutó.	

TABLA XLIX: CASO DE PRUEBA 8: CALENDARIO DE TURNOS

<b>Numero de Caso de Prueba: 08</b>	<b>Numero de Historia de Usuario: 03</b>
<b>Caso de Prueba:</b> calendario de turnos	
<b>Nombre de Caso de Prueba:</b> Visualizar turnos.	
<p><b>Descripción:</b></p> <p>Verificará que el administrador de fisioterapia y los fisioterapeutas: puedan visualizar los turnos registrados.</p>	
<p><b>Condiciones de Ejecución:</b></p> <p>Los turnos han sido ingresados previamente.</p>	
<p><b>Entrada:</b></p> <ol style="list-style-type: none"> <li>1. Seleccionar el menú Calendario.</li> <li>2. Visualiza los días que están disponibles los turnos basada en colores: rojo ocupado, verde disponible.</li> </ol>	
<p><b>Resultado Esperado:</b></p> <p>Se visualiza la información correspondiente a los turnos dentro de fisioterapia.</p>	
<p><b>Evaluación:</b></p> <p>Se observan a detalle los datos de los turnos correctamente.</p>	

TABLA L: CASO DE PRUEBA 9: CALENDARIO DE TURNOS

<b>Numero de Caso de Prueba: 09</b>	<b>Numero de Historia de Usuario: 03</b>
<b>Caso de Prueba:</b> Calendario de turnos	
<b>Nombre de Caso de Prueba:</b> Registrar diagnósticos.	
<p><b>Descripción:</b></p> <p>Verificará que el fisiatra: pueda registrar el diagnóstico posterior a la atención del paciente.</p>	
<p><b>Condiciones de Ejecución:</b></p> <p>El paciente ha sido atendido.</p>	
<p><b>Entrada:</b></p> <ol style="list-style-type: none"> <li>1. Seleccionar el menú Calendario.</li> <li>2. Seleccionar día de terapia.</li> <li>3. Seleccionar horario de terapia</li> <li>4. Seleccionar paciente</li> <li>5. Seleccionar registrar</li> <li>6. Registra correctamente el diagnóstico.</li> <li>7. Selecciona botón Guardar.</li> </ol>	
<p><b>Resultado Esperado:</b></p> <p>Se almacena diagnóstico del paciente.</p>	
<p><b>Evaluación:</b></p> <p>Se registró diagnóstico correctamente.</p>	

TABLA LI: CASO DE PRUEBA 10: ADMINISTRADOR DE TERAPISTAS

<b>Numero de Caso de Prueba: 10</b>	<b>Numero de Historia de Usuario: 04</b>
<b>Caso de Prueba:</b> Administrador de terapistas	
<b>Nombre de Caso de Prueba:</b> Ingresar terapeuta correctamente	
<b>Descripción:</b> Verificará que el administrador de fisioterapia: pueden ingresar un nuevo terapeuta.	
<b>Condiciones de Ejecución:</b> El usuario ha ingresado correctamente los datos requeridos para ingresar.	
<b>Entrada:</b> <ol style="list-style-type: none"> <li>1. Seleccionar el menú Terapeutas.</li> <li>2. Seleccionar botón Nuevo Terapeuta.</li> <li>3. Llenar correctamente todos campos solicitados.</li> <li>4. Seleccionar el botón Guardar.</li> </ol>	
<b>Resultado Esperado:</b> Se registra un nuevo terapeuta con la información correspondiente.	
<b>Evaluación:</b> Se ingresó un nuevo terapeuta correctamente.	

TABLA LII: CASO DE PRUEBA 11: ADMINISTRADOR DE TERAPISTAS

<b>Numero de Caso de Prueba: 11</b>	<b>Numero de Historia de Usuario: 04</b>
<b>Caso de Prueba:</b> Administrador de terapistas	
<b>Nombre de Caso de Prueba:</b> ingresar terapeuta incorrectamente	
<b>Descripción:</b> Verificará que el administrador de terapistas: no ingrese un terapeuta incorrectamente.	
<b>Condiciones de Ejecución:</b> El usuario ha ingresado correctamente los datos requeridos para ingresar.	
<b>Entrada:</b> <ol style="list-style-type: none"> <li>1. Seleccionar el menú Terapistas.</li> <li>2. Seleccionar botón Nuevo Terapeuta.</li> <li>3. Llenar incorrectamente los campos solicitados.</li> <li>4. Seleccionar el botón Guardar.</li> </ol>	
<b>Resultado Esperado:</b> Se presenta un mensaje de error.	
<b>Evaluación:</b> No se almacena el nuevo terapeuta.	



TABLA LIII: CASO DE PRUEBA 12: ADMINISTRADOR DE TERAPISTAS

<b>Numero de Caso de Prueba: 12</b>	<b>Numero de Historia de Usuario: 04</b>
<b>Caso de Prueba:</b> Administrador de terapistas	
<b>Nombre de Caso de Prueba:</b> Editar terapeuta	
<b>Descripción:</b> Verificará que el administrador de fisioterapia pueda editar a los datos de un fisiatra.	
<b>Condiciones de Ejecución:</b> El fisiatra ha sido registrado previamente.	
<b>Entrada:</b> <ol style="list-style-type: none"> <li>1. Seleccionar el menú Terapistas.</li> <li>2. Buscar terapeuta.</li> <li>3. Click en el botón editar.</li> <li>4. Llenar correctamente todos campos solicitados.</li> <li>5. Seleccionar el botón Guardar</li> </ol>	
<b>Resultado Esperado:</b> Se actualiza la información correspondiente al terapeuta.	
<b>Evaluación:</b> Se modifican los datos correctamente.	

TABLA LIV: CASO DE PRUEBA 13: ADMINISTRADOR DE TERAPISTAS

<b>Numero de Caso de Prueba: 13</b>	<b>Numero de Historia de Usuario: 04</b>
<b>Caso de Prueba:</b> Administrador de terapistas	
<b>Nombre de Caso de Prueba:</b> Suspende terapeuta	
<b>Descripción:</b> Verificará que el administrador de fisioterapia pueda suspender a un terapeuta.	
<b>Condiciones de Ejecución:</b> El terapeuta ha sido registrado previamente.	
<b>Entrada:</b> <ol style="list-style-type: none"> <li>1. Seleccionar el menú Terapeutas.</li> <li>2. Busca terapeuta.</li> <li>3. Seleccionar suspender.</li> <li>4. Seleccionar OK.</li> </ol>	
<b>Resultado Esperado:</b> Se actualiza la información correspondiente al terapeuta.	
<b>Evaluación:</b> El terapeuta se suspendió correctamente.	

TABLA LV: CASO DE PRUEBA 14: ADMINISTRADOR DE TERAPISTAS

<b>Numero de Caso de Prueba: 14</b>	<b>Numero de Historia de Usuario: 04</b>
<b>Caso de Prueba:</b> Administrador de terapistas	
<b>Nombre de Caso de Prueba:</b> Activar terapeuta	
<b>Descripción:</b> Verificará que el administrador de fisiatría pueda activar a los terapeutas.	
<b>Condiciones de Ejecución:</b> El terapeuta ha sido ingresado previamente y está en estado de suspensión.	
<b>Entrada:</b> 1. Seleccionar el menú Terapeutas. 2. Buscar terapeuta. 3. Seleccionar activar. 4. Seleccionar OK.	
<b>Resultado Esperado:</b> Se actualiza la información correspondiente al terapeuta.	
<b>Evaluación:</b> El terapeuta se activó correctamente.	

TABLA LVI: CASO DE PRUEBA 15: REPORTES DE FISIATRÍA

<b>Numero de Caso de Prueba: 15</b>	<b>Numero de Historia de Usuario: 05</b>
<b>Caso de Prueba:</b> Reportes de fisioterapia.	
<b>Nombre de Caso de Prueba:</b> Generar reportes de fisioterapia por terapeuta	
<b>Descripción:</b> Verificará que el administrador de fisioterapia pueda generar los reportes de fisioterapia por cada terapeuta.	
<b>Condiciones de Ejecución:</b> El usuario ha ingresado correctamente los datos requeridos para ingresar.	
<b>Entrada:</b> <ol style="list-style-type: none"> <li>1. Seleccionar el menú Reportes.</li> <li>2. Seleccionar el terapeuta.</li> <li>3. Llenar correctamente los datos solicitados: fecha inicial, fecha final.</li> <li>4. Seleccionar el botón Buscar.</li> <li>5. Seleccionar descargar PDF.</li> </ol>	
<b>Resultado Esperado:</b> Se genera el reporte PDF con la información correspondiente al terapeuta.	
<b>Evaluación:</b> Se emite el reporte de fisioterapia correctamente.	

TABLA LVII: CASO DE PRUEBA 16: REPORTES DE FISIATRÍA

<b>Numero de Caso de Prueba: 16</b>	<b>Numero de Historia de Usuario: 05</b>
<b>Caso de Prueba:</b> Reportes de fisioterapia	
<b>Nombre de Caso de Prueba:</b> Generar reportes de fisioterapia y certificado por paciente	
<b>Descripción:</b> Verificará que el administrador de fisioterapia pueda generar los reportes de fisioterapia y certificado por cada paciente.	
<b>Condiciones de Ejecución:</b> El usuario ha ingresado correctamente los datos requeridos para ingresar; y el paciente ha sido atendido previamente.	
<b>Entrada:</b> <ol style="list-style-type: none"> <li>1. Seleccionar el menú Reportes.</li> <li>2. Busca el paciente: nombre o historia clínica.</li> <li>3. Llenar correctamente los datos solicitados: fecha inicial, fecha final.</li> <li>4. Seleccionar el botón Buscar.</li> <li>5. Seleccionar descargar PDF.</li> </ol>	
<b>Resultado Esperado:</b> Se genera el archivo PDF con la información solicitada.	
<b>Evaluación:</b> El archivo se ha generado correctamente.	

TABLA LVIII: CASO DE PRUEBA 17: REPORTES DE FISIATRÍA

<b>Numero de Caso de Prueba: 17</b>	<b>Numero de Historia de Usuario: 05</b>
<b>Caso de Prueba:</b> Reportes de fisioterapia	
<b>Nombre de Caso de Prueba:</b> Generar reportes de fisioterapia por factura	
<b>Descripción:</b> Verificará que el administrador de fisioterapia puede generar los reportes de fisioterapia mediante las facturas.	
<b>Condiciones de Ejecución:</b> El usuario ha ingresado correctamente los datos requeridos para ingresar.	
<b>Entrada:</b> <ol style="list-style-type: none"> <li>1. Seleccionar el menú Reportes.</li> <li>2. Seleccionar tipo de búsqueda: fecha o número factura.</li> <li>3. Llenar correctamente los datos solicitados: fecha inicial, fecha final / número inicial, número final.</li> <li>4. Seleccionar el botón Buscar.</li> <li>5. Seleccionar descargar PDF.</li> </ol>	
<b>Resultado Esperado:</b> Se genera el archivo PDF con la información solicitada.	
<b>Evaluación:</b> El archivo se ha generado correctamente.	

TABLA LIX: CASO DE PRUEBA 18: ADMINISTRACIÓN DE TRATAMIENTOS Y TERAPIAS

<b>Numero de Caso de Prueba: 18</b>	<b>Numero de Historia de Usuario: 06</b>
<b>Caso de Prueba:</b> Administración de tratamientos y terapias	
<b>Nombre de Caso de Prueba:</b> Ingresar tratamientos y terapias correctamente	
<p><b>Descripción:</b></p> <p>Verificará que el administrador de fisioterapia ingrese correctamente nuevos tratamientos y terapias.</p>	
<p><b>Condiciones de Ejecución:</b></p> <p>El administrador ha ingresado correctamente los datos para ingresar.</p>	
<p><b>Entrada:</b></p> <ol style="list-style-type: none"> <li>1. Seleccionar el menú Configuración.</li> <li>2. Seleccionar botón nuevo (+).</li> <li>3. Llenar correctamente los campos y no dejarlos en blanco: presionar el botón +Tratamiento en caso de querer ingresar más de un tratamiento en dicha terapia.</li> <li>4. Seleccionar el botón Guardar.</li> </ol>	
<p><b>Resultado Esperado:</b></p> <p>Se actualiza la información correspondiente al nuevo tratamiento y terapia.</p>	
<p><b>Evaluación:</b></p> <p>El tratamiento y terapia son aprobados e incorporados a los servicios que se ofertan</p>	

TABLA LX: CASO DE PRUEBA 19: ADMINISTRACIÓN DE TRATAMIENTOS Y TERAPIAS

<b>Numero de Caso de Prueba: 19</b>	<b>Numero de Historia de Usuario: 06</b>
<b>Caso de Prueba:</b> Administración de tratamientos y terapias	
<b>Nombre de Caso de Prueba:</b> Ingresar tratamientos y terapias incorrectamente.	
<b>Descripción:</b> Verificará que si el administrador de fisioterapia ingresa mal los datos no se puede ingresar nuevos tratamientos y terapias.	
<b>Condiciones de Ejecución:</b> El usuario debe ingresar mal los datos o dejar campos obligatorios en blanco.	
<b>Entrada:</b> <ol style="list-style-type: none"> <li>1. Seleccionar el menú Configuración.</li> <li>2. Seleccionar botón nuevo (+).</li> <li>3. Llenar incorrectamente los campos o dejarlos en blanco: presionar el botón +Tratamiento en caso de querer ingresar más de un tratamiento en dicha terapia.</li> <li>4. Seleccionar el botón Guardar.</li> </ol>	
<b>Resultado Esperado:</b> No se ingresan nuevos tratamientos y terapias. Se presentan mensajes indicando los errores cometidos o los datos no ingresados.	
<b>Evaluación:</b> La evaluación fue aprobada, no se realizó ningún ingreso.	



TABLA LXI: CASO DE PRUEBA 20: ADMINISTRACIÓN DE TRATAMIENTOS Y TERAPIAS

<b>Numero de Caso de Prueba: 20</b>	<b>Numero de Historia de Usuario: 06</b>
<b>Caso de Prueba:</b> Administración de tratamientos y terapias	
<b>Nombre de Caso de Prueba:</b> Editar tratamientos y terapias.	
<b>Descripción:</b> Verificará que el administrador de fisioterapia puede editar los tratamientos y terapias.	
<b>Condiciones de Ejecución:</b> El tratamiento o terapia debe de estar registrado previamente.	
<b>Entrada:</b> <ol style="list-style-type: none"> <li>1. Seleccionar el menú Configuración.</li> <li>2. Seleccionar el tratamiento o terapia.</li> <li>3. Modificar correctamente los campos: código, nombre.</li> <li>4. Presionar enter o click fuera del campo editado.</li> </ol>	
<b>Resultado Esperado:</b> Se actualiza la información correspondiente al tratamiento o terapia.	
<b>Evaluación:</b> El tratamiento o terapia se editó correctamente.	

TABLA LXII: CASO DE PRUEBA 21: ADMINISTRACIÓN DE TRATAMIENTOS Y TERAPIAS

<b>Numero de Caso de Prueba: 21</b>	<b>Numero de Historia de Usuario: 06</b>
<b>Caso de Prueba:</b> Administración de tratamientos y terapias	
<b>Nombre de Caso de Prueba:</b> Habilitar tratamientos y terapias	
<b>Descripción:</b> Verificará que el administrador de fisioterapia puede habilitar los tratamientos.	
<b>Condiciones de Ejecución:</b> El tratamiento debe de estar registrado previamente y en estado de suspensión.	
<b>Entrada:</b> 1. Seleccionar el menú Configuración. 2. Buscar tratamiento. 3. Seleccionar botón Habilitar.	
<b>Resultado Esperado:</b> Se actualiza la información correspondiente al tratamiento. Se presenta mensaje "Tratamiento habilitado"	
<b>Evaluación:</b> El tratamiento se habilitó correctamente.	

TABLA LXIII: CASO DE PRUEBA 22: ADMINISTRACIÓN DE TRATAMIENTOS Y TERAPIAS

<b>Numero de Caso de Prueba: 22</b>	<b>Numero de Historia de Usuario: 06</b>
<b>Caso de Prueba:</b> Administración de tratamientos y terapias	
<b>Nombre de Caso de Prueba:</b> Suspender tratamiento.	
<b>Descripción:</b> Verificará que el administrador de fisioterapia puede suspender tratamientos.	
<b>Condiciones de Ejecución:</b> El tratamiento debe de estar registrado previamente y en estado habilitado.	
<b>Entrada:</b> 1. Seleccionar el menú Configuración. 2. Buscar tratamiento. 3. Seleccionar botón Suspender.	
<b>Resultado Esperado:</b> Se actualiza la información correspondiente al tratamiento. Se presenta mensaje "Tratamiento suspendido".	
<b>Evaluación:</b> El tratamiento se ha suspendido correctamente.	

TABLA LXIV: CASO DE PRUEBA 23: ADMINISTRACIÓN DE HORARIOS

<b>Numero de Caso de Prueba: 23</b>	<b>Numero de Historia de Usuario: 07</b>
<b>Caso de Prueba:</b> Administración de horarios	
<b>Nombre de Caso de Prueba:</b> Crear nuevos horarios correctamente	
<b>Descripción:</b> Verificará que el administrador de fisioterapia puede crear nuevos horarios.	
<b>Condiciones de Ejecución:</b> El administrador ha ingresado correctamente los datos requeridos para ingresar. El horario no debe existir.	
<b>Entrada:</b> 1. Seleccionar el menú Configuración. 2. Seleccionar pestaña Horarios. 3. Ingresar correctamente los datos solicitados: hora inicio, hora final. 4. Seleccionar botón Guardar.	
<b>Resultado Esperado:</b> Se actualiza la información de los horarios disponibles.	
<b>Evaluación:</b> El horario se ha creado correctamente.	

TABLA LXV: CASO DE PRUEBA 24: ADMINISTRACIÓN DE HORARIOS

<b>Numero de Caso de Prueba: 24</b>	<b>Numero de Historia de Usuario: 07</b>
<b>Caso de Prueba:</b> Administración de horarios.	
<b>Nombre de Caso de Prueba:</b> Crear nuevos horarios incorrectamente	
<b>Descripción:</b> Verificará que si el administrador de fisioterapia ingresa datos incorrectos o deja campos vacíos no se puedan crear nuevos horarios.	
<b>Condiciones de Ejecución:</b> El administrador ha ingresado correctamente los datos requeridos para ingresar.	
<b>Entrada:</b> <ol style="list-style-type: none"> <li>1. Seleccionar el menú Configuración.</li> <li>2. Seleccionar pestaña Horarios.</li> <li>3. Ingresar incorrectamente los datos solicitados o deja en blanco algún campo: hora inicio, hora final.</li> <li>4. Seleccionar botón Guardar.</li> </ol>	
<b>Resultado Esperado:</b> No se actualiza la información de los horarios disponibles. Se presentan mensajes con los errores dentro del proceso.	
<b>Evaluación:</b> El horarios no se almacena.	

TABLA LXVI: CASO DE PRUEBA 25: ADMINISTRACIÓN DE HORARIOS

<b>Numero de Caso de Prueba: 25</b>	<b>Numero de Historia de Usuario: 07</b>
<b>Caso de Prueba:</b> Administración de horarios.	
<b>Nombre de Caso de Prueba:</b> Suspende horario.	
<b>Descripción:</b> Verificará que el administrador de fisioterapia pueda suspender horarios.	
<b>Condiciones de Ejecución:</b> El horario debe estar registrado previamente y en estado habilitado.	
<b>Entrada:</b> <ol style="list-style-type: none"> <li>1. Seleccionar el menú Configuración</li> <li>2. Seleccionar la pestaña Horarios.</li> <li>3. Buscar horario.</li> <li>4. Seleccionar botón suspender.</li> </ol>	
<b>Resultado Esperado:</b> Se actualiza la información de los horarios disponibles. Se presenta mensaje "Horario suspendido"	
<b>Evaluación:</b> El horario se ha suspendido correctamente.	

TABLA LXVII: CASO DE PRUEBA 26: ADMINISTRACIÓN DE HORARIOS

<b>Numero de Caso de Prueba: 26</b>	<b>Numero de Historia de Usuario: 07</b>
<b>Caso de Prueba:</b> Administración de horarios.	
<b>Nombre de Caso de Prueba:</b> Habilitar horario.	
<b>Descripción:</b> Verificará que el administrador de fisioterapia puede habilitar horarios.	
<b>Condiciones de Ejecución:</b> El horario debe estar registrado previamente y en estado suspendido.	
<b>Entrada:</b> <ol style="list-style-type: none"> <li>1. Seleccionar el menú Configuración</li> <li>2. Seleccionar la pestaña Horarios.</li> <li>3. Buscar horario.</li> <li>4. Seleccionar botón habilitar.</li> </ol>	
<b>Resultado Esperado:</b> Se actualiza la información de los horarios disponibles. Se presenta mensaje "Horario habilitado"	
<b>Evaluación:</b> El horario ha sido habilitado correctamente.	

TABLA LXVIII: CASO DE PRUEBA 27: ADMINISTRACIÓN DE HORARIOS

<b>Numero de Caso de Prueba: 27</b>	<b>Numero de Historia de Usuario: 08</b>
<b>Caso de Prueba:</b> Administración de horarios.	
<b>Nombre de Caso de Prueba:</b> Establecer número de horarios por hora.	
<b>Descripción:</b> Verificará que el administrador de fisioterapia puede establecer el número de turnos por horario.	
<b>Condiciones de Ejecución:</b> Debe de ser administrador de fisioterapia.	
<b>Entrada:</b> <ol style="list-style-type: none"> <li>1. Seleccionar el menú Configuración.</li> <li>2. Seleccionar la pestaña Horarios.</li> <li>3. Presionar sobre el número marcado en Turnos por hora.</li> <li>4. Asignar un valor numérico al campo.</li> <li>5. Presionar enter o click fuera del campo.</li> </ol>	
<b>Resultado Esperado:</b> Se actualiza la información de turnos por horario.	
<b>Evaluación:</b> Se establece la cantidad de turnos a ser atendidos por horario correctamente.	



TABLA LXIX: CASO DE PRUEBA 28: ENVIAR A FISIATRÍA PACIENTE

<b>Numero de Caso de Prueba: 28</b>	<b>Numero de Historia de Usuario: 08</b>
<b>Caso de Prueba:</b> Enviar a fisioterapia paciente.	
<b>Nombre de Caso de Prueba:</b> Diagnosticar paciente a fisioterapia	
<b>Descripción:</b> Verificará que el Doctor pueda diagnosticar y remitir pacientes a fisioterapia.	
<b>Condiciones de Ejecución:</b> El Doctor debe ingresar correctamente sus datos para ingresar. El paciente debe estar ingresado previamente	
<b>Entrada:</b> 1. Seleccionar el menú Ingresar. 2. Seleccionar paciente. 3. Seleccionar botón Consulta Morbilidad 4. Llenar correctamente los datos solicitados: Condición del paciente, Ficha de registro médico. 5. Marcar casilla Necesita terapia. 6. Presionar botón Guardar. 7. Presiona botón OK.	
<b>Resultado Esperado:</b> Se envía el paciente a fisioterapia. Se presenta mensaje "almacenado".	
<b>Evaluación:</b> Se remite el paciente a fisioterapia correctamente.	

## **4.3. Plan de Validación**

### **4.3.1. Propósito**

En el desarrollo del software se ve implicado la realización de un conjunto de actividades las cuales tienen como finalidad comprobar su correcto funcionamiento, así como también el cumplimiento de los requerimientos. Las pruebas (testing) son un proceso de ejecución del sistema que buscan descubrir la mayor cantidad de errores en el código fuente, teniendo como fin una herramienta confiable. Para realizar las pruebas de software, se usó el servidor de la institución.

### **4.3.2. Introducción**

Las pruebas de software permiten identificar, verificar y revelar la calidad de un producto de software. Estas nos indican los posibles fallos de implementación, calidad o usabilidad. Las pruebas se integran dentro de diferentes fases del ciclo de vida de software, y mediante técnicas experimentales se trata de descubrir que errores tiene.

#### **4.3.2.1. Objetivos**

El plan de pruebas para el módulo de fisioterapia del sistema SISFA del Hospital Básico 7 B.I Loja deben procurar los siguientes objetivos:

- Detallar las actividades requeridas para la preparación y conducción de las pruebas al sistema.
- Definir el origen de la información usada para la preparación de las pruebas
- Definir las herramientas para pruebas y el ambiente necesario para su ejecución

#### **4.3.2.2. Antecedentes**

Las especificaciones del sistema del proyecto en el documento se encuentran en resultados, fase 1 del presente informe.

#### **4.3.2.3. Alcances**

Este plan de pruebas detalla el procedimiento del módulo de fisioterapia.

#### 4.3.2.4. Referencias

Se desarrollaron varios documentos como fuente de información para la preparación de las pruebas, entre los cuales están:

- Especificación de requerimientos de software (Resultados, Fase 1).
- Historias de usuario (Resultados, Fase 1, punto 1.3.4).
- Manual de usuario (se encuentra adjunto con este informe).
- Manual del programador (se encuentra adjunto con este informe).

#### 4.3.3. Elementos software a probar

Los módulos que serán probados en este plan de pruebas son los siguientes:

TABLA LXX: FUNCIONALIDAD POR USUARIOS

<b>Categoría</b>	<b>Funcionalidad</b>
Administrador	Acceder al sistema.
	Administrar usuarios.
Fisiatría	Registro de turnos de terapia.
	Visualizar turnos.
	Editar registros.
	Generar reportes por: terapeuta, paciente/certificado, factura.
	Administrar terapeutas.
	Administración de tratamientos y terapias.
	Administrar de horarios.
Doctores	Remitir pacientes por CTEXM a fisiatría.
Terapeutas	Registrar diagnósticos de pacientes.

#### **4.3.4. Características a probar**

Las características que serán probadas son las siguientes:

- Accesos al sistema.
- Registro de turnos de terapia.
- Administración de terapeutas.
- Administración de tratamientos y terapias.
- Registro de diagnósticos.
- Envío de pacientes a fisioterapia.
- Generación de reportes por: terapeuta, paciente/certificado, factura.

#### **4.3.5. Enfoque general de la prueba**

##### **4.3.5.1. Aspectos generales**

###### **4.3.5.1.1. En lo relativo a la documentación**

Se usa como referencia los documentos indicados en el apartado 1.3.4. (Historias de Usuario).

###### **4.3.5.1.2. En lo relativo a los encargados del desarrollo**

El desarrollo de los procedimientos de los módulos del sistema, fueron de la mano con el diseño de los casos de prueba (apartado 4.2., de resultados) y las historias de usuario.

###### **4.3.5.1.3. En lo relativo a la confidencialidad**

Para respetar el acuerdo de confidencialidad con la institución, todas las pruebas deberán realizarse usando datos que permitan representar eficientemente los datos reales.

##### **4.3.5.2. Enfoques**

###### **4.3.5.2.1. Enfoque para probar los accesos al sistema**

Realizar múltiples accesos al sistema en forma concurrente con los diferentes tipos de usuarios.

#### **4.3.5.2.2. Enfoque para probar registro de turnos de terapia**

Crea un turno dentro del departamento de fisioterapia y asigna un terapeuta.

#### **4.3.5.2.3. Enfoque para probar administración de terapeutas**

Realizar la creación de un terapeuta, suspender y habilitar, además de modificar sus datos.

#### **4.3.5.2.4. Enfoque para probar administración de tratamientos y terapias**

Realizar la creación de tratamientos y terapias, suspender y habilitar, además de modificar sus datos.

#### **4.3.5.2.5. Enfoque para probar registro de diagnósticos**

Registrar un diagnóstico de un paciente luego de ser atendido.

#### **4.3.5.2.6. Enfoque para probar envío de pacientes a fisioterapia**

Emitir un turno a un doctor el mismo que remite un paciente a través de CTEXM al departamento de fisioterapia.

#### **4.3.5.2.7. Enfoque para probar generación de reportes por: terapeuta, certificado/paciente, factura**

Generar un reporte de acuerdo al terapeuta, paciente o factura dentro de un periodo de tiempo establecido.

#### **4.3.5.3. Amplitud de la prueba**

Cada módulo se refiere a una serie de procesos y tendrá asociado las pruebas de aceptación y funcionalidad.

#### **4.3.6. Criterios de paso/fallo para cada elemento**

Los criterios que determinan el éxito o fallo de una prueba se describirán en las especificaciones de casos de pruebas correspondientes.

#### **4.3.7. Criterios de suspensión y requisitos de reanudación**

Una prueba deberá ser suspendida si el servidor no está funcionando apropiadamente.

### **4.3.8. Productos entregables**

Los generarán los siguientes documentos:

1. Pruebas funcionales
2. Pruebas de aceptación
3. Pruebas de usabilidad
4. Pruebas de Carga y Estrés

### **4.3.9. Necesidades de entorno**

#### **4.3.9.1. Hardware**

Se requiere un servidor y un computador por usuario.

##### **4.3.9.1.1. Características del servidor**

Servidor Intel core i5, 1.6 GHz, 6GB RAM 1 TB Disco Duro.

##### **4.3.9.1.2. Características del computador**

Computadora Intel core i3, 1.2 GHz, 4 GB RAM 500 GB Disco Duro.

#### **4.3.9.2. Software**

Sistema operativo Ubuntu Server (u otra versión de Linux), también se puede utilizar el sistema operativo Microsoft Windows XP, W7, server 2008.

Servidor web: Apache 2.2.17 o superior.

Gestor de base de datos MySQL 5.0 o superior Requerimientos especiales Intérprete de Ruby 2.2 o superior.

Terminales:

Sistema operativo Windows, Linux, Mac

Navegador Web (Mozilla Firefox, Google Chrome, Internet Explorer, Opera, etc.).

#### **4.3.9.3. Documentos**

El documento indispensable es el Manual del Usuario.

#### **4.3.10. Responsabilidades en la organización y realización de pruebas**

##### **4.3.10.1. Responsable del proyecto**

El responsable del proyecto seleccionó los módulos a ser probados, luego se realizó la administración y soporte técnico de las pruebas en general.

##### **4.3.10.2. Grupo de usuarios del sistema del Hospital Básico 7 B.I Loja.**

Las pruebas fueron aplicadas al personal del Hospital Básico 7 B.I Loja, 5 encuestas en el departamento de fisioterapia y 2 doctores, siendo un total de 7 encuestas (Ver Anexo 1, 2 y 3).

Las pruebas fueron dirigidas con la ayuda del Ing. Juan Pablo Sánchez, encargado del departamento de Sistemas, en primera instancia se realizó algunas explicaciones y una capacitación final del funcionamiento de cada módulo del sistema, indicando cada perfil de usuario y sus respectivas funciones dentro del sistema.

Luego se realizó encuestas para cada perfil de usuario para medir la funcionalidad del sistema. A continuación se detallan cada uno de los procesos evaluados en cada parámetro que permitieron en su totalidad evaluar el sistema.

##### **4.3.11. Pruebas de Funcionalidad:**

Este tipo de pruebas permitió lograr la funcionalidad del sistema, cumpliendo de manera eficaz con los requerimientos solicitados por el usuario (ver Anexo 2). Los parámetros a evaluar:

Existió algún problema al realizar las siguientes actividades:

- Crea una Terapia.
- Suspender una Terapia
- Crear un tratamiento.
- Suspender un Tratamiento
- Habilitar un Tratamiento
- Crea un Horario.
- Suspender un Horario
- Habilitar un Horario

- Crear un Terapista
- Modificar los de datos un Terapista.
- Suspender un Terapista.
- Habilitar un Terapista
- Imprimir Reportes
- Búsqueda Reportes
- Crear número de turno para el día.
- Modificar turnos por día
- Imprimir certificado.
- Modificar Certificado
- Generar reporte de facturas

#### **4.3.12. Pruebas de Usabilidad**

Con este tipo de prueba se puede evaluar la utilidad y robustez del sistema, planteando los siguientes parámetros:

- ¿Cómo considera el diseño del sistema?
- ¿Cómo considera el tiempo empleado para iniciar sesión?
- ¿Cómo calificaría el menú del sistema?
- ¿Cómo califica la información presentada en cada una de las tareas que existen?
- ¿Cómo califica el ingreso de la información que se requiere para el buen uso de datos del paciente o cliente?
- ¿Cómo califica los menús, botones y títulos de la aplicación?
- ¿Considera que los campos de texto de los diferentes formularios son claros como requeridos u obligatorios (\*)?
- ¿Cómo considera la presentación de las hojas de impresión?
- ¿Cómo califica la presentación de los reportes y presentaciones estadísticas?
- ¿Considera que el sistema presenta los menús, de acuerdo al tipo de usuario?
- Dentro del uso del sistema en general. ¿Cómo califica la navegación en todas las páginas, los colores de los menús, los botones e iconos?
- ¿Cómo califica los mensajes de error y aprobación?



#### **4.3.13. Pruebas de Seguridad:**

Para el manejo de la seguridad al ingreso del software, se realiza mediante un nombre de usuario y su respectiva clave, además se toma en cuenta el rol de cada usuario donde el sistema presenta las funciones y privilegios correspondientes, Para la seguridad de las claves se usa el código de encriptación bcrypt.

#### **4.3.14. Pruebas de Carga y Estrés**

Las pruebas de carga y estrés tiene la finalidad de verificar que el sistema no tenga ningún inconveniente al momento de ser utilizado. Para estas pruebas se utilizó la herramienta JMETER la cual es una herramienta libre, en el anexo 6 se encuentra el detalle de la prueba.

#### **4.3.15. Resultados**

Como se mencionó anteriormente se realizaron encuestas (Ver anexo 1, 2 y 3), para los tipos de pruebas ya citadas, realizando un análisis de las mismas, en la siguiente tabla se muestran los procesos evaluados de acuerdo a las preguntas realizadas a los usuarios.

**MB:** Muy Bueno; **B:** Bueno; **R:** Regular; **M:** Malo.

No.	Preguntas	MB	B	R	M	Total
1	¿Cómo considera el diseño del sistema?	5	1	0	0	6
2	¿Cómo considera el tiempo empleado para iniciar sesión?	5	1	0	0	6
3	¿Cómo calificaría el menú del sistema?	4	2	0	0	6
4	¿Cómo califica la información presentada en cada una de las tareas que existen?	5	1	0	0	6
5	¿Cómo califica el ingreso de la información que se requiere para el buen uso de datos del paciente o cliente?	5	1	0	0	6
6	¿Cómo califica los menús, botones y títulos de la aplicación?	6	0	0	0	6
7	¿Considera que los campos de texto de los diferentes formularios son claros como requeridos u obligatorios (*)?	5	1	0	0	6
8	¿Cómo considera la presentación de las hojas de impresión?	6	0	0	0	6
9	¿Cómo califica la presentación de los reportes y presentaciones estadísticas?	4	2	0	0	6
10	¿Considera que el sistema presenta los menús, de acuerdo al tipo de usuario?	6	0	0	0	6
11	Dentro del uso del sistema en general. ¿Cómo califica la navegación en todas las páginas, los colores de los menús, los botones e iconos?	4	2	0	0	6
12	¿Cómo califica los mensajes de error y aprobación?	4	2	0	0	6
<b>TOTAL</b>		59	13	0	0	

Con estos resultados procedemos a calcular los valores en porcentajes con una regla de tres:

1. MUY BUENA =  $59 \times 100 \div 5 = 82 \%$
2. BUENA =  $13 \times 100 \div 5 = 18 \%$
3. REGULAR = 0%
4. MALO = 0 %

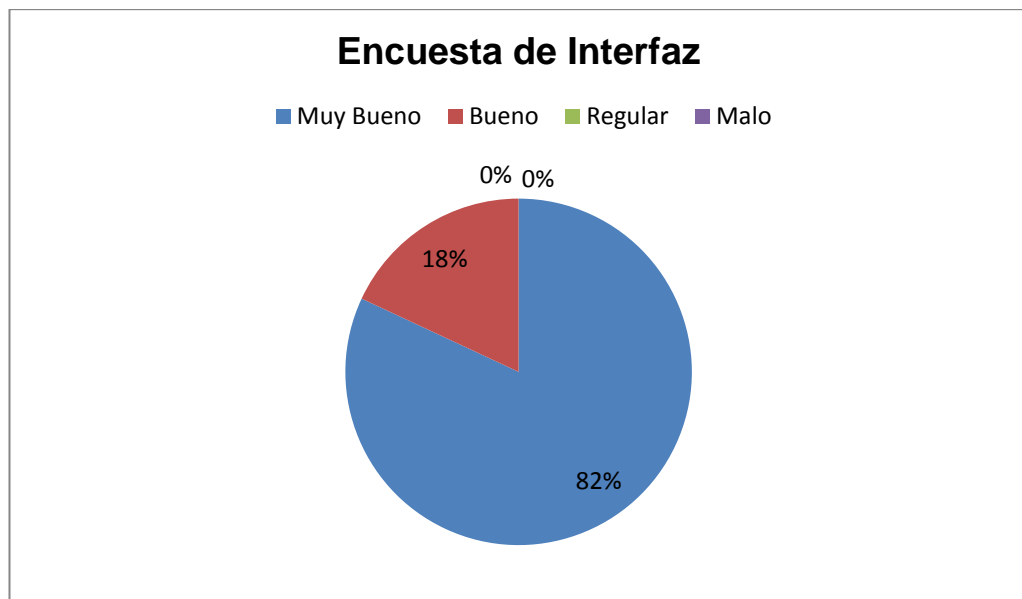


Figura 12. Análisis de encuesta de interfaz

### Análisis de Resultados

En cuanto al diseño de la aplicación el 18% de los usuarios creen que es buena, y un 82% creen que es muy buena, lo que permite calificar al sistema como fácil e intuitivo.

## MÓDULO DE FISIATRÍA

No.	PREGUNTAS	SI	NO	TOTAL
1	Crea una Terapia.	0	1	1
2	Suspender Terapia	0	1	1
3	Crear un tratamiento.	0	1	1
4	Suspender un Tratamiento	0	1	1
5	Habilitar un Tratamiento	0	1	1
6	Crea un Horario.	0	1	1
7	Suspender un Horario	0	1	1
8	Habilitar un Horario	0	1	1
9	Crear un Terapeuta	0	1	1
10	Modificar los de datos un Terapeuta.	0	1	1
11	Suspender un Terapeuta.	0	1	1
12	Habilitar un Terapeuta	0	1	1
12	Imprimir Reportes	0	1	1
14	Búsqueda Reportes	0	1	1
15	Crear número de turno para el día.	0	1	1
16	Modificar turnos por día	0	1	1
17	Imprimir certificado.	0	1	1
18	Modificar Certificado	0	1	1
19	Generar reporte de facturas	0	1	1
20	Enviar paciente a fisioterapia	0	2	2
<b>TOTAL</b>		0	21	21

1. **SI** =  $0 \times 100 \div 21 = 0 \%$
2. **NO** =  $21 \times 100 \div 21 = 100 \%$

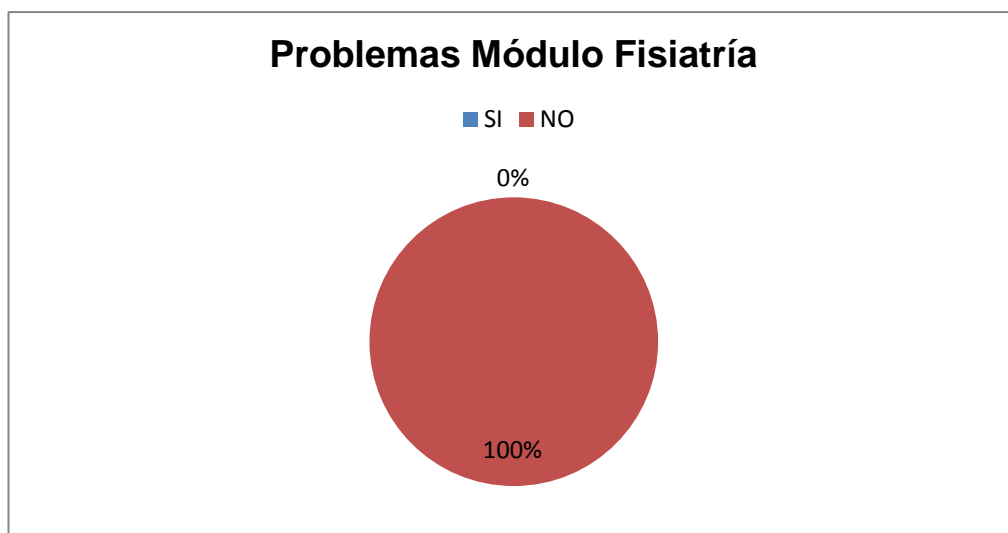


Figura 13. Análisis de problemas del módulo de fisiatría.

**Análisis:** El 100% de los usuarios no tuvieron ningún problema al realizar las actividades.

No.	¿Cómo calificaría la manera de realizar las siguientes actividades?	Fácil	Normal	Difícil	Total
1	Crea una Terapia.	1	1	0	2
2	Suspender Terapia	1	1	0	2
3	Crear un tratamiento.	1	1	0	2
4	Suspender un Tratamiento	1	1	0	2
5	Habilitar un Tratamiento	1	1	0	2
6	Crea un Horario.	1	1	0	2
7	Suspender un Horario	0	2	0	2
8	Habilitar un Horario	2	0	0	2
9	Crear un Terapeuta	2	0	0	2
10	Modificar los de datos un Terapeuta.	2	0	0	2
11	Suspender un Terapeuta.	2	0	0	2
12	Habilitar un Terapeuta	2	0	0	2
13	Imprimir Reportes	0	2	0	2
14	Búsqueda Reportes	2	0	0	2
15	Crear número de turno para el día.	2	0	0	2
16	Modificar turnos por día	2	0	0	2
17	Imprimir certificado.	1	1	0	2
18	Modificar Certificado	1	1	0	2
19	Generar reporte de facturas	2	0	0	2
<b>TOTAL</b>		26	12	0	38

Con los resultados obtenidos procedemos a calcular los valores porcentuales para cada opción de respuesta.

1. **Fácil** =  $26 \times 100 \div 36 = 68 \%$
2. **Normal** =  $12 \times 100 \div 36 = 32 \%$
3. **Difícil** =  $0 \times 100 \div 36 = 0 \%$

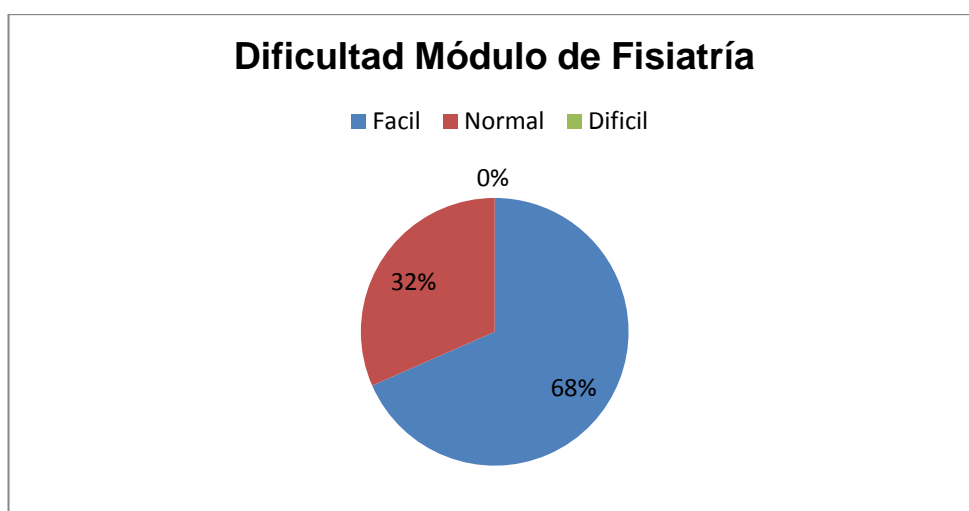


Figura 14. Análisis de dificultad módulo de central de enfermería

**Análisis:** El 61% de los usuarios creen que es fácil de realizar los procesos, y un 39% creen que es normal.

## **g. DISCUSIÓN**

### **1. Desarrollo de la Propuesta Alternativa**

El presente PFC denominado: “Análisis, Diseño e implantación del módulo de Fisiatría al sistema web “SISFA” para el Hospital Básico 7 B.I Loja”, proporcione a la institución la implantación del módulo de fisiatría al sistema web SISFA, el cual integra el departamento de Fisiatría con los módulos de Estadística y Doctores, con la finalidad que la sociedad e institución se beneficien mutuamente, tanto en calidad de servicio y mejor desarrollo de actividades internas.

El objetivo general así como sus objetivos específicos lograron ser abarcados en su totalidad, gracias a la aplicación ordenada de métodos y técnicas orientadas al desarrollo del proyecto así como también de la metodología XP.

A continuación se detalla los objetivos específicos y como se logró llegar a su cumplimiento.

**Objetivo específico 1. Realizar el estudio preliminar en la oficina de Fisiatría para detectar los requerimientos del personal que trabaja aquí como el de los usuarios que utilizan este servicio.**

Para llevar a cabo dicho objetivo se ha hecho uso de las fases de la metodología aplicada, (Ver sección resultados, Fase 1 Planificación del proyecto), cumpliendo con las siguientes actividades.

- Elaboración de especificaciones y funciones a realizar el módulo de fisiatría.(Ver sección: 1.2.2 Funciones del producto)
- Creación de Historias de Usuario (Ver sección 1.3.4 Historias de Usuario).
- Modelado de la base de datos.
- Pruebas de Aceptación. (Ver sección 4. Fases, apartado 4.2).

Una vez realizado estas actividades se procedió a la codificación haciendo uso de las herramientas descritas en el Capítulo III.



## **Objetivo específico 2: Analizar y comprender el funcionamiento de la Oficina de Fisiatría.**

Para llevar a cabo dicho objetivo se ha hecho uso de las fases de la metodología aplicada, (Ver sección resultados, Fase 1 Planificación del proyecto), cumpliendo con las siguientes actividades.

- Elaboración de especificaciones y funciones a realizar el módulo de Estadística. (Ver sección: 1.2.2 Funciones del producto).
- Creación de Historias de Usuario (Ver sección: 1.3.4 Historias de Usuario).

Al realizar el análisis del módulo de fisiatría se desprenden los problemas dentro de esta sección del hospital, los cuales procedemos a codificar haciendo uso de las herramientas descritas en el Capítulo III.

## **Objetivo específico 3: Diseñar el módulo de Fisiatría en base al análisis y determinación de requerimientos de la oficina de Fisiatría.**

Para llevar a cabo dicho objetivo se ha hecho uso de las fases de la metodología aplicada, (Ver sección resultados, Fase 1 Planificación del proyecto), cumpliendo con las siguientes actividades.

- Elaboración de especificaciones y funciones a realizar el módulo de Estadística. (Ver sección: 1.2.2 Funciones del producto).
- Creación del módulo de fisiatría.
- Creación de Historias de Usuario (Ver sección: 1.3.4 Historias de Usuario).
- Pruebas de Aceptación. (Ver sección: 4. Fases, apartado 4.2).

Al desarrollar primero el módulo de fisiatría, se tomó como base los registros que se realizan en los módulos de estadística y doctores, los cuales son los módulos con los cuales se integra el módulo de fisiatría. Esto se lo puede ver en los diagramas de: base de datos, modelo de dominio y de clases.

**Objetivo específico 4: Implementar el módulo de Fisiatría y su integración al sistema “SISFA” para llevarlo a modo producción.**

El cumplimiento de este objetivo se lo puede observar tanto en el modelo de dominio y base de datos, (Ver sección: f. Resultados, apartado 2.4 y 2.6) en el cual se puede apreciar cómo están conectado al resto de módulos del sistema “SISFA”.

La integración de fisiatría con estadística, se la realiza a través de paciente y cliente los cual tienen una relación: “Paciente belongs\_to cliente”, lo cual permite que si un cliente es registrado en: farmacia estará disponible también en estadística o viceversa.

La integración de estadística con administrador de fisiatría, se la realiza en el modelo “HospitalizacionRegistro” y en la base de datos en la tabla “Hospitalización\_registro”, desde doctores envían los pacientes a recibir terapia y luego la información llega a fisiatría.

La aplicación se la llevo a modo producción de manera local en un servidor de la institución con las características mencionadas en el apartado 4.3.9.1.1, con el propósito de realizar las respectivas pruebas y de manera web en los servidores de “Heroku” (Visitar el link: [sisfa.herokuapp.com](http://sisfa.herokuapp.com))

## **2. Valoración Técnica Económica Ambiental**

Se elaboró este tema tomando en cuenta las ventajas que representa tener un sistema integrado con los departamentos del hospital y la necesidad de contar con un software, que permita compartir información, además de gestionar procesos entre departamentos, por lo cual se ha creído conveniente justificar el siguiente proyecto en los siguientes ámbitos:

### **2.1 Académica**

El desarrollo e Implantación de un sistema web de gestión de los servicios de fisioterapia para el Hospital Básico 7 B.I Loja, es de gran importancia por el aporte que dará a la Institución y ciudadanía, el mismo que permitió ganar experiencia al realizar este tema de proyecto de fin de carrera. Además de poner en práctica los conocimientos adquiridos, principalmente de los Análisis, Diseño e Implementación de Sistemas.

### **2.2 Técnica**

La tecnología se ha posicionado fuertemente hoy en día brindando un sin fin de poderosas herramientas de desarrollo de aplicaciones web. Dentro de este contexto el presente proyecto es factible ya que los recursos tecnológicos necesarios para su desarrollo e implantación son accesibles tanto para la institución como desarrolladores, representando una aplicación viable para la institución.

### **2.3 Económica**

Se justifica económicamente porque se cuenta con los recursos necesarios por parte de la institución, cuyo proyecto minimizara el impacto económico permitiendo obtener ahorros significativos al medir y controlar la utilización de recursos humanos y materiales representando un realce positivo no solo a los directivos sino a la ciudadanía en general ya que dará una mejor atención y mayor eficiencia en el desarrollo de sus actividades. A continuación se detalla los recursos humanos, económicos, materiales, técnicos y tecnológicos, con sus costes respectivos.

## 1. Talento Humano

Se muestra un estimado del tiempo y costo asignado al investigador y tutor responsables de la terminación del proyecto.

TABLA LXXI: TIEMPO Y COSTO DEL PROYECTO.

Descripción	Cantidad	Horas	Valor/unit	Valor/Total
Investigador	1	400	\$3.00	\$1200.00
Asesor de Proyecto	1	120	\$5.00	\$600.00
Director de Tesis	1	0	\$0.00	\$0.00

## 2. Bienes

A continuación el estimado del presupuesto de hardware y software y demás materiales que se requieren para la elaboración del presente proyecto.

### Presupuesto Tecnológico.

TABLA LXXII. PRESUPUESTO TECNOLÓGICO.

Cantidad	Equipos Técnicos	Tiempo Utilización (Mes)	Valor/unit	Tiempo de Vida (Año)	Valor/Total
1	Computador HP Pavillon Dv4	12	\$ 1100	5	\$ 220.00
1	Impresora Canon Mp190	12	\$ 80	3	\$ 27
1	Servidor HP Profile	1	\$ 600.00		\$ 600.00
<b>Software</b>					
1	Ubuntu server 12.04 LTS	1	\$ 0.00		\$ 0.00
1	Ruby on Rails	1	\$ 0.00		\$ 0.00
1	Mysql	1	\$ 0.00		\$ 0.00

## Presupuesto Materiales

TABLA LXXIII: PRESUPUESTO DE MATERIALES

	<b>Materiales</b>	<b>Valor Unitario</b>	<b>Valor Total</b>
1000	Hojas A4	\$ 0.015	\$15.00
3	Anillado	\$ 2.00	\$ 6.00
8	Materiales de oficina	\$ 1.00	\$ 8.00
4	Cartuchos de impresora Canon	\$ 22.00	\$88.00
3	Borde o Perfil	\$ 0.50	\$ 1.50
5	CD's	\$ 0.80	\$ 4.00

## 3. Servicios

Son los servicio que se utilizara a lo largo del desarrollo, indicando precios reales que implican el uso de dichos recursos. Cabe mencionar que la capacitación virtual son videos tutoriales pagados, que se desarrollan en línea.

TABLA LXXIV: PRESUPUESTO SERVICIOS

<b>Servicio</b>	<b>Descripción</b>	<b>Valor/unit</b>	<b>Valor/Total</b>
Transporte	250	\$ 0.30	\$ 75.00
Internet	12 meses	\$ 20.00	\$ 240.00
Capacitación Virtual	Video Tutoriales Lynda.com	\$ 500.00	\$ 500.00

## 4. Resumen del Presupuesto

Se presenta el presupuesto total de las tablas ya descritas anteriormente y a su vez un estimado de un 4% de imprevistos del total de gastos.

TABLA LXXV: RESUMEN DE PRESUPUESTO

<b>Resumen del Presupuesto</b>	<b>Costo Total</b>
<b>Talento Humano</b>	\$ 1800.00
<b>Bienes</b>	\$ 969.50
<b>Servicios</b>	\$ 815.00
<b>SUBTOTAL</b>	\$ 3584.50
<b>Imprevistos 4 %</b>	\$ 143.38
<b>TOTAL</b>	<b>\$3727,88</b>

#### **2.4 Valoración Ambiental**

Es factible porque disminuirá el consumo de papel de oficina, al registrar formularios y reportes digitales, los equipos utilizados para la implementación como el servidor se encuentra en la nube lo cual no representa ningún daño ambiental dentro de la institución.

## **h. Conclusiones**

Se ha llegado a las siguientes conclusiones luego de haber realizado el PFC y cumpliendo con los objetivos planteados al inicio del proyecto.

- El uso de la metodología ágil XP (Programación Extrema), permitió desarrollar el sistema en base a las necesidades reales del usuario, además permitió una correcta integración con sistema web "SISFA" actualmente implementado en el Hospital Básico 7BI Loja.
- Las pruebas realizadas al sistema permitieron detectar y reparar errores que se presentaron durante la etapa de desarrollo, mejorando la seguridad, tiempos de respuesta cumpliendo con las expectativas previstas.
- La integración del módulo de fisiatría al Sistema Web "SISFA", permite obtener control y agilidad entre procesos que se lleva en cada Departamento.
- La utilización de formularios digitales, establecidos por el Hospital y Ministerio de Salud del Ecuador, permitió mejorar el cumplimiento de las funciones del personal de Fisiatría y de los médicos que trabajan en el hospital.
- El cierre automático de sesión de usuario inactivo cada 10 min, permite evitar el robo de información y manejo de la misma, convirtiéndola en una técnica de seguridad del sistema.

## **i. Recomendaciones**

- Manejar la misma metodología de desarrollo, para la creación de nuevos módulos para evitar problemas en la obtención de requerimientos y codificación del sistema.
- En caso de aumentar el volumen de datos y usuarios, realizar las pruebas de carga y estrés con actores reales para mejorar el rendimiento del software.
- Elegir un framework de desarrollo de tipo MVC, el cual permita mantener la misma línea de desarrollo, asegurando la integración entre componentes, lenguajes y herramientas.
- Cada usuario debe mantener la confiabilidad de su clave y nombre de usuario, para que no existan inconvenientes en el registro de procesos y emisión de reportes.
- Integrar a otras dependencias del Hospital, por medio de nuevos módulos para la automatización de procesos y manejo de un solo sistema



## **j. Bibliografía**

[1] Ruby on Rails [En línea]: <<https://www.ruby-lang.org/es/about/>> [Consultado el 16 de Agosto de 2013].

[2] Wilson Andrés Guachun G, Súper productividad en el desarrollo de aplicaciones Web usando el lenguaje Ruby, Universidad Politécnica Salesiana del Ecuador [En línea]: <<http://dspace.ups.edu.ec/bitstream/123456789/518/5/CAPITULO3.pdf>> [Consultado el 22 de Agosto de 2013].

[3] REALPE ROSERO; Cristian Fernando. Implementación del portal web 2.0, para la mancomunidad de la cuenca del rio mira. Universidad Técnica del Norte [En línea]: <<http://repositorio.utn.edu.ec/bitstream/123456789/1075/1/04%20ISC%20228-TEISIS.pdf>> [Consultado en 11 de Septiembre de 2013].

[4] Arquitectura de una aplicación desarrollada con RoR. [En línea]: <[http://www.windaroooci.com/assets/service/main\\_image\\_url/1/rails\\_architecture.png](http://www.windaroooci.com/assets/service/main_image_url/1/rails_architecture.png)> [Consultado el 12 de Septiembre de 2013].

[5] Componentes Rails MODELO VISTA CONTROLADOR (MVC). [En línea]: <[http://ftp.gul.uc3m.es/pub/gul/cursos/2005/noviembre/ruby\\_on\\_rails\\_ajax\\_web2.0\\_gul\\_uc3m/imagenes/mvc.png](http://ftp.gul.uc3m.es/pub/gul/cursos/2005/noviembre/ruby_on_rails_ajax_web2.0_gul_uc3m/imagenes/mvc.png)> [Consultado el 12 de Septiembre de 2013].

[6] Cristian Fernando Realpe Rosero, Análisis y Estudio de Tecnología Ruby on Rails con bases de datos Postgres para Aplicaciones Web 2.0, Universidad Técnica del Norte [En línea]: <<http://repositorio.utn.edu.ec/bitstream/123456789/1075/1/04%20ISC%20228-TEISIS.pdf>>, [Consultado el 12 de Septiembre de 2013].

[7] FERNÁNDEZ Gerardo. Introducción a Extreme Programming, <http://aalbertovargasc.files.wordpress.com/2011/07/presentacion-xp.pdf>, [Consultado 10 de febrero del 2014].

- [8] José H. Canós, Patricio Letelier y M<sup>a</sup> Carmen Penadés, Metodologías Ágiles en el Desarrollo de Software, [http://noqualityinside.com.ar/nqi/nqifiles/XP\\_Agil.pdf](http://noqualityinside.com.ar/nqi/nqifiles/XP_Agil.pdf), [Consultado 3 Marzo del 2014].
- [9] Echeverry Tobón, Luís Miguel, Delgado Carmona, Luz Elena, Caso práctico de la metodología ágil XP al desarrollo de software, <http://recursosbiblioteca.utp.edu.co/dspace/handle/11059/794>, [Consultado 3 Marzo del 2014].
- [10] Universidad del Valle. Cali. Colombia. Programación Extrema. <http://eisc.univalle.edu.co/materias/WWW/material/lecturas/xp.pdf>, [Consultado 4 Julio del 2014]
- [11] Flanagan, David, JavaScript: The Definitive Guide (4<sup>a</sup> Edición edición) (2002).
- [12] Miguel Ángel Álvarez, Introducción a JQuery <http://www.desarrolloweb.com/articulos/introduccion-jquery.html> [Consultado 4 Julio 2014]
- [13] J Ashkenas, The Little Book on CoffeeScript [http://jameslavin.com/Little\\_Book\\_on\\_CoffeeScript.pdf](http://jameslavin.com/Little_Book_on_CoffeeScript.pdf), [Consultado Agosto 2014]
- [14] Javier Eguiluz, Introducción a CSS, <http://librosweb.es/css/>, [Consultado 19 de Agosto 2014].
- [15] Mark Otto, Jacob Thornton, Bootstrap 3 el manual oficial [http://librosweb.es/bootstrap\\_3](http://librosweb.es/bootstrap_3), [Consultado 19 de Agosto 2014].

## **k. Anexos**

## Anexo 1: Encuesta realizada al sobre la interfaz del sistema.

### UNIVERSIDAD NACIONAL DE LOJA

Área de la Energía, las industrias y los Recursos Naturales No Renovables

#### CARRERA DE INGENIERÍA EN SISTEMAS

Con motivo de realizar el tema de tesis “ANÁLISIS, DISEÑO E IMPLANTACIÓN DEL MÓDULO DE FISIATRÍA AL SISTEMA WEB “SISFA” PARA EL HOSPITAL BÁSICO 7 B.I LOJA”, en esta prestigiosa institución me permito saludar a cada uno de ustedes señores encargados del departamento de estadística, con el fin de solicitar muy comedidamente se digne en dar respuesta a las siguientes preguntas.

Nombre: \_\_\_\_\_

Fecha: \_\_\_\_\_ Cargo: \_\_\_\_\_

**MB:** Muy Bueno; **B:** Bueno; **R:** Regular; **M:** Malo.

Preguntas	MB	B	R	M
¿Cómo considera el diseño del sistema?				
¿Cómo considera el tiempo empleado para iniciar sesión?				
¿Cómo calificaría el menú del sistema?				
¿Cómo califica la información presentada en cada una de las tareas que existen?				
¿Cómo califica el ingreso de la información que se requiere para el buen uso de datos del paciente o cliente?				
¿Considera que los formularios de registro son claros?				
¿Cómo califica los menús, botones y títulos de la aplicación?				

¿Considera que los campos de texto de los diferentes formularios son claros como requeridos u obligatorios (*)?				
¿Cómo considera la presentación de las hojas de impresión?				
¿Cómo considera las tablas de consultas?				
¿Considera que las tablas donde están organizados los diferentes pacientes, tratamientos, terapistas, etc., brindan la búsqueda por criterios (Nombre, terapeuta, fecha de atención)?				
¿Cómo califica la presentación de las tablas, reportes y presentaciones estadísticas?				
¿Considera que el sistema presenta los menús, de acuerdo al tipo de usuario?				
Dentro del uso del sistema en general. ¿Cómo califica la navegación en todas las páginas, los colores de los menús, los botones e iconos?				
¿Cómo califica los mensajes de error y aprobación?				

---

Firma

## Anexo 2: Encuesta realizada sobre problemas en procesos del sistema.

### UNIVERSIDAD NACIONAL DE LOJA

Área de la Energía, las industrias y los Recursos Naturales No Renovables

#### CARRERA DE INGENIERÍA EN SISTEMAS

Con motivo de realizar el tema de tesis “ANÁLISIS, DISEÑO E IMPLANTACIÓN DEL MÓDULO DE FISIATRÍA AL SISTEMA WEB “SISFA” PARA EL HOSPITAL BÁSICO 7 B.I LOJA”, en esta prestigiosa institución me permito saludar a cada uno de ustedes señores encargados del departamento de estadística, con el fin de solicitar muy comedidamente se digne en dar respuesta a las siguientes preguntas.

Nombre: \_\_\_\_\_

Fecha: \_\_\_\_\_ Cargo: \_\_\_\_\_

Existió algún problema al realizar las siguientes actividades:

PREGUNTAS	SI	NO
1. Crea una Terapia.		
2. Suspende una Terapia		
3. Crear un tratamiento.		
4. Suspende un Tratamiento		
5. Habilitar un Tratamiento		
6. Crea un Horario.		
7. Suspende un Horario		
8. Habilitar un Horario		

9. Crear un Terapista		
10. Modificar los de datos un Terapista.		
11. Suspender un Terapista.		
12. Habilitar un Terapista		
13. Imprimir Reportes		
14. Búsqueda Reportes		
15. Crear número de turno para el día.		
16. Modificar turnos por día		
17. Imprimir certificado.		
18. Modificar Certificado		
19. Generar reporte de facturas		

---

Firma

### Anexo 3: Encuesta realizada sobre dificultad en procesos del sistema.

## UNIVERSIDAD NACIONAL DE LOJA

Área de la Energía, las industrias y los Recursos Naturales No Renovables

### CARRERA DE INGENIERÍA EN SISTEMAS

Con motivo de realizar el tema de tesis “ANÁLISIS, DISEÑO E IMPLANTACIÓN DEL MÓDULO DE FISIATRÍA AL SISTEMA WEB “SISFA” PARA EL HOSPITAL BÁSICO 7 B.I LOJA”, en esta prestigiosa institución me permito saludar a cada uno de ustedes señores encargados del departamento de estadística, con el fin de solicitar muy comedidamente se digne en dar respuesta a las siguientes preguntas.

Nombre: \_\_\_\_\_

Fecha: \_\_\_\_\_ Cargo: \_\_\_\_\_

¿Cómo calificaría la manera de realizar las siguientes actividades?

PREGUNTAS	MB	B	R	M
1. Crea una Terapia.				
2. Finalizar una Terapia				
3. Crear un tratamiento.				
4. Suspender un Tratamiento				
5. Habilitar un Tratamiento				
6. Crea un Horario.				
7. Suspender un Horario				
8. Habilitar un Horario				
9. Crear un Terapeuta				

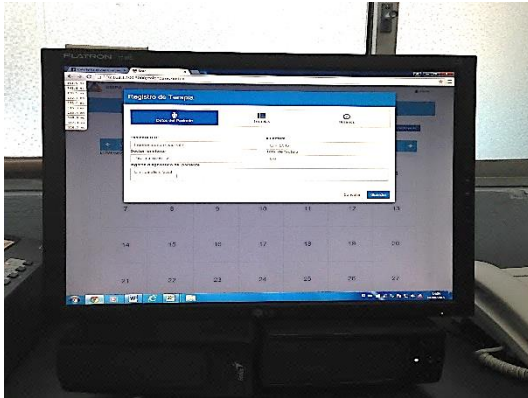


10. Modificar los de datos un Terapista.				
11. Suspende un Terapista.				
12. Habilitar un Terapista				
13. Imprimir Reportes				
14. Búsqueda Reportes				
15. Crear número de turno para el día.				
16. Modificar turnos por día				
17. Imprimir certificado.				
18. Modificar Certificado				
19. Generar reporte de facturas				

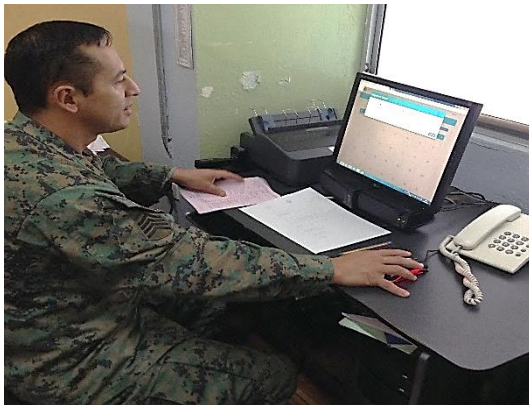
---

Firma

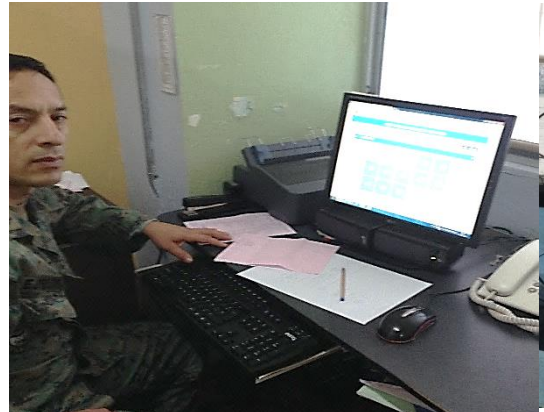
## Anexo 4: Fotografías del personal del hospital utilizando el sistema implantando.



Departamento de Fisiatría



Administrador de Fisiatría



Auxiliar de Fisiatría

## Anexo 5: Validación HTML.

validator.w3.org

Esta página se ha traducido del inglés al español. Mostrar original Configuración

**W3C** Servicio de Validación de marcado  
Compruebe el marcado (HTML, XHTML, ...) de los documentos Web

Validar por URI Validar por carga de archivos Validar mediante entrada directa

Validar por URI  
Validar un documento en línea:

Dirección:

► Más opciones

Comprobar

**W3C** Servicio de Validación de marcado  
Compruebe el marcado (HTML, XHTML, ...) de los documentos Web

Saltar a: [Notas y Problemas Potenciales](#) [Enhorabuena · Iconos](#)

**Este documento se comprobó con éxito como HTML5!**

Resultado:	Pasado, 1 de alerta (s)	
Dirección :	<input type="text" value="http://sisfa.herokuapp.com/login"/>	
Codificación :	UTF-8	<input type="text" value="(Detectar automáticamente)"/>
Doctype :	HTML5	<input type="text" value="(Detectar automáticamente)"/>
Raíz Elemento:	html	

## **Anexo 6: Pruebas de carga y estrés con JMETER.**

Las pruebas de carga, rendimiento y estrés fueron realizadas por el desarrollador del proyecto, permitiendo verificar el correcto funcionamiento de la aplicación cuando existen accesos simultáneos de usuarios.

Para este tipo de pruebas se ha utilizado la herramienta Apache JMeter™, ya que fue la herramienta seleccionada en la creación de los primeros módulos.

Estas pruebas se las ha realizado contra el servidor de Heroku, de características básicas:

- 1X CPU Share
- RAM: 512MB
- Swap space: 1GB max
- Storage space: 100MB max

### **1.1 Prueba de Carga**

Las pruebas de carga tienen la finalidad de medir la capacidad máxima que tiene un servidor web (hardware y software), simulando varios usuarios que realizan una petición a la aplicación al mismo tiempo.

En la Figura 1 se muestra las pruebas realizadas al módulo de fisioterapia con 6 usuarios, número actual de personal en este departamento. Se realiza la misma petición a la vez, con lo que nos devuelve un margen de error del 0%.

Label	# Muestras	Media	Mediana	Linea de 90%	Mín	Máx	% Error
Pagina de Inicio	6	1117	1174	1474	692	1474	0,00%
Inicio de sesion	6	2263	2156	2592	2065	2592	0,00%
Registrar terapias de paciente	6	1004	1010	1066	940	1066	0,00%
Eliminar registro de terapias	6	993	1002	1034	932	1034	0,00%
Registro de tratamiento recibido	6	951	970	1093	875	1093	0,00%
Crear Terapeuta	6	1104	1148	1211	959	1211	0,00%
Suspender / Activar Terapeuta	6	1170	1362	1469	725	1469	0,00%
Crear Horario	6	1262	1381	1744	989	1744	0,00%
Activar/Suspender horario	6	1576	1464	2536	1158	2536	0,00%
Crear Terapia	6	1184	1221	1474	933	1474	0,00%
Activar/Suspender Tratamiento	6	1300	1023	2612	996	2612	0,00%
Agregar configuraciones	6	1029	1050	1153	850	1153	0,00%
Reporte mensual estadística	6	1201	1235	1402	1038	1402	0,00%
Reporte por Factura	6	1187	1159	1432	1078	1432	0,00%
Reporte por terapeuta	6	1132	1175	1399	853	1399	0,00%
Reporte por Paciente	6	1484	1261	2476	1205	2476	0,00%
Logout	6	1025	970	1494	741	1494	0,00%
TOTAL	102	1234	1116	1679	692	2612	0,00%

Fig. 1 Resultado de la prueba de carga del módulo de fisioterapia.

Se estima un crecimiento de 12 usuarios más en los próximos 3 años, esto en base a las nuevas instalaciones de la institución, por lo cual se realiza las pruebas con 12 usuarios al mismo tiempo, los resultados se aprecian en la Fig 2.

Label	# Muestras	Media	Mediana	Linea de 90%	Mín	Máx	% Error
Pagina de Inicio	10	1053	1339	1567	522	1567	0,00%
Inicio de sesion	10	2542	2547	3219	1876	3219	0,00%
Registrar terapias de paciente	10	1464	1588	1690	1146	1690	0,00%
Eliminar registro de terapias	10	1225	1308	1537	881	1537	0,00%
Registro de tratamiento recibido	10	1733	1368	4185	1061	4185	0,00%
Crear Terapeuta	10	1446	1427	2246	1014	2246	0,00%
Suspender / Activar Terapeuta	10	1565	1406	2164	1236	2164	0,00%
Crear Horario	10	1368	1374	1579	1108	1579	0,00%
Activar/Suspender horario	10	1157	1204	1389	910	1389	0,00%
Crear Terapia	10	1111	1140	1372	900	1372	0,00%
Activar/Suspender Tratamiento	10	1065	1007	1616	854	1616	0,00%
Agregar configuraciones	10	1094	1047	1744	866	1744	0,00%
Reporte mensual estadística	10	1498	1387	2340	1051	2340	0,00%
Reporte por Factura	10	1256	1191	2003	876	2003	0,00%
Reporte por terapeuta	10	1305	1016	3263	676	3263	0,00%
Reporte por Paciente	10	1273	1105	1934	913	1934	0,00%
Logout	10	1208	1154	1972	628	1972	0,00%
TOTAL	170	1374	1236	1981	522	4185	0,00%

Fig. 2 Resultado de la prueba de carga del módulo de fisioterapia con proyección a 10 usuarios.

## 1.2 Prueba de Rendimiento

Las pruebas de rendimiento son utilizadas para medir el tiempo de respuesta de una cierta tarea.

En la Fig. 3 Se visualiza el tiempo de respuesta en milisegundos que demoran en responder cada una de las operaciones, en un servidor de características básicas descritas al principio de este informe, el rendimiento mínimo es 825ms, el rendimiento

máximo es de 2896ms debido a la cantidad de usuarios interactuando al mismo tiempo y el rendimiento promedio de estas actividades es de 1323ms.

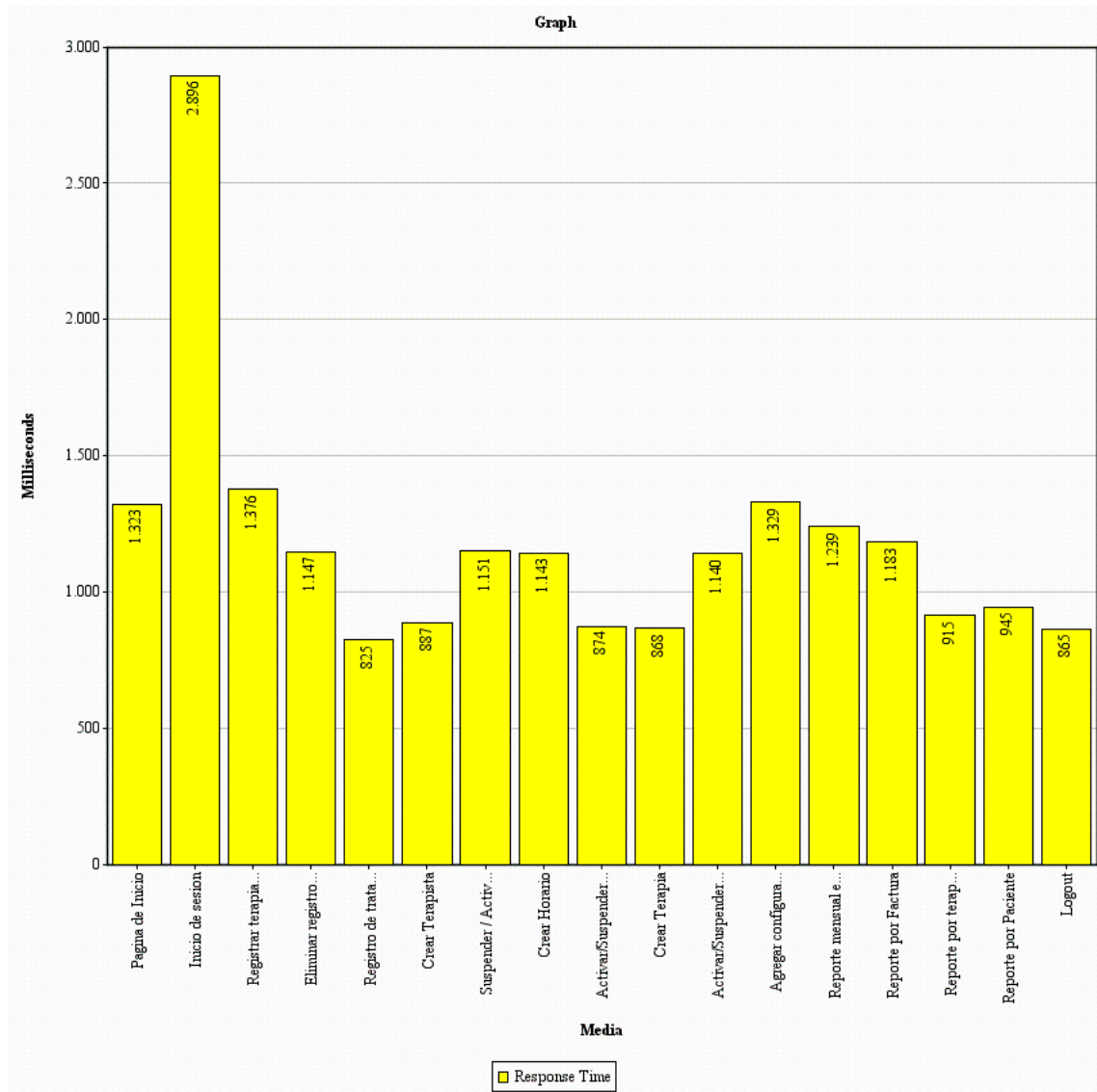


Fig. 3 Resultado de la prueba de rendimiento del módulo de fisioterapia.

### 1.3 Prueba de estrés

Las pruebas de estrés son utilizadas para determinar cuándo una aplicación es incapaz de responder a las peticiones, en la Fig. 4 se muestra que si realizan 180 peticiones por segundo, el sistema no es capaz de responder a todas las solicitudes, dando un margen de error del 56,11% en la parte de Inicio de sesión.

Label	# Muestras	Media	Mediana	Linea de 90%	Mín	Máx	% Error
Pagina de Inicio	180	5381	4986	9209	414	14777	0,00%
Inicio de sesion	180	39260	42997	45872	30187	49049	56,11%
Registro terapias de paciente	79	6150	5144	10040	2553	20509	0,00%
Eliminar registro de terapias	79	3922	3607	4957	2977	12986	0,00%
Registro tratamiento recibido	79	3978	3277	7430	2884	8111	0,00%
Crear terapeuta	79	4839	4613	5784	3276	9345	0,00%
Suspender / Activar Terapeuta	79	3471	3165	4442	2613	7057	0,00%
Crear Horario	79	3996	3708	5281	2636	8380	0,00%
Activar/Suspender Horario	79	4283	4133	5785	2589	10042	0,00%
Crear Terapia	79	3584	3351	4451	2562	8868	0,00%
Activar/Suspender tratamiento	79	3942	3620	4659	3152	7688	0,00%
Agregar configuraciones	79	5878	5628	8201	4367	9952	0,00%
Reporte mensual estadística	79	5496	5290	5705	4592	13099	0,00%
Reporte por factura	79	5438	5354	5973	2180	8809	0,00%
Reporte por terapeuta	79	4184	3514	8153	435	10291	0,00%
Reporte por Pacientes	79	3733	3461	6463	443	9113	0,00%
Logout	79	2542	2854	3784	462	5043	0,00%
TOTAL	1545	8547	4452	30218	414	49049	6,54%

Fig. 14 Resultado de la prueba de estrés del módulo de fisiatría.

## **Anexo 7: Certificación de traducción del resumen.**



**Anexo 8: Certificación de haber implantado el sistema en el Hospital Básico 7bi Loja.**

## Anexo 9: Licencias.

Licencia Creative Commons

Atribución-No Comercial-Compartir igual CC BY-NC-SA

Esta licencia permite a otros distribuir, re mezclar, retocar, y crear a partir de tu obra de modo no comercial, siempre y cuando den crédito y licencien sus nuevas creaciones bajo las mismas condiciones.



**Atribución** El beneficiario de la licencia tiene el derecho de copiar, distribuir, exhibir y representar la obra y hacer obras derivadas siempre y cuando reconozca y cite la obra de la forma especificada por el autor o el licenciante.



**No Comercial** El beneficiario de la licencia tiene el derecho de copiar, distribuir, exhibir y representar la obra y hacer obras derivadas para fines no comerciales.



**No Derivadas** El beneficiario de la licencia solamente tiene el derecho de copiar, distribuir, exhibir y representar copias literales de la obra y no tiene el derecho de producir obras derivadas.



**Compartir Igual** El beneficiario de la licencia tiene el derecho de distribuir obras derivadas bajo una licencia idéntica a la licencia que regula la obra original.

El presente trabajo se encuentra bajo la licencia Creative Commons

**Desarrollo del proyecto:**

“Análisis, Diseño e Implantación del módulo de Fisiatría al sistema web “SISFA” para el Hospital Básico 7 B.I Loja”

**Licencia:**



Análisis, Diseño e implantación del módulo de Fisiatría al sistema web “SISFA” para el Hospital Básico 7 B.I Loja por Leonardo Xavier Alvarado Motoche se distribuye bajo una [Licencia Creative Commons Atribución-NoComercial-CompartirIgual 4.0 Internacional](#)

**Licencia de Código:**

El código desarrollador en este sistema se encuentra bajo una licencia: **Apache**

La Licencia Apache no exige que las obras derivadas (versiones modificadas) del software se distribuyan usando la misma licencia, ni siquiera que se tengan que distribuir como software libre/open source. La Licencia Apache sólo exige que se mantenga una noticia que informe a los receptores que en la distribución se ha usado código con la Licencia Apache. Así, en contraste a las licencias copyleft, quienes reciben versiones modificadas de código con Licencia Apache no reciben necesariamente las mismas libertades. O, si se considera la situación desde el punto de vista de los licenciarios de código con Licencia Apache, reciben la libertad de usar el código de la forma que prefieran, incluyendo su uso en productos de código cerrado.