

**UNIVERSIDAD POLITÉCNICA SALESIANA  
SEDE QUITO**

**CARRERA: INGENIERÍA DE SISTEMAS**

**Trabajo de titulación previo a la obtención del título de: INGENIERO DE  
SISTEMAS**

**TEMA:  
DESARROLLO DE UNA APLICACIÓN WEB DE ADMINISTRACIÓN DE  
CLIENTES Y DESARROLLO DE UN PLAN DE EJERCICIOS DE  
ENTRENAMIENTO FÍSICO PERSONALIZADO, Y MÓDULO DE  
FACTURACIÓN DEL GIMNASIO “ABSOLUT GYM”**

**AUTOR:  
EDGAR EDUARDO CONRADO DÍAZ**

**DIRECTOR:  
XAVIER ALEXANDER CALDERÓN HINOJOSA**

**Quito, mayo de 2015**

**DECLARATORIA DE RESPONSABILIDAD Y AUTORIZACIÓN DE USO  
DEL TRABAJO DE TITULACIÓN**

Yo, autorizo a la Universidad Politécnica Salesiana la publicación total o parcial de este trabajo de titulación y su reproducción sin fines de lucro.

Además, declaro que los conceptos, análisis desarrollados y las conclusiones del presente trabajo son de exclusiva responsabilidad del autor.

Quito, mayo 2015.

---

Edgar Eduardo Conrado Díaz  
C.C. 171184852-1

## DEDICATORIA

Este esfuerzo lo dedico a mi madre lo más hermoso que me ha regalado Dios, a las familias de mis hermanos David y Belén, mis tíos y primos, y en especial a mi querida esposa Katherine Vega y mi precioso hijo Aaron Ezequiel, que nunca se acabe el amor, comprensión y respeto.

Edgar Eduardo Conrado Díaz

*“Muchos de los fracasos de la vida vienen de las personas que no se dieron cuenta de lo cerca que estaban del éxito, cuando se dieron por vencidos”*

*Thomas A. Edison*

## **AGRADECIMIENTO**

A mis maestros de la Universidad Politécnica Salesiana, quienes me inculcaron sus valores morales y conocimientos, ¡muchas gracias!

Edgar Eduardo Conrado Díaz

## ÍNDICE

<b>INTRODUCCIÓN .....</b>	<b>1</b>
<b>CAPÍTULO 1.....</b>	<b>2</b>
<b>APLICACIONES WEB.....</b>	<b>2</b>
1.1.1 El navegador web - browser .....	3
1.1.2 HTML dinámico .....	4
1.1.3 Redes informáticas .....	5
1.2 Software libre .....	5
1.2.1 Software propietario .....	6
1.3 Plataforma de desarrollo .....	6
1.3.1 Evolución de la tecnología .....	6
1.3.2 Tecnología JSP .....	6
1.4 Netbeans .....	8
1.4.1 Java EE .....	9
1.4.1.1 Historia.....	9
1.4.1.2 Plataforma Java EE.....	10
1.4.1.2.1 Arquitectura en Java EE.....	11
1.4.1.2.2 Arquitectura en Java SE.....	11
1.4.1.2.3 APIs generales.....	12
1.4.1.2.3.1 javax.ejb.* .....	12
1.4.1.2.3.2 javax.naming .....	12
1.4.1.2.3.3 java.sql.....	12
1.4.1.2.3.4 java.transaction.* .....	12
1.4.1.2.3.5 javax.xml.* .....	12
1.4.1.2.3.6 javax.jms.* .....	12
1.4.1.2.3.7 javax.persistence.....	13
1.4.1.3 Desarrollo de bajo coste – plataforma .....	13
1.4.1.4 Servidores de aplicaciones Java EE 5 certificados .....	14
1.4.1.5 Servidores de aplicaciones J2EE 1.4 certificados.....	14
1.4.1.6 Servidores de aplicaciones J2EE 1.3 certificados.....	15
1.5 Base de datos.....	15
1.5.1 Sistema de administración de bases de datos .....	15
1.5.2 Propiedades fundamentales .....	15

1.5.3	Características de los DBMS .....	16
1.5.4	Propiedades ACID.....	16
1.5.5	Usuarios.....	16
1.5.6	Componentes funcionales de un DBMS .....	17
1.6	MySQL.....	17
1.7	Metodología de desarrollo: XP eXtremeProgramming.....	18
1.7.1	Reglas y prácticas .....	18
1.7.1.1	Etapa de planificación.....	18
1.7.1.2	Etapa de diseño .....	19
1.7.1.3	Etapa de desarrollo del código.....	19
1.7.2	Filosofía diseño centrado en el usuario (DCU) .....	19
1.7.2.1	Propuesta de adaptación metodológica DCU .....	19
1.7.3	Estudio y modelado del usuario .....	21
1.7.4	Diseño y prototipado .....	22
1.7.5	Evaluación de la accesibilidad.....	23
1.7.6	Test con usuarios .....	24
1.8	Normativas de pruebas de sistemas de software .....	24
1.8.1	Pruebas de carga.....	25
1.8.2	Prueba de estrés .....	26
1.8.3	Prueba de estabilidad (Soak Testing) .....	26
1.8.4	Pruebas de picos (Spike Testing) .....	26
<b>CAPÍTULO 2.....</b>		<b>27</b>
<b>ANÁLISIS DE REQUERIMIENTOS.....</b>		<b>27</b>
2.1	Información del gimnasio .....	27
2.2	Misión del gimnasio.....	27
2.3	Visión del gimnasio .....	27
2.4	Valores .....	27
2.5	Organigrama gimnasio .....	28
2.6	Infraestructura del gimnasio.....	28
2.7	Procesos del gimnasio .....	28
2.8	Análisis de la información .....	29
2.8.1	Documento utilizado en el proceso de valoración física.....	29
2.8.1.1	Formulario para valoración física.....	29
2.8.1.1.1	Objetivo.....	29

2.8.1.1.2	Motivo de valoración .....	29
2.8.1.1.3	Antecedentes familiares .....	30
2.8.1.1.4	Enfermedad o problema actual.....	30
2.8.1.1.5	Signos vitales y antropometría .....	30
2.8.1.1.6	Área físico regional .....	31
2.8.1.1.7	Diagnóstico .....	32
2.8.1.1.8	Planes de tratamiento .....	32
2.8.1.1.9	Planes de entrenamiento.....	32
2.8.1.1.10	Evolución del estado físico .....	32
2.8.1.1.11	Evolución .....	32
2.9	Detalle del funcionamiento de los procesos en la aplicación web .....	33
2.9.1	Proceso de asignación de valoraciones físicas y médicas .....	33
2.9.2	Cobro de servicios .....	34
2.9.3	Reportes.....	34
<b>CAPÍTULO 3.....</b>		<b>35</b>
<b>DISEÑO DEL SOFTWARE .....</b>		<b>35</b>
3.1	Diseño de la interface basado en el usuario .....	35
3.1.1	Identificación de usuarios.....	36
3.1.1.1	Descripción de las tareas de los usuarios.....	37
3.1.1.1.1	Administrador .....	37
3.1.1.1.2	Entrenador físico .....	37
3.1.1.1.3	Médico.....	37
3.1.1.1.4	Personal administrativo .....	37
3.1.1.1.5	Recepcionista .....	37
3.1.2	Análisis de las necesidades de los usuarios .....	38
3.1.3	Diseño y evaluación de las interfaces.....	38
3.1.3.1	Interface acceso al sitio web .....	38
3.1.3.2	Interface crear usuario .....	39
3.1.3.3	Interface ver/editar/borrar usuario .....	39
3.1.3.4	Interface crear cliente.....	40
3.1.3.5	Interface antecedentes personales .....	40
3.1.3.6	Interface tipo entrenamiento .....	41
3.1.3.7	Interface pantalla principal .....	41
3.1.3.8	Estados de la cita.....	42

3.1.3.9	Interface valoración física.....	42
3.1.3.10	Interface signos vitales.....	43
3.1.3.11	Interface generar cobro .....	43
3.1.3.12	Interface generar factura .....	44
3.1.3.13	Interface ingresar valoración física.....	44
3.1.3.14	Interface área físico regional.....	45
3.1.3.15	Interface diagnóstico .....	45
3.1.3.16	Interface plan de tratamiento .....	46
3.1.3.17	Interface plan de entrenamiento.....	46
3.1.3.18	Interface reportes del sistema web.....	46
3.1.4	Evaluación del proceso.....	47
3.2	Modelamientos.....	47
3.2.1	Modelos aplicados a cada proceso de aplicación web .....	47
3.2.1.1	Diagrama de clases .....	48
3.2.1.2	Diagramas casos de uso .....	49
3.2.1.3	Diagramas de estado .....	54
3.2.1.4	Diagramas de secuencia.....	60
3.2.1.5	Diagrama conceptual .....	69
<b>CAPÍTULO 4.....</b>		<b>71</b>
<b>PRUEBAS DEL SISTEMA INFORMÁTICO E IMPLEMENTACIÓN.....</b>		<b>71</b>
4.1	Pruebas de sistema de software.....	71
4.1.1	Flujo trabajo de la prueba.....	71
4.1.2	Tipo de prueba adecuada al nivel de sistema .....	71
4.1.3	Rendimiento .....	72
4.1.4	Características de la prueba .....	73
4.1.4.1	Herramienta Apache JMeter .....	73
4.1.4.2	Realización de la prueba .....	74
4.1.4.3	Prueba .....	76
4.1.4.4	Gráficos de los resultados .....	78
4.1.4.5	Resultados obtenidos .....	79
4.1.4.6	Interpretación de resultados .....	80
4.1.5	Robustez .....	80
4.1.5.1	Capacidad del programa para soportar entradas incorrectas .....	80
4.1.5.2	JUnit.....	81

4.1.6	Seguridad.....	84
4.1.6.1	Niveles de permisos de usuarios, y los accesos a los datos.....	84
4.1.6.2	Flujo de trabajo de la prueba de seguridad.....	84
4.1.6.3	Tipos de usuarios.....	84
4.1.7	Usabilidad.....	85
4.1.7.1	Principios básicos.....	85
4.2	Implementación del software en el gimnasio.....	86
4.2.1	Condiciones específicas.....	86
<b>CAPÍTULO 5.....</b>		<b>88</b>
<b>ELABORACIÓN DE LAS GUÍAS DEL SOFTWARE.....</b>		<b>88</b>
5.1	Guía del administrador.....	88
5.1.1	Requerimientos de los equipos.....	88
5.1.1.1	Servidor de base de datos.....	88
5.1.1.2	Estaciones de trabajo.....	88
5.1.1.3	Software.....	88
5.1.2	Creación de la base de datos.....	88
5.1.3	Ingreso de información tablas catálogo.....	89
5.1.4	Funcionamiento.....	89
5.2	Guía del usuario.....	89
5.2.1	Acceso al sistema web interno.....	90
5.2.2	Menú principal.....	91
5.2.3	Tipos de usuarios.....	91
5.2.4	Información de usuario.....	92
5.2.5	Crear usuario.....	93
5.2.6	Perfiles de usuarios.....	94
5.2.6.1	Crear nuevo perfil.....	94
5.2.7	Información del cliente.....	95
5.2.7.1	Crear cliente.....	96
5.2.8	Información de especialidades.....	97
5.2.8.1	Crear nueva especialidad.....	97
5.2.9	Estados de una valoración física.....	98
5.2.10	Información de cobros.....	98
5.2.11	Crear nuevo tipo de cobro.....	99
5.2.12	Información de signos vitales.....	99

5.2.13	Información de área física.....	100
5.2.14	Información de tipo entrenamiento.....	100
5.2.15	Ingresar diagnóstico .....	101
5.2.16	Ingresar tratamiento .....	101
5.2.17	Ingresar plan de entrenamiento.....	101
5.2.18	Información de datos personales – nomenclatura antecedentes familiares .....	102
5.2.19	Nomenclatura tipo de entrenamiento.....	103
5.2.20	Reportes .....	103
5.2.20.1	Tipos de reportes.....	103
5.2.20.1.1	Con información del cliente .....	103
5.2.20.1.2	Con información del usuario .....	105
	<b>CONCLUSIONES.....</b>	<b>106</b>
	<b>RECOMENDACIONES.....</b>	<b>107</b>
	<b>LISTA DE REFERENCIAS .....</b>	<b>108</b>
	<b>ANEXOS .....</b>	<b>110</b>

## ÍNDICE DE FIGURAS

Figura 1. Esquema general de las tecnologías web.....	3
Figura 2. Red informática .....	5
Figura 3. JSP modelo 1 .....	7
Figura 4. JSP modelo2 .....	8
Figura 5. Pasos de descarga NetBeans .....	9
Figura 6. Estructura n Tier en Java EE .....	11
Figura 7. Estructura n Tier en Java SE.....	11
Figura 8. Esquema de la metodología DCU.....	20
Figura 9. Organigrama gimnasio .....	28
Figura 10. Diseño de la interface basado en el usuario .....	35
Figura 11. Usuarios de la aplicación web .....	36
Figura 12. Interface acceso al sitio web .....	38
Figura 13. Interface crear usuario .....	39
Figura 14. Interface ver/editar/borrar usuario .....	39
Figura 15. Interface crear cliente .....	40
Figura 16. Interface antecedentes personales .....	40
Figura 17. Tipo entrenamiento .....	41
Figura 18. Interface pantalla principal .....	41
Figura 19. Interface crear valoración física .....	42
Figura 20. Interface signos vitales .....	43
Figura 21. Interface generar cobro .....	43
Figura 22. Interface generar factura .....	44
Figura 23. Interface ingresar valoración física .....	44
Figura 24. Interface área físico regional .....	45
Figura 25. Interface diagnóstico .....	45
Figura 26. Interface plan de tratamiento .....	46
Figura 27. Interface plan de entrenamiento .....	46
Figura 28. Interface reportes del sistema web .....	46
Figura 29. Diagrama de clases .....	48
Figura 30. Caso de uso acceso al sistema .....	49
Figura 31. Caso de uso ingreso cliente .....	50

Figura 32. Caso de uso asignar valoración física .....	50
Figura 33. Caso de uso medir signos vitales .....	51
Figura 34. Caso de uso ingresar antecedentes familiares .....	52
Figura 35. Caso de uso ingresar datos personales .....	52
Figura 36. Caso de uso atender valoración física .....	53
Figura 37. Estado acceso al sistema .....	54
Figura 38. Estado ingreso cliente .....	55
Figura 39. Estado asignar valoración física .....	55
Figura 40. Estado medir signos vitales .....	56
Figura 41. Estado ingresar antecedentes familiares .....	57
Figura 42. Estado ingresar datos personales .....	57
Figura 43. Estado ingresar área físico regional .....	58
Figura 44. Estado atender valoración física .....	59
Figura 45. Secuencia acceso al sistema .....	60
Figura 46. Secuencia ingreso cliente .....	61
Figura 47. Secuencia asignar valoración física .....	62
Figura 48. Secuencia medir signos vitales .....	63
Figura 49. Secuencia ingresar antecedentes familiares .....	64
Figura 50. Secuencia ingresar datos personales .....	65
Figura 51. Secuencia ingresar área física regional .....	66
Figura 52. Secuencia atender valoración física .....	67
Figura 53. Diagrama diseño conceptual de la aplicación web .....	69
Figura 54. Agregar servidor proxy .....	76
Figura 55. Crear grupo de hilos .....	77
Figura 56. Propiedades de hilos .....	77
Figura 57. Seleccionar plan de pruebas .....	78
Figura 58. Apache JMeter permite visualizar el resultado .....	78
Figura 59. Resultado final .....	80
Figura 60. Crear JUnit tests .....	81
Figura 61. Generación de código JUnit .....	82
Figura 62. Ejecutar prueba JUnit .....	83
Figura 63. Verificación de ejecución JUnit .....	83
Figura 64. Acceso al sistema web interno .....	90
Figura 65. Pantalla menú principal .....	91

Figura 66. Mostrar lista de usuarios .....	92
Figura 67. Crear usuario .....	93
Figura 68. Mostrar lista de perfiles de usuarios .....	94
Figura 69. Crear nuevo perfil .....	94
Figura 70. Mostrar lista de clientes .....	95
Figura 71. Modificar clientes ingresados .....	95
Figura 72. Ingreso de nuevo cliente .....	96
Figura 73. Mostrar lista de especialidades .....	97
Figura 74. Crear nueva especialidad .....	97
Figura 75. Mostrar estados de una valoración física .....	98
Figura 76. Mostrar lista de cobros .....	98
Figura 77. Crear nuevo tipo de cobro .....	99
Figura 78. Ingresar signos vitales .....	99
Figura 79. Determinar área física .....	100
Figura 80. Tipo de entrenamiento .....	100
Figura 81. Ingresar diagnóstico .....	101
Figura 82. Ingresar tratamiento .....	101
Figura 83. Ingresar plan de entrenamiento .....	101
Figura 84. Lista de antecedentes familiares .....	102
Figura 85. Lista tipo entrenamiento .....	103
Figura 86. Submenú reportes cliente .....	103
Figura 87. Buscar información por id de cliente .....	104
Figura 88. Reporte búsqueda de cliente .....	104
Figura 89. Buscar información por id del usuario .....	105
Figura 90. Reporte búsqueda de usuario .....	105

## ÍNDICE DE TABLAS

Tabla 1. Antecedentes personales .....	30
Tabla 2. Signos vitales y antropometría.....	31
Tabla 3. Área físico regional.....	31
Tabla 4. Diagnóstico .....	32
Tabla 5. Estados de la cita.....	42
Tabla 6. Resumen del reporte.....	79

## ÍNDICE DE ANEXOS

Anexo 1. Formulario para valoración física.....	110
Anexo 2. Factura Absolut Gym .....	111
Anexo 3. Glosario de términos .....	112

## **RESUMEN**

El proyecto presentado expone la información sobre el desarrollo de la aplicación web para el gimnasio “Absolut Gym”. Mediante el cual se pretende mejorar los procesos actuales de atención mediante un software que permita registrar los datos realizando una adecuada administración de clientes y plan de ejercicios de entrenamiento físico personalizado.

En el capítulo uno se desarrolla las bases teóricas para el desarrollo del software. Describiendo los temas desde las aplicaciones web hasta la metodología de desarrollo a utilizar.

En el capítulo dos se describe los datos obtenidos en las visitas al Gimnasio. Análisis de la información de los procesos y documentos utilizados. Posteriormente se describe cómo va a ser el funcionamiento de la aplicación. Describiendo los módulos de Gestión de Cliente que administra la información personal del cliente. Planes de Entrenamiento que registra el criterio del entrenador en base a los análisis realizados. Facturación que registra los servicios y consumos que realice el cliente en el gimnasio. Reportes que muestra información personal del cliente, citas de evaluación de aptitud física y facturas generadas en la aplicación.

En el capítulo tres se realiza el diseño del software, diseños de interfaces y modelamientos de la aplicación.

En el capítulo cuatro se muestra las pruebas realizadas a la aplicación, en donde se evalúa el rendimiento, robustez, seguridad, usabilidad.

En el capítulo cinco se redacta las guías de administrador y de usuario para el manejo de la aplicación.

## **ABSTRACT**

The project shows the information presented on the development of the web application to the gym "Absolut Gym". By which aims to improve the current processes of care by software that allows recording data by performing a proper client management and exercise plan designed physical training.

In Chapter one develop the theoretical basis for the development of software. Describing topics from web applications to the development methodology used.

In Chapter two are described data from visits described Gym. Like information analysis of processes and documents used on Gym. Later described how you will be running the application. Describing the Client Management Modules to manage personal customer information. Training Plans that records the coach's criteria based on analyzes performed. Billing services and consumption records made by the client in the gym. Reports showing personal customer information, quotes fitness assessment and invoices generated in the application.

In Chapter three is made design software, interfaces, designs and models application

In Chapter four are show the tests on the application where performance, robustness, security, usability assesses shown.

In Chapter five are written administrator and user Guides to manage the application developed.

## INTRODUCCIÓN

Antes de internet, para transmitir las ideas de una persona a otra, y poder conservarlas para futuras generaciones, se usaban muros, pergaminos, libros, y otras más. En la actualidad toda la información que se desee exponer al mundo se lo puede hacer mediante el internet.

Las aplicaciones web tienen como concepto que mediante un navegador o browser (en inglés), podemos visualizar documentos escritos en HTML, que se encuentran albergados en un servidor Web en algún lugar del mundo.

Como la tecnología sigue progresando, la creación de estos documentos HTML, ya no solo son un hiper texto estático, sino que el usuario ya puede interactuar con estas páginas dinámicas, introduciendo datos actualizaciones en línea.

Según la situación o requerimientos de los usuarios, el diseño de la aplicación web y la tecnología se escoge por factores como, la cantidad de incidencias a la aplicación web, trabajo con base de datos, y robustez de la aplicación web.

Básicamente las aplicaciones web están albergadas en un servidor web (lado del servidor). Los usuarios mediante un navegador por internet pueden acceder a esta aplicación (lado del cliente), enviando peticiones al servidor, para que estas sean respondidas por el servidor según la funcionalidad de la aplicación web.

# CAPÍTULO 1

## APLICACIONES WEB

Antes de internet, para transmitir las ideas de una persona a otra, y poder conservarlas para futuras generaciones, se usaban muros, pergaminos, libros, y otras más. En la actualidad toda la información que se desee exponer al mundo se lo puede hacer mediante el internet.

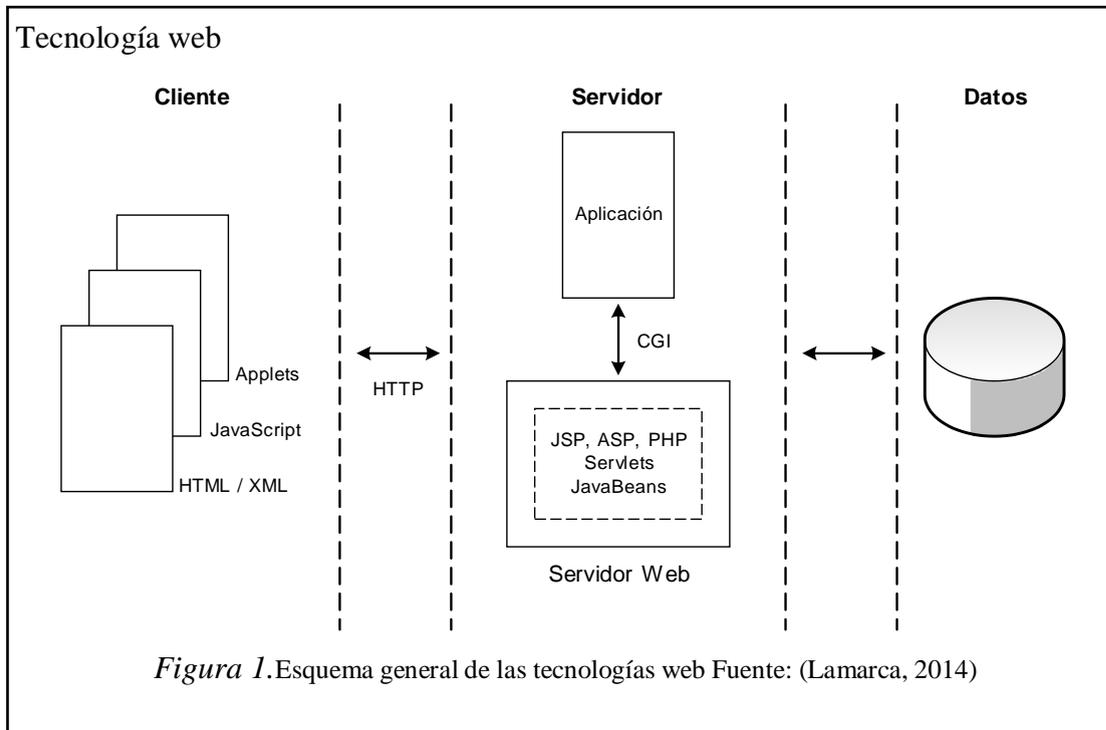
Las aplicaciones web tienen como concepto que mediante un navegador o browser (en inglés), podemos visualizar documentos escritos en HTML, que se encuentran albergados en un servidor Web en algún lugar del mundo.

Como la tecnología sigue progresando, la creación de estos documentos HTML, ya no solo son un hiper texto estático, sino que el usuario ya puede interactuar con estas páginas dinámicas, introduciendo datos actualizaciones en línea.

Según la situación o requerimientos de los usuarios, el diseño de la aplicación web y la tecnología se escoge por factores como, la cantidad de incidencias a la aplicación web, trabajo con base de datos, y robustez de la aplicación web.

Básicamente las aplicaciones web están albergadas en un servidor web (lado del servidor). Los usuarios mediante un navegador por internet pueden acceder a esta aplicación (lado del cliente), enviando peticiones al servidor, para que estas sean respondidas por el servidor según la funcionalidad de la aplicación web.

A continuación se presenta en la figura 1, un esquema general donde se muestran los tipos de tecnologías involucradas en la generación e interacción de la aplicación web dinámicas que utiliza CGI, *Common Gateways Interface*, y *servlets* a través de la tecnología *Java Server Pages*.(Lamarca, 2014)



Para entender mejor el funcionamiento y características de una aplicación web, se describirá las partes básicas y tecnologías usadas, a pesar que se encuentran en continua evolución y se corre el riesgo de quedar obsoletas.

### 1.1.1 El navegador web - browser

Es considerado como la interface universal del usuario. Sus funciones principales están: Petición de páginas Web, presentación correcta de sus contenidos, y gestión de los posibles errores.

Los creadores de navegadores utilizan archivos de texto tipo *script*. (SENA, 2011) Un conjunto de instrucciones que deben ser interpretados línea a línea en tiempo real para su ejecución. Los scripts pueden estar embebidos en otro lenguaje para aumentar las funcionalidades, como es el caso de los scripts PHP o Javascript en código HTML. El soporte de scripts de HTML es independiente del lenguaje de scripts.

Los scripts ofrecen a los autores la posibilidad de extender los documentos HTML de manera activa e interactiva. Por ejemplo:

- Pueden evaluarse los scripts a medida que se carga el documento para modificar los contenidos del documento dinámicamente.
- Los scripts pueden acompañar a un formulario para procesar los datos a medida que éstos se introducen. Los diseñadores pueden rellenar dinámicamente partes de un formulario según los valores de los otros campos. También pueden asegurarse de que los datos introducidos concuerden con rangos de valores predeterminados, de que los campos sean consistentes entre sí, etc.
- Los scripts pueden ser llamados por eventos que afecten al documento, como la carga, la descarga, el movimiento del foco sobre los elementos, los movimientos del ratón, etc.
- Los scripts pueden ser vinculados a controles de formulario (p.ej., botones) para producir elementos gráficos para la interfaz del usuario.

Estos archivos permiten la manipulación de contenidos en los documentos, y proporcionan las soluciones llamadas del lado del cliente (*client side*), para hacer validaciones en las páginas antes de enviarlos al servidor, con un alto grado de interacción con el usuario dentro del documento.

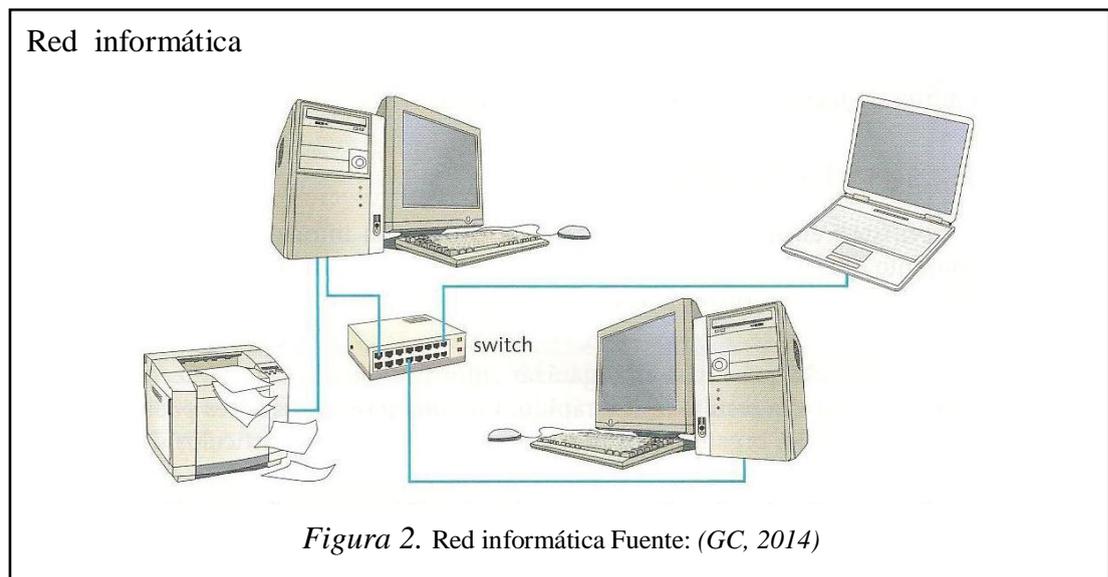
### **1.1.2 HTML dinámico**

También gestionan el llamado HTML dinámico (Dynamic HTML, DHTML), está compuesto de HTML y hojas de estilo en cascada (Cascade Style Sheet, CSS), modelo de objetos y *scripts* de programación, que realizan un formateo o posicionan correctamente los distintos elementos del documento HTML.(SENA, 2011)

Los navegadores permiten la ejecución de aplicaciones dentro de documentos mostrados. Las dos posibilidades más populares son la tecnología *ActiveX* y los *Applets Java*. Son pequeños programas que se descargan del servidor web y se ejecutan en el cliente.

### 1.1.3 Redes informáticas

Una red informática está constituida por un conjunto de ordenadores y otros dispositivos, conectados por medios físicos o sin cable, con el objetivo de compartir unos determinados recursos. Éstos pueden ser aparatos (hardware), como impresoras, sistemas de almacenamiento, etc., o programas (software), que incluyen aplicaciones, archivos, etc.(GC, 2014)



### 1.2 Software libre

Según Free Software Foundation, el software libre se refiere a las 4 libertades básicas de los usuarios:

- *Libertad 1*, libertad de usar el programa, con cualquier propósito
- *Libertad 2*, libertad de estudiar cómo funciona el programa, y adaptarlo a tus necesidades
- Libertad 3, libertad de distribuir copias
- Libertad 4, libertad de mejorar el programa y hacer públicas las mejoras a los demás

El software libre permite ejecutar, copiar, distribuir, estudiar, modificar el software y distribuirlo modificado. Adaptándolo a las necesidades del usuario y sin estar atados a un fabricante con sus derechos de autor.

### 1.2.1 Software propietario

El software no libre (también llamado software propietario, software privativo, software privado y software con propietario) se refiere a cualquier programa informático en el que los usuarios tienen limitadas las posibilidades de usarlo, modificarlo o redistribuirlo (con o sin modificaciones), o cuyo código fuente no está disponible o el acceso a éste se encuentra restringido.

## 1.3 Plataforma de desarrollo

### 1.3.1 Evolución de la tecnología

En base de la tecnología aplicada se divide en generaciones. La tecnología usada de *primera generación* incluye a **CGI**, que es un componente para ejecutar programas externos en un servidor web. El inconveniente con los scripts CGI es la escalabilidad, porque se crea un nuevo proceso para cada repetición.

En la segunda generación incluye vendedores de servidores web que proporcionan **plug-ins** y **APIs** para sus servidores, obligando a usar productos específicos. Como es el ejemplo de Microsoft con **ASP**, *Active Server Pages*, que hicieron más sencillo crear contenido dinámico, a pesar de recurrir solo a Microsoft IIS o Personal Web Server. Otra tecnología de segunda generación son los **Servlets**, que hacen más fácil escribir aplicaciones del lado del servidor usando tecnología Java. El problema con los Servlets o CGI, debemos seguir su ciclo de vida; escribir, compilar y desplegar.

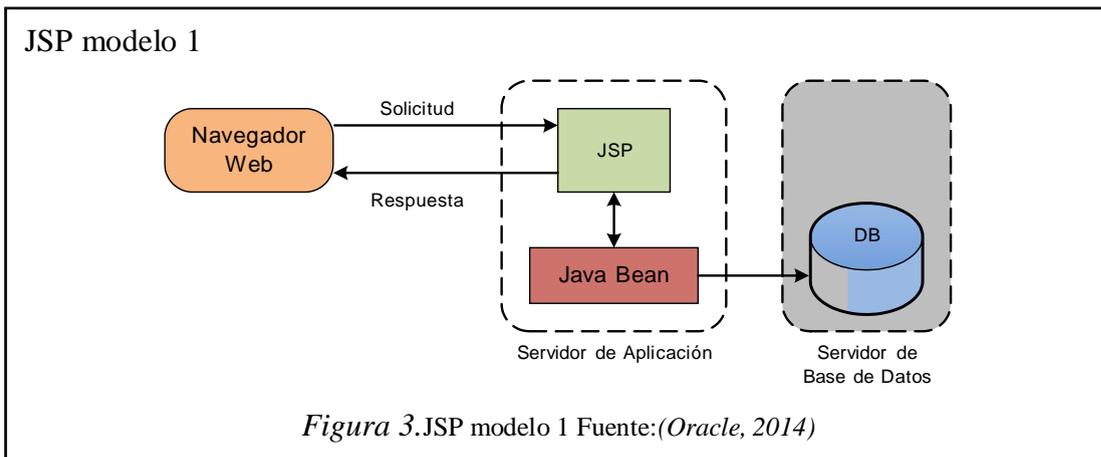
### 1.3.2 Tecnología JSP

Las páginas **JSP** son la solución de *tercera generación*, que se pueden combinar fácilmente con algunas soluciones de segunda generación, más fácil y rápido de construir aplicaciones basadas en la Web con contenido dinámico. Podemos crear

aplicaciones que se ejecuten en varios servidores, de múltiples plataformas, por la razón que Java es en esencia un lenguaje multiplataforma. (Torrijos , 2014)

Las páginas JSP están compuestas de código HTML / XML mezclado con etiquetas especiales para programar scripts del servidor.El motor de JSP está basado en los *Servlets* de Java, dentro de un contenedor como Tomcat. Para crear las aplicaciones en JSP se generan los archivos con extensión **.jsp** que incluyen dentro la estructura de etiquetas HTML, las sentencias Java para ejecutar en el servidor. Antes de que las páginas sean funcionales, el motor lleva a cabo una fase de interpretación del archivo a un servlet, por lo general esta interpretación se realiza cuando se recibe la primera solicitud de la página.

La tecnología JSP presenta dos enfoques para la construcción de aplicaciones web: Las arquitecturas “JSP Model 1” y “JSP Model 2”. La diferencia entre estos dos modelos está donde tiene lugar el procesamiento. Para el modelo 1, la página web se encarga de tramitar las solicitudes y devolver respuestas a los clientes.(Magaña, 2014)



El modelo 2, integra el uso de los servlets y las páginas JSP. En este modelo las páginas JSP se usan para la capa de presentación, y los servlets para las tareas de procesamiento, o para la lógica del negocio. Los servlets son los responsables del procesamiento de las solicitudes y la creación de los Beans necesarios para la página JSP.

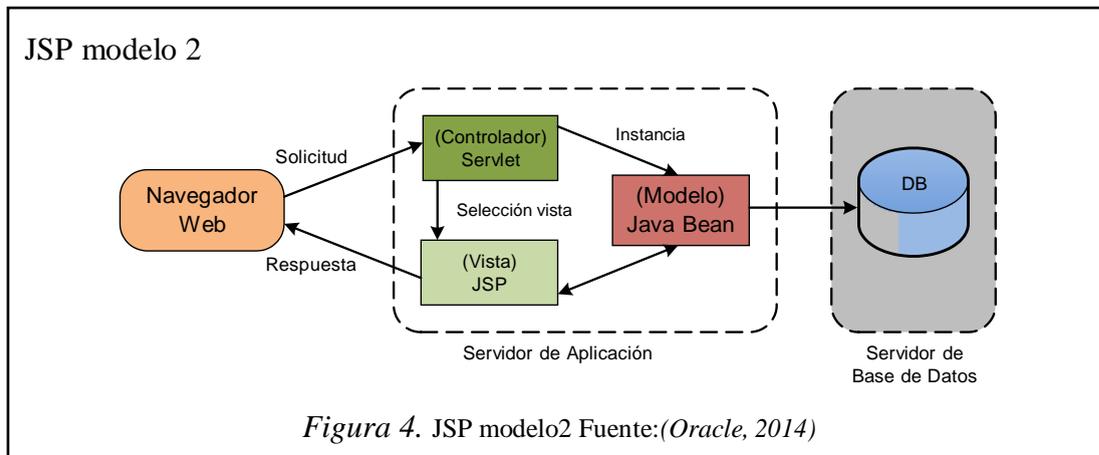


Figura 4. JSP modelo2 Fuente:(Oracle, 2014)

#### 1.4 Netbeans

Fue desarrollado por Sun Microsystems, es un software libre y gratuito. NetBeans es un software denominado “Entorno de Desarrollo”, donde se realizan proyectos para desarrollar programas en un lenguaje de programación determinado, de manera más fácil al método tradicional con un editor de texto.

NetBeans se usa principalmente para el desarrollo de aplicaciones en Java, con paquetes adicionales se puede programar en otros lenguajes como Ruby, C/C++ o PHP. Una característica importante que destacar es que funciona en diversos sistemas operativos, tal como Linux, Windows, Mac o Solaris.

Obtener este software es muy fácil, solo se debe ir a la dirección <http://netbeans.org/downloads/>, luego se elige la plataforma en la que se va a instalar (1), y se escoge el tipo de descarga (2).

## Descarga NetBeans

Descargar NetBeans IDE 7.0.1

Idioma del IDE: English 1 Plataforma: Windows

Paquetes de descarga de NetBeans IDE

Tecnologías *	Java SE	Java EE	C/C++	PHP	All
NetBeans Platform SDK	•	•			•
Java SE	•	•			•
Java EE		•			•
Java ME					•
Java Card™ 3 Connected					•
C/C++			•		•
Groovy				•	•
PHP				•	•
Servidores incluidos					
GlassFish Server Open Source Edition 3.1.1		•			•
Apache Tomcat 7.0.14					•

Download Libre, 66 MB   Download Libre, 158 MB   Download Libre, 44 MB   Download Libre, 42 MB   Download Libre, 245 MB

Figura 5. Pasos de descarga NetBeans Fuente: (Netbeans, 2014)

Para este caso de descarga para plataforma Windows NetBeans IDE versión 7.0.1, el paquete Java EE.

### 1.4.1 Java EE

Es una plataforma de programación parte de la Plataforma Java para desarrollar y ejecutar software de aplicaciones en el lenguaje de programación Java con arquitectura de N capas distribuidas y que se apoya ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones. (Oracle, 2014)

#### 1.4.1.1 Historia

La especificación original J2EE fue desarrollada por Sun Microsystems. Comenzando con J2EE 1.3, la especificación fue desarrollada bajo el Java Community Process. JSR 58 especifica J2EE 1.3 y JSR 151 especifica J2EE 1.4.

El SDK de J2EE 1.3 fue distribuido inicialmente como beta en abril de 2001. La beta del SDK de J2EE 1.4 fue distribuida por Sun en diciembre de 2002. La especificación Java EE 5 fue desarrollada bajo el JSR 244 y el lanzamiento final se hizo el 11 de mayo de 2006.

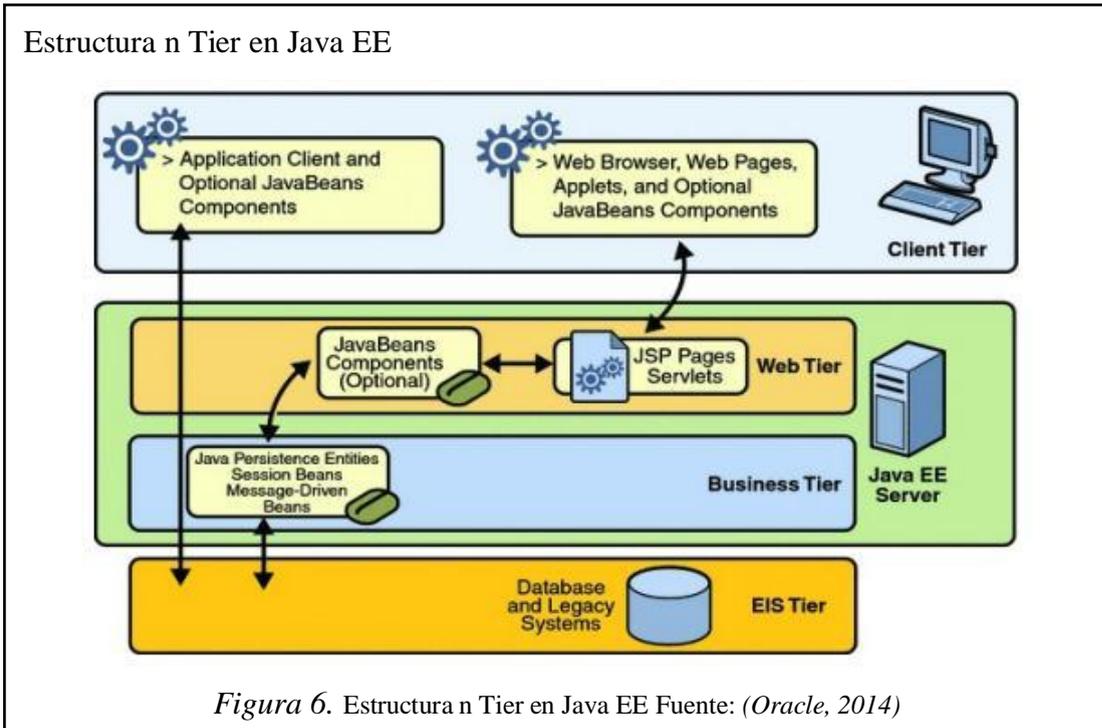
### **1.4.1.2 Plataforma Java EE**

La plataforma Java EE está definida por una especificación. Similar a otras especificaciones del Java Community Process, Java EE es también considerada informalmente como un estándar debido a que los proveedores deben cumplir ciertos requisitos de conformidad para declarar que sus productos son conformes a Java EE; estandarizado por The Java Community Process / JCP.(Oracle, 2014)

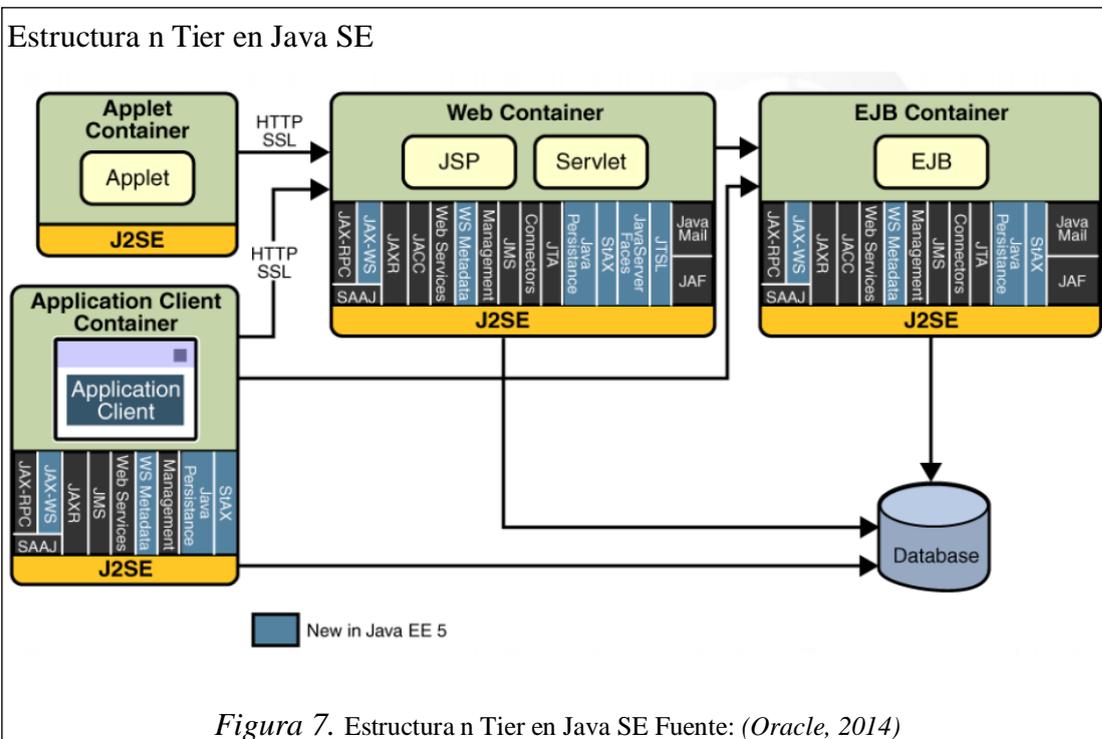
Java EE tiene varias especificaciones de API, tales como JDBC, RMI, e-mail, JMS, Servicios Web, XML, etc. y define cómo coordinarlos. Java EE también configura algunas especificaciones únicas para Java EE para componentes. Estas incluyen Enterprise JavaBeans, servlets, portlets (siguiendo la especificación de Portlets Java), JavaServer Pages y varias tecnologías de servicios web. Ello permite al desarrollador crear una Aplicación de empresa portable entre plataformas y escalable, a la vez que integrable con tecnologías anteriores. Otros beneficios añadidos son, por ejemplo, que el servidor de aplicaciones puede manejar transacciones, la seguridad, escalabilidad, concurrencia y gestión de los componentes desplegados, significando que los desarrolladores pueden concentrarse más en la lógica de negocio de los componentes en lugar de en tareas de mantenimiento de bajo nivel.

### 1.4.1.2.1 Arquitectura en Java EE

Java EE tiene una arquitectura con 3 o 4 capas diferenciadas



### 1.4.1.2.2 Arquitectura en Java SE



### **1.4.1.2.3 APIs generales**

Las APIs de Java EE incluyen varias tecnologías que extienden la funcionalidad de las APIs base de Java SE.(Oracle, 2014)

#### **1.4.1.2.3.1 javax.ejb.\***

La API Enterprise JavaBeans define un conjunto de APIs que un contenedor de objetos distribuidos soportará para suministrar persistencia, RPCs (usando RMI o RMI-IIOP), control de concurrencia, transacciones y control de acceso para objetos distribuidos.

#### **1.4.1.2.3.2 javax.naming**

Los paquetes javax.naming, javax.naming.directory, javax.naming.event, javax.naming.ldap y javax.naming.spi definen la API de Java Naming and Directory Interface (JNDI).

#### **1.4.1.2.3.3 java.sql**

Los paquetes java.sql y javax.sql definen la API de JDBC.

#### **1.4.1.2.3.4 java.transaction.\***

Estos paquetes definen la Java Transaction API (JTA).

#### **1.4.1.2.3.5 javax.xml.\***

Estos paquetes definen la API JAXP.

#### **1.4.1.2.3.6 javax.jms.\***

Estos paquetes definen la API JMS.

#### 1.4.1.2.3.7 **javax.persistence**

Este paquete provee las clases e interfaces para gestionar la interacción entre los proveedores de persistencia, las clases administradas y los clientes de la Java *Persistence API*.

#### 1.4.1.3 **Desarrollo de bajo coste – plataforma**

Uno de los beneficios de Java EE como plataforma es que es posible empezar con poco o ningún coste. La implementación Java EE de Sun Microsystems puede ser descargada gratuitamente, y hay muchas herramientas de código abierto disponible para extender la plataforma o para simplificar el desarrollo. (Oracle, 2014)

Ejemplos de herramientas de desarrollo Java de código abierto de terceras partes son:

- NetBeans IDE, un IDE basado en Java
- La plataforma Eclipse, un IDE basado en Java
- Expand, un plugin de Eclipse, para desarrollo rápido.
- Jedit, de código abierto, un IDE basado en Java
- Apache Software Foundation Apache Ant, una herramienta de construcción automática
- Apache Software Foundation Apache Maven, una herramienta de construcción automática y gestión de dependencias
- JUnit, un framework para Pruebas de unidad automatizadas
- Apache Software Foundation Apache Tomcat, un contenedor web de Servlet/JSP
- Jetty, un servidor web y un contenedor web Servlet/JSP
- Spring, un framework para desarrollo de aplicaciones Java EE
- Struts, un framework para desarrollar aplicaciones web EE conforme al patrón Modelo Vista Controlador
- OpenXava, un framework de código abierto para desarrollo fácil de aplicaciones de negocio J2EE
- JDeveloper, un IDE basado en Java y desarrollado por Oracle
- JBuilder, desarrollado por Borland

- JavaServer Faces, un framework para desarrollar aplicaciones web EE conforme al patrón Modelo Vista Controlador, desarrollado por Sun
- Aries, un framework de código abierto para desarrollo rápido de aplicaciones de gestión JEE.

#### **1.4.1.4 Servidores de aplicaciones Java EE 5 certificados**

- JOnAS, un servidor de aplicaciones de código abierto de ObjectWeb
- JBoss, desarrollado inicialmente por JBoss Inc y adquirido posteriormente por Red Hat. Existe una versión de código abierto soportada por la comunidad y otra empresarial.
- Sun Java System Application Server Platform Edition 9.0, basado en GlassFish
- Oracle WebLogic Application Server 10.0 (Antes BEA Systems)
- Servidor de Aplicaciones SAP NetWeaver, Java EE 5 Edition de SAP
- JEUS 6, un Servidor de aplicaciones específico de Linux de TmaxSoft
- Apache Geronimo 2.0
- IBM WebSphere Application Server Community Edition 2.0, based on Apache Geronimo
- Oracle Containers for Java EE 11
- GlassFish, un servidor de aplicaciones de código abierto de Sun
- Apache OpenEJB via Apache Geronimo

#### **1.4.1.5 Servidores de aplicaciones J2EE 1.4 certificados**

- Geronimo, un servidor de aplicaciones de código abierto de Apache
- JOnAS, un servidor de aplicaciones de código abierto de ObjectWeb
- Servidor de Aplicaciones SAP NetWeaver, Java EE 5 Edition de SAP
- Sun Java System Web Server
- Sun Java System Application Server
- IBM WebSphere Application Server (WAS) de IBM
- Servidor de aplicaciones WebLogic de BEA Systems

#### **1.4.1.6 Servidores de aplicaciones J2EE 1.3 certificados**

- Servidor de Aplicaciones JRun de Macromedia

### **1.5 Base de datos**

Los datos son hechos y cifras que al ser procesados constituyen una información, sin embargo, muchas veces datos e información se utilizan como sinónimos. En su forma más amplia los datos pueden ser cualquier forma de información: campos de datos, registros, archivos, texto (colección de palabras), hojas de cálculo (datos en forma matricial), imágenes (lista de vectores o cuadros de bits), vídeo (secuencia de tramas), etc.(DNTI, 2011)

Una base de datos es un conjunto de datos organizados, entre los cuales existe una correlación y que además, están almacenados con criterios independientes de los programas que los utilizan. También puede definirse, como un conjunto de archivos interrelacionados que es creado y manejado por un sistema de gestión o de administración de base de datos (Data Base Management System - DBMS).

#### **1.5.1 Sistema de administración de bases de datos**

Un sistema de administración de bases de datos (DBMS) es un software que permite manipular las bases de datos. (DNTI, 2011)

- Construir
- Utilizar
- Mantener
- Reorganizar

#### **1.5.2 Propiedades fundamentales**

- Independencia de los datos
- Acceso eficiente a los datos
- Integridad y seguridad de los datos
- Administración de los datos

- Acceso concurrente y recuperación en caso de accidente (*crash*)

### **1.5.3 Características de los DBMS**

Las características que presenta un DBMS son las siguientes:

- Brinda seguridad e integridad a los datos.
- Provee lenguajes de valoración física (interactivo).
- Provee una manera de introducir y editar datos en forma interactiva.
- Existe independencia de los datos, es decir, que los detalles de la organización de los datos no necesitan incorporarse a cada programa de aplicación.

### **1.5.4 Propiedades ACID**

Los DBMS deben cumplir con las propiedades ACID para las transacciones:

- Atomicity (Atomicidad: las transacciones son atómicas)
- Consistency (Consistencia: una transacción transforma un estado consistente de la BD en otro)
- Isolation (Aislamiento: las transacciones están aisladas entre sí)
- Durability (Durabilidad: después que una transacción ha sido confirmada ella persiste)

### **1.5.5 Usuarios**

- Usuario final: el que interactúa con la base de datos, por lo general a través de aplicaciones e interfaces
- Usuario especialista: el que diseña y programa aplicaciones para usuarios finales.
- DBA (*Database Administrator*): el que administra la base de datos.

### **1.5.6 Componentes funcionales de un DBMS**

Metadatos o diccionario de datos (D.D): contiene el esquema de la base de datos, los usuarios, los permisos de acceso, etc. Son datos sobre los datos. Almacena la información que permite la traducción entre los 3 niveles de la arquitectura ANSI/SPARC. (DNTI, 2011)

Optimizador de valoración física: define el plan de ejecución de operaciones solicitadas por los usuarios, de tal manera que se lleven a cabo de la manera más eficiente posible.

Manejador de transacciones: controla el acceso y la concurrencia de operaciones.

#### **Manejador de almacenamiento tiene dos componentes:**

- Manejador de archivos: recupera desde disco los bloques que contienen la información solicitada por una transacción.
- Manejador de buffer: mantiene en memoria principal la información más usada y decide cuando llevar a disco alguno de sus bloques.

## **1.6 MySQL**

MySQL es un sistema de administración de base de datos (Database Management System, DBMS) para base de datos relacionales. En otras palabras es un software para gestionar archivos de base de datos.(López, 2007)

Existen varios tipos, desde un simple archivo hasta base de datos relacionales orientada a objetos.

MySQL fue escrito en C/C++ y se destaca por su gran adaptación a diferentes entornos de desarrollo, permitiendo ser muy flexible al usar lenguajes de programación como Perl, PHP y Java, al igual que su integración en distintos sistemas operativos. (UTS, 2013)

Desde la dirección <http://dev.mysql.com/downloads/> se puede descargar el software con una breve suscripción en la página. La versión que se utiliza en la MySQL Installer 5.5.16 para la plataforma de Windows.

## **1.7 Metodología de desarrollo: XP eXtreme Programming**

XP es una metodología ágil XP que se fundamenta en la simplicidad para enfrentar los cambios dentro del desarrollo del software. XP se utiliza cuando los requisitos cambian desde la etapa inicial del diseño. La comunicación fluida permite cumplir con éxito las labores que realiza el equipo de trabajo de desarrollo de software. (Letelier, 2006)

La programación entre pares es una técnica que consiste en que dos programadores comparten una misma estación de trabajo. Se justifica su implementación con el propósito de compartir los conocimientos del equipo de trabajo con respecto a la implementación de código. La aplicación de la metodología evita así problemas de mantenimiento del software, por ejemplo es útil cuando la persona que realiza los cambios no está presente, si se presenta un problema existe otra persona que lo puede sustituir para resolver cualquier inconveniente.

### **1.7.1 Reglas y prácticas**

La metodología XP tiene se agrupa para las etapas de: planificación, diseño, desarrollo, pruebas.

#### **1.7.1.1 Etapa de planificación**

Se basa en la comunicación continua entre las partes involucradas en el proyecto, como cliente, programadores y a los coordinadores o gerentes. Se registra las “Historias de usuarios” o “casos de uso”. Luego se estima el tiempo de desarrollo, se realizan pruebas, y finalmente un cronograma de entregas. Las iteraciones entre etapas son necesarias para conocer el funcionamiento de cada código desarrollado.

### **1.7.1.2 Etapa de diseño**

El diseño se lo realiza en tres partes: simplicidad, recodificación, metáforas.

### **1.7.1.3 Etapa de desarrollo del código**

Para esta etapa se debe considerar los siguientes puntos: disponibilidad del cliente, programación dirigida por las pruebas, última versión, propiedad colectiva del código, ritmo sostenido, pruebas unitarias, detección y corrección de errores, pruebas de aceptación. (Joskowicz, 2008)

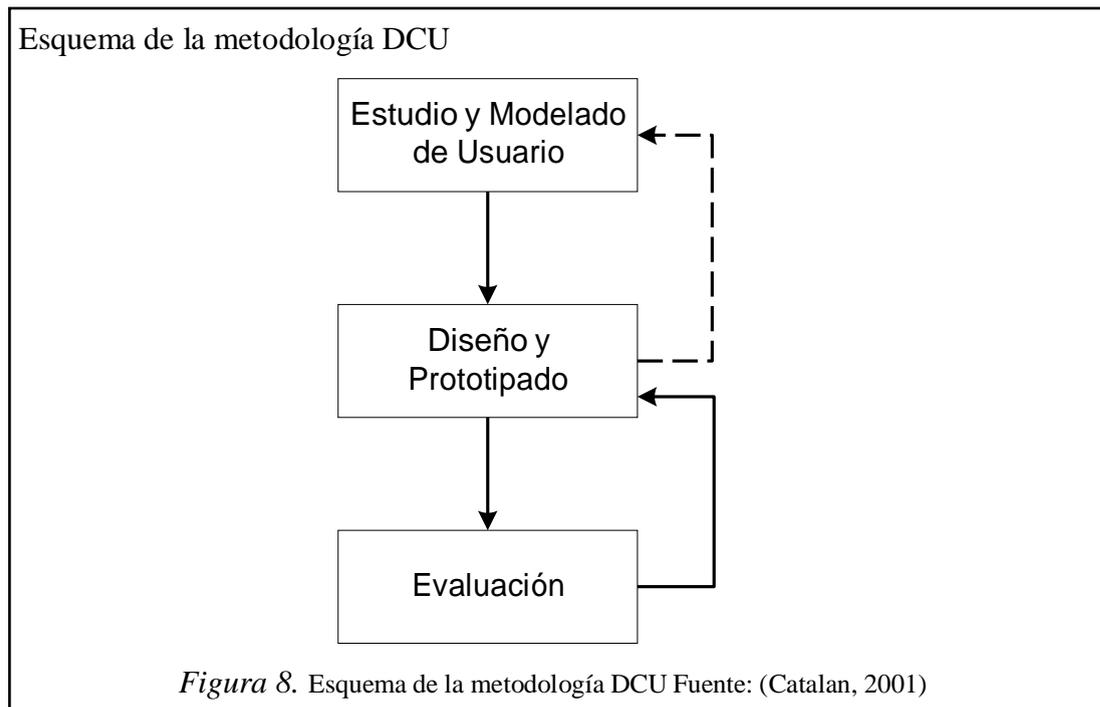
## **1.7.2 Filosofía diseño centrado en el usuario (DCU)**

DCU es todo el proceso de diseño debe estar encaminado por el usuario, objetivos, sus necesidades y características y se asume la necesidad participativa del usuario en el proceso de diseño. (Montero & Fernández, 2013)

### **1.7.2.1 Propuesta de adaptación metodológica DCU**

Se trata de un modelo muy sencillo, basado en la adaptación de técnicas y métodos de diseño centrado en el usuario para la inclusión de más tipos de usuarios en más contextos de uso, por lo que se establece una familiaridad la preferencia entre desarrolladores y modelo, ya que estos métodos son ampliamente conocidos por el entorno de profesionales del diseño web. (Catalan, 2001)

El proceso interactivo de diseño centrado en el usuario es muy sencillo y se lo representa en el siguiente esquema:



El número de etapas de DCU, así como de métodos y técnicas a aplicar en cada etapa, dependerá del tipo y complejidad del sitio web a desarrollar. Desde el punto de vista de la accesibilidad final del sitio web, la etapa más importante y que requiere de mayor adaptación para satisfacer las necesidades de “todos” los usuarios es la de evaluación, por lo que será en ésta sobre la que se prestará mayor atención, frente a otras como la de diseño, que en su aplicación no diferirán tanto.

La filosofía DCU propuesta puede ser descrita a través de varias reglas básicas:

1. El proceso de diseño debe estar centrado en el usuario, en “todos” los usuarios. Cuanto más se conozca al usuario, más adaptado podrá estar el diseño a sus necesidades y características, y por lo tanto, más usable y accesible resultará.
2. Las etapas de diseño, prototipado y evaluación son cíclicas e iterativas. A lo largo de todo el proceso de desarrollo, lo que se diseñe deberá ser evaluado a través de su prototipado. Siempre será menos costoso reconducir un diseño que rediseñar completamente.
3. La etapa de estudio y modelado del usuario se realizará antes de cualquier diseño. Sólo debería ser revisada o remodelada cuando en la etapa de

evaluación se descubra que un error de diseño está causado por algún defecto en el estudio y/o modelado del usuario.

### **1.7.3 Estudio y modelado del usuario**

El primer paso en el modelado del usuario es identificar y definir la audiencia de sitio web. Al acoger una metodología basada en diseño inclusivo no significa que entre la audiencia objetiva se deba incluir a todo tipo de usuarios, significa que, no se tomará en cuenta a los usuarios que no están destinados al proyecto.

Por lo tanto, en esta etapa se deben identificar los objetivos y necesidades del usuario objetivo, sin excluir a usuarios que en este caso serán personas mayores, o usuarios limitados por la lectura de textos relativamente pequeños, porque en su mayoría estos usuarios formarán parte de la audiencia potencial. La razón de incluir a estos usuarios en el proceso de desarrollo se fundamenta en que, aunque compartirán objetivos con el “usuario medio”, tendrán necesidades de acceso diferentes.

Esta información sobre el usuario será obtenida a través de dos cauces. Por un lado mediante estudios directos (encuestas, estudios de campo, entrevistas...) y por otro, mediante el estudio y análisis de literatura científica sobre las limitaciones y necesidades propias de usuarios que accederán al sistema informático.

La investigación sobre la audiencia nos proporcionará una información básica para empezar a trabajar y sobre la que basar cualquier decisión sobre el diseño.

Para que toda esta información sea útil y manejable por el diseñador, es necesario resumirla, sintetizarla y organizarla, es decir, realizar el modelado del usuario. En este paso se definen clases o perfiles de usuarios, para así poder diseñar con el objetivo de satisfacer las necesidades propias de cada grupo de usuarios.

Esta técnica de modelado del usuario se basa en la definición de arquetipos de usuarios, que patrones de conducta, objetivos y necesidades. Estos arquetipos, representan llamados "personas", son descripciones en forma narrativa de usuarios, con una identidad ficticia (nombre, edad, sexo...), pero con perfiles y características

basadas en la investigación sobre la audiencia real del sitio web. Así mismo, se deben definir "escenarios" descripciones de situaciones de uso del producto.

#### **1.7.4 Diseño y prototipado**

Un único diseño no podrá satisfacer las necesidades de interacción de todos los usuarios de la misma forma que lo harían diferentes diseños para cada clase de usuario. Una opción bastante común entre sitios web autodenominados "accesibles" es la de proporcionar versiones alternativas, como la popular versión "solo texto" o las de distinto idioma.

Un opción más eficaz que la de ofrecer múltiples interfaces, aunque más compleja desde el punto de vista técnico, es la de diseñar interfaces adaptables dinámicamente a las necesidades del usuarios. En estos casos se debe representar el conocimiento del sistema acerca de las diferentes clases o perfiles de usuario, así como reglas de inferencia y condiciones que determinen cómo y cuándo adaptar la forma y presentación de los contenidos dinámicamente.

La mejor forma de asegurar la adaptabilidad de un sitio web es mediante la separación de contenido, presentación, estructura lógica e interacción. La separación entre contenido, presentación y estructura lógica, así como los beneficios derivados de este modo de diseñar, son ampliamente conocidos entre desarrolladores web. Diferentes usuarios necesitarán diferentes formas de presentación de información, pero también elementos de interacción adaptados. En la web, el elemento básico de interacción es el enlace, debido a su simplicidad de comportamiento no necesitará de esta abstracción, sin embargo otros elementos más complejos deberán presentarse y responder ante la interacción de diferente forma según las características, habilidades y limitaciones del usuario.

La etapa de diseño en el proceso de DCU y por ende del diseño de software para el gimnasio, debe ser entendida como una toma de decisiones predictiva, en base al conocimiento extraído de la audiencia en la fase de estudio del usuario e interpretativa en base a los resultados de la fase de evaluación, por parte del

diseñador sobre qué solución de diseño será la más adecuada para asegurar la facilidad de uso y acceso del sitio web.

Existen diferentes tipos de prototipado que pueden ser clasificados según el nivel de funcionalidad reproducida:

- Prototipado horizontal: se reproduce gran parte del aspecto del sitio, pero sin que esos modelos de interfaz estén respaldados por la funcionalidad real que tendrá finalmente el sitio.
- Prototipado vertical: se reproduce únicamente el aspecto visual de una parte del sitio, pero la parte reproducida poseerá la misma funcionalidad que el sitio web una vez implementado.

Otra clasificación complementaria y que nos puede resultar de más utilidad es según el grado de fidelidad o calidad del prototipo:

- Prototipado de alta fidelidad: el aspecto del prototipo será muy parecido al del interfaz del sitio web una vez finalizado su desarrollo.
- Prototipado de baja fidelidad: el aspecto del prototipo distará bastante del que tenga el interfaz del sitio web final. Como ventaja tiene el bajo coste de su desarrollo.

### **1.7.5 Evaluación de la accesibilidad**

Esta etapa del desarrollo es probablemente la más importante, ya que será en la evaluación cuando se valide cualquier decisión de diseño, así como cuando se descubran errores metodológicos. La evaluación debe realizarse desde las primeras etapas del ciclo de desarrollo del sitio, primero sobre los prototipos y después sobre el sitio web implementado. Esto quiere decir que cuanto más tarde se evalúe el diseño, más costosa resultará la reparación de errores de usabilidad y accesibilidad, ya que siempre es más económico reconducir el diseño que rediseñar completamente.

La evaluación de la accesibilidad puede ser llevar a cabo a través de un conjunto de técnicas y métodos, lo que se denomina una meta-metodología ("meta-method"). Dependiendo del grado de exhaustividad requerido y del presupuesto disponible, se optará por hacer uso de más o menos técnicas de evaluación, así como por realizar mayor o menor número de evaluaciones durante el ciclo de desarrollo.

### **1.7.6 Test con usuarios**

Las pruebas con usuarios reales son técnicas de evaluación de usabilidad tradicionalmente usadas en DCU. Se trata de pruebas realizadas en "laboratorio", en las que se observa cómo un grupo de usuarios, reales o potenciales, lleva a cabo una serie de tareas encomendadas, para posteriormente realizar un análisis de los problemas de usabilidad con los que se han encontrado durante la prueba.

Las pruebas de usabilidad con usuarios, con algunos cambios, pueden ser usadas también para evaluar la accesibilidad.

## **1.8 Normativas de pruebas de sistemas de software**

En la Ingeniería del Software, las pruebas de rendimiento son aquellas que son realizadas para determinar qué tan rápido un sistema realiza una tarea bajo ciertas condiciones pre-planificadas de trabajo. Estas pruebas también son utilizadas para validar y verificar diferentes aspectos de la calidad de software, como por ejemplo, escalabilidad, fiabilidad y el buen uso de los recursos. Las pruebas de rendimiento constituyen un subconjunto de la Ingeniería de Pruebas, la cual se esfuerza en mejorar el rendimiento, basándose en el diseño y la arquitectura de un sistema, antes de la realización del proceso de codificación.

Las pruebas de rendimiento pueden servir para diferentes propósitos. Pueden demostrar que el sistema cumple los criterios de rendimiento. Pueden comparar dos sistemas para encontrar cuál de ellos funciona mejor o medir que partes del sistema o de cargas de trabajo provocan que el conjunto ofrezca bajo rendimiento. Para su diagnóstico, los ingenieros de software utilizan herramientas que permiten

monitorear y medir qué partes de un dispositivo o software contribuyen a disminuir el rendimiento.

Es fundamental para alcanzar un buen nivel de rendimiento de un nuevo sistema, que los esfuerzos en estas pruebas comiencen en el inicio del proyecto de desarrollo y se amplíen durante su construcción. Cuanto más se tarde en detectar un defecto de rendimiento, mayor es el costo de la solución. Esto es cierto también para las pruebas funcionales, pero mucho más en las pruebas de rendimiento, debido a que su ámbito de uso es de principio a fin.

En las pruebas de rendimiento, a menudo es crucial (y con frecuencia difícil de conseguir) que las condiciones de prueba sean similares a las esperadas en el uso real. Esto es, sin embargo, casi imposible en la práctica. La razón es que los sistemas en producción tienen un carácter aleatorio sobre la carga de trabajo y aunque en las pruebas se intente dar lo mejor de sí para imitar el volumen de trabajo que pueda tener el entorno de producción, es imposible reproducir exactamente la variabilidad de ese trabajo, salvo en el sistema más simple.

Existen diferentes tipos de pruebas de rendimiento, las cuales son muy útiles, siempre y cuando sean usadas a través de herramientas que permitan visualizar los detalles técnicos y gráficos altamente informativos de los resultados de las pruebas. Entre los tipos de pruebas de rendimiento más importantes están:

### **1.8.1 Pruebas de carga**

Este es el tipo más sencillo de pruebas de rendimiento. Una prueba de carga se realiza generalmente para observar el comportamiento de una aplicación bajo una cantidad de peticiones esperada. Esta carga puede ser el número esperado de usuarios concurrentes utilizando la aplicación y que realizan un número específico de transacciones durante el tiempo que dura la carga. Esta prueba puede mostrar los tiempos de respuesta de todas las transacciones importantes de la aplicación. Si la base de datos, el servidor de aplicaciones y otros componentes también se monitorizan, entonces esta prueba puede mostrar cual es el cuello de botella en la aplicación.

### **1.8.2 Prueba de estrés**

Esta prueba se utiliza normalmente para romper la aplicación. Se va doblando el número de usuarios que se agregan a la aplicación y se ejecuta una prueba de carga hasta que se rompe. Este tipo de prueba se realiza para determinar la solidez de la aplicación en los momentos de carga extrema y ayuda a los administradores para determinar si la aplicación rendirá lo suficiente en caso de que la carga real supere a la carga esperada.

### **1.8.3 Prueba de estabilidad (Soak Testing)**

Esta prueba normalmente se hace para determinar si la aplicación puede aguantar una carga esperada continua. Generalmente esta prueba se realiza para determinar si hay alguna fuga de memoria en la aplicación.

### **1.8.4 Pruebas de picos (Spike Testing)**

La prueba de picos, como el nombre sugiere, trata de observar el comportamiento del sistema variando el número de usuarios, tanto cuando bajan, como cuando tiene cambios drásticos en su carga.

Como buena práctica, siempre es aconsejable disponer de un entorno de pruebas de rendimiento lo más parecido como se pueda al entorno de producción.

## CAPÍTULO 2

### ANÁLISIS DE REQUERIMIENTOS

#### 2.1 Información del gimnasio

Para el desarrollo del presente proyecto se realizó varias visitas al Gimnasio Absolut Gym. Dentro del cual se ha recopilado la siguiente información:

#### 2.2 Misión del gimnasio

Proporciona un ambiente positivo y estimulante que promueve la salud, *fitness* y bienestar a sus clientes mediante el entrenamiento adecuado, ejecutado por profesionales con experiencia y altamente cualificados. Esforzándose por la excelencia en todo momento, garantizando los mejores resultados, como entrenadores, atletas y la comunidad. (Absolut Gym, 2014)

#### 2.3 Visión del gimnasio

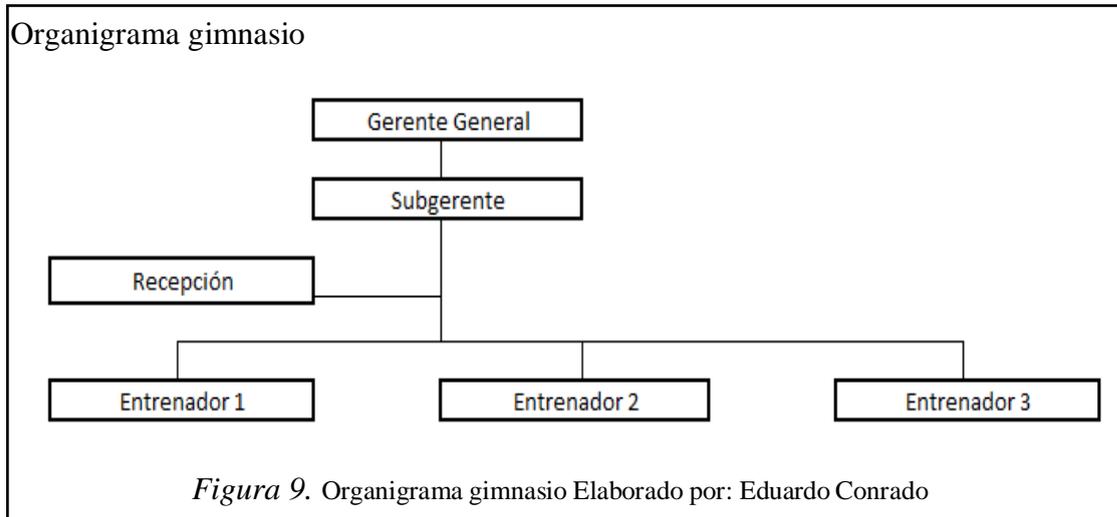
Absolut Gym proporciona los mejores programas de acondicionamiento físico de la ciudad, nuestra distinción se basa en programas personalizados para ayudar a que cada cliente logre sus objetivos individuales de salud, bienestar y *fitness*. (Absolut Gym, 2014)

#### 2.4 Valores

- Integridad: prescribimos ejercicios que creemos que tienen una base científica sólida y que proporcionan el mayor beneficio con el menor riesgo posible para el cliente.
- Habilidad: ofrecemos la mejor calidad en todo lo que hacemos.
- Responsabilidad: asumimos la responsabilidad de nuestros compromisos, y nos comprometemos a ayudar a nuestros clientes en sus objetivos
- Innovación: somos creativos en la implementación de nuestros servicios.

- Repetitividad: seguimos sistemas progresivos y probados de entrenamiento personal que garanticen resultados.

## 2.5 Organigrama gimnasio



Actualmente el Gimnasio está integrado por profesionales como: gerente, sub gerente, secretaria - recepcionista, entrenadores físicos y médicos deportólogos.

## 2.6 Infraestructura del gimnasio

El gimnasio cuenta con un área principal donde se encuentran aparatos totalmente equipados, como pesas, pesas integradas, isolaterales, spinning, trotadoras. Los espacios son adecuados especialmente para cada una de las diferentes secciones que conforman el gimnasio con equipos para cada grupo muscular. El gimnasio cuenta con área médica equipado para realizar diversos tipos de procedimientos como: valoraciones y evaluaciones.

## 2.7 Procesos del gimnasio

El gimnasio realiza los siguientes procesos:

- Registro de clientes
- Registro de valoración física inicial del cliente

- Registro de evaluaciones de aptitud física del cliente
- Registro de control de peso para clientes
- Registro de planes de entrenamiento físico personalizadas del cliente
- Impresión de facturas

El rol del gimnasio es dar asistencia en la práctica del deporte. Los instructores del gimnasio son los encargados del entrenamiento físico y cuidado de la salud de los clientes.

## **2.8 Análisis de la información**

### **2.8.1 Documento utilizado en el proceso de valoración física**

#### **2.8.1.1 Formulario para valoración física**

Contiene la información que especifica la condición física y médica del cliente al asistir a una valoración física en el gimnasio. (Anexo 1)

##### **2.8.1.1.1 Objetivo**

Es mantener un registro sistemático de los datos recopilados de diagnóstico, tratamiento y evolución de los problemas físicos y de salud.

##### **2.8.1.1.2 Motivo de valoración**

Se describe la causa que origina la solicitud de la valoración

### **2.8.1.1.3 Antecedentes familiares**

Se registra si existen los antecedentes definidos, registrar los problemas de salud relevantes de los miembros de la familia.

Tabla 1.  
*Antecedentes Personales*

1	Cardiopatía
2	Diabetes
3	Enfermedad Cerebro Muscular
4	Hipertensión
5	Cáncer
6	Tuberculosis
7	Enfermedad Mental
8	Enfermedad Infecciosa
9	Mal Formación
10	Otros

Nota. Elaborado por: Eduardo Conrado

### **2.8.1.1.4 Enfermedad o problema actual**

Se registra el resultado organizado del interrogatorio de síntomas acerca del problema encontrado en la valoración.

### **2.8.1.1.5 Signos vitales y antropometría**

Fecha: en el formulario se anota la fecha de atención de primera vez y de las atenciones subsecuentes respectivas.

Signos: en el formulario se anota los valores obtenidos sobre temperatura, presión arterial, pulso, frecuencia respiratoria y peso / talla.

Tabla 2.  
*Signos Vitales y Antropometría*

Temperatura °C		Sistólica	Presión Arterial		Diastólica
Presión Arterial			Presión Arterial		
Pulso / min			Frecuencia Respiratoria		
Peso / kg			Talla / cm		

Nota. Elaborado por: Eduardo Conrado

Los signos vitales serán registrados en la previa valoración física por la enfermera y verificados por el médico durante la valoración física.

#### **2.8.1.1.6 Área físico regional**

En el formulario se registra si se encuentra evidencia de patología en las opciones de examen señaladas en el documento.

Tabla 3.  
*Área Físico Regional*

<b>Ord</b>	<b>Descripción</b>
1	Cabeza
2	Cuello
3	Tórax
4	Abdomen
5	Pelvis
6	Extremidades

Nota. Elaborado por: Eduardo Conrado

### 2.8.1.1.7 Diagnóstico

En el formulario se escribe el diagnóstico indicando si son presuntivos o definitivos.

Tabla 4.  
*Diagnóstico*

Ord	Diagnóstico
1	
2	

Nota. Elaborado por: Eduardo Conrado

### 2.8.1.1.8 Planes de tratamiento

- Se registra el conjunto organizado de solicitudes de exámenes complementarios de laboratorio clínico, imagenología y otros.
- Si se necesita, se registra pedido de visita a especialistas a centros de salud.

### 2.8.1.1.9 Planes de entrenamiento

- Se registra en el documento los ejercicios físicos de acuerdo a la condición física y de salud, especificando la parte del cuerpo a la que se enfoca el entrenamiento.
- Se registra las recomendaciones o indicaciones acerca de la nutrición y estilos de vida saludable.

### 2.8.1.1.10 Evolución del estado físico

En el documento se registra una secuencia del estado y progreso físico del cliente, variaciones del entrenamiento y recomendaciones realizadas por el profesional responsable del cliente.

### 2.8.1.1.11 Evolución

Fecha – notas: anotar la fecha (día / mes / año)

En el documento se describe el progreso en el estado físico, entrenamiento y nuevos datos de valoración física.

## **2.9 Detalle del funcionamiento de los procesos en la aplicación web**

De acuerdo a la información obtenida en el gimnasio, se puede detallar, que la aplicación web es un software que va a ser utilizado solo por el personal del gimnasio. Es una aplicación interna con acceso de acuerdo a un perfil de usuarios. El personal gestionará las valoraciones físicas y planes de entrenamiento de sus clientes. Cada dato almacenado permitirá la organización del historial de cada cliente con su información personal, y el seguimiento de su entrenamiento. En el momento de la valoración física se toma los signos vitales, esto les sirve al entrenador y al médico del gimnasio para evaluar la condición del cliente. Los datos pueden ser modificados en cualquier momento en valoraciones físicas y médicas posteriores. El sistema web permitirá realizar la búsqueda del historial del cliente de una forma más ágil.

### **2.9.1 Proceso de asignación de valoraciones físicas y médicas**

El proceso de asignación de valoraciones es el siguiente:

- Al ver las citas del día, se le delega al médico o entrenador físico su cita (si él se encuentra en el gimnasio).
- El médico o entrenador físico le dice a la recepcionista (por medio del estado “en espera”) que está listo para atender al cliente.
- El médico o entrenador físico finaliza la cita y se alista para la siguiente, donde podrá su estado de “en espera”.

En cada valoración se mide la evolución de la condición tomando los signos vitales y revisando la masa muscular de acuerdo a los planes de entrenamiento. El entrenador físico describe el nuevo plan de entrenamiento de acuerdo a cada cliente.

### **2.9.2 Cobro de servicios**

Para el cobro de servicios existirá una opción para imprimir el valor de la valoración física o servicio ofrecido al cliente, la recepcionista es el usuario que se encargará de imprimir la factura.

### **2.9.3 Reportes**

- Clientes atendidos por entrenador físico por día o por lapsos de tiempo variables
- Clientes atendidos por médico por día o por lapsos de tiempo variables
- Total de ingresos por entrenador físico por día ( o por lapsos de tiempo variables)
- Total de ingresos por medico por día ( o por lapsos de tiempo variables)
- Total ingresos por día ( o por lapsos de tiempo variables)

## CAPÍTULO 3

### DISEÑO DEL SOFTWARE

#### 3.1 Diseño de la interface basado en el usuario

La metodología de diseño centrado en el usuario (DCU) está planteada para establecer las características para diseñar un sitio web. DCU tiene como objetivo generar sensaciones y valoraciones de los usuarios hacia el sitio web lo más agradables, positivas y satisfactorias posibles, además de reseñar la fidelidad del usuario como consecuencia de alcanzar este objetivo. (DNX, 2005)

En el contexto de la Web, Morville (2004) propone el análisis de la Experiencia del usuario es base a siete facetas o propiedades que debe cumplir un sitio web:

- Útil
- Usable
- Deseable
- Localizable
- Accesible
- Creíble
- Valioso

Diseño de la interface basado en el usuario



*Figura 10.* Diseño de la interface basado en el usuario Fuente:(DNX, 2005)

El diseño centrado en el usuario (DCU), como filosofía de diseño, engloba o se relaciona con un heterogéneo conjunto de metodologías y técnicas que comparten un objetivo común: conocer y comprender las necesidades, limitaciones, comportamiento y características del usuario, involucrando en muchos casos a usuarios potenciales o reales en el proceso.

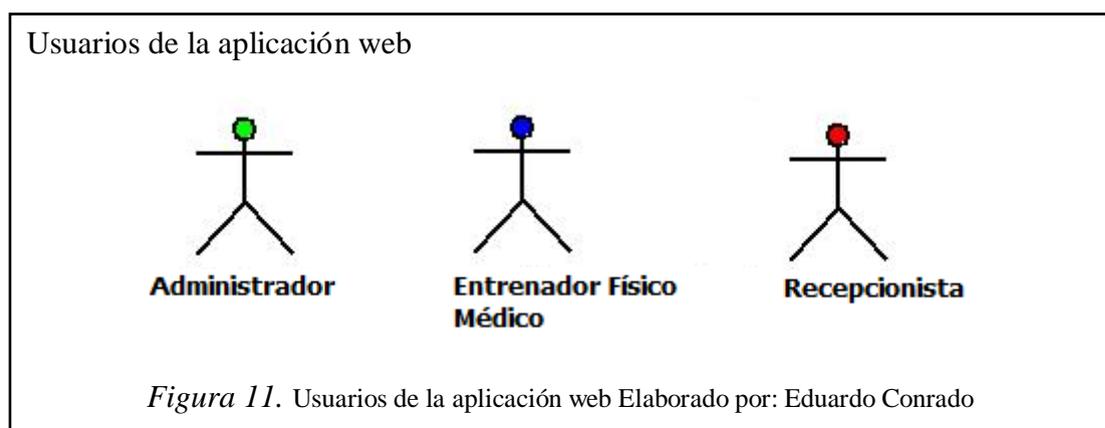
Fases metodológicas de la metodología

- Definición: identificación de usuarios
- Análisis: analizar las necesidades de los usuarios
- Diseño: diseñar y evaluar las interfaces
- Evaluación: evaluar el proceso

### 3.1.1 Identificación de usuarios

Los usuarios de aplicación web son:

- Administrador
- Entrenador físico
- Medico
- Personal de Administración
- Recepcionista



### **3.1.1.1 Descripción de las tareas de los usuarios**

#### **3.1.1.1.1 Administrador**

El actor tendrá acceso a todos los procesos de la aplicación. Con cada panel podrá crear, borrar, editar y ver todas las funciones de cada página. Se aplican privilegios para gestionar usuarios, clientes, valoraciones físicas, asignación de valoraciones físicas, atención de la valoración física, generación de notas de evolución, planes de entrenamiento y generación de reportes.

#### **3.1.1.1.2 Entrenador físico**

El actor manejará la información de los clientes, visualizará solo sus asignaciones de valoraciones físicas, valoración físicas médicas, planes de entrenamiento, notas de evolución y también mostrara determinados reportes.

#### **3.1.1.1.3 Médico**

El actor manejará la información de los clientes, visualizará solo sus asignaciones de valoraciones médicas, nutrición, peso, notas de evolución y también mostrará determinados reportes.

#### **3.1.1.1.4 Personal administrativo**

Es el que visualiza, o edita determinados procesos específicos. Tendrá acceso a determinados reportes.

#### **3.1.1.1.5 Recepcionista**

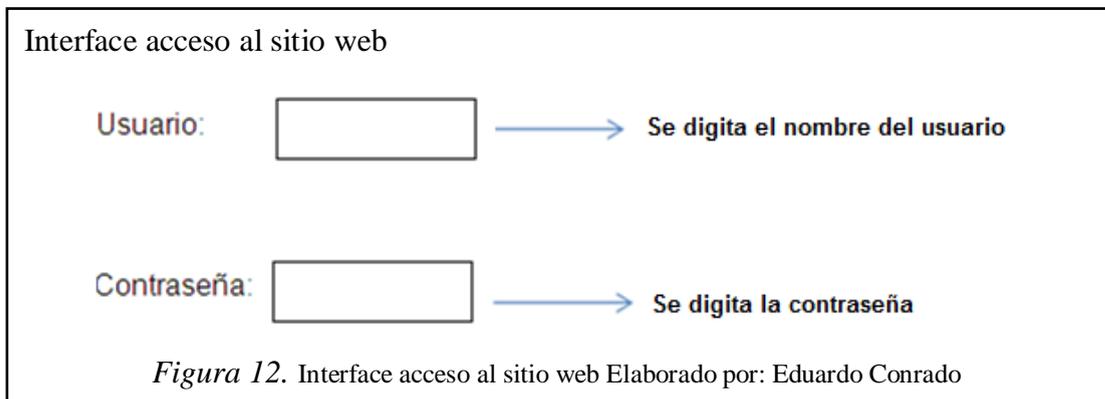
Es la que maneja la información de valoraciones físicas, recibe información de los clientes, y asignará valoraciones físicas a entrenadores físicos o médico.

### 3.1.2 Análisis de las necesidades de los usuarios

- a) Observación
- b) Investigación
- c) Indagación del usuario
- d) Técnicas
  - Test de usuarios
  - Evaluación heurística
  - *Card Sorting*
  - *Eye-Tracking*
  - Observación participante
  - Entrevistas
  - Encuestas
  - Analítica web

### 3.1.3 Diseño y evaluación de las interfaces

#### 3.1.3.1 Interface acceso al sitio web



En la figura 12, la interface permitirá realizar el control de ingreso al sitio web, a través de validación de usuario y contraseña

### 3.1.3.2 Interface crear usuario

Interface crear usuario

Nombre:	<input type="text"/>
Apellido:	<input type="text"/>
Email:	<input type="text"/>
Teléfono:	<input type="text"/>
Celular:	<input type="text"/>
Cargo:	<input type="text"/>
Perfil:	<input type="text"/> ↓
Nombre de Usuario:	<input type="text"/>
Contraseña:	<input type="text"/>

*Figura 13.* Interface crear usuario Elaborado por: Eduardo Conrado

En la figura 13, la interface permite ingresar los datos del usuario. Con estos datos se podrá realizar el proceso acceso al sitio web.

### 3.1.3.3 Interface ver/editar/borrar usuario

Interface ver/editar/borrar usuario

ID	Nombre	Apellido	Email	Cargo	Teléfono	
1	Luis	Duarte		Entrenador	22334455	Ver/Editar - Borrar
2	Felipe	Ballesteros		Recepcionista	22667788	Ver/Editar - Borrar
3	Juan	Durán		Sistemas	22887744	Ver/Editar - Borrar

*Figura 14.* Interface ver/editar/borrar usuario Elaborado por: Eduardo Conrado

En la figura 14, la interface permitirá visualizar a la lista de usuarios del sistema web.

### 3.1.3.4 Interface crear cliente

Interface crear cliente

<b>Id</b>	<input type="text" value="automático"/>
Nombre:	<input type="text"/>
Apellido:	<input type="text"/>
Cédula:	<input type="text"/>
Sexo:	<input type="text"/>
Fecha de Nacimiento:	<input type="text"/>
Edad:	<input type="text"/>
Fecha de Ingreso:	<input type="text"/>
Teléfono:	<input type="text"/>
Celular:	<input type="text"/>
Email:	<input type="text"/>
Persona de Contacto:	<input type="text"/>
Teléfono Contacto:	<input type="text"/>

*Figura 15.* Interface crear cliente Elaborado por: Eduardo Conrado

En la figura 15, la interface facilita el ingreso de la información personal del cliente. Datos que servirán para identificar a cada cliente. Además de contactos para informar de alguna información que se requiera transmitir al cliente o familiares.

### 3.1.3.5 Interface antecedentes personales

Interface antecedentes personales

Presencial	
<b>Antecedentes Personales:</b>	<input type="checkbox"/> Cardiopatía
	<input type="checkbox"/> Diabetes
	<input type="checkbox"/> Enf. C. Muscular
	<input type="checkbox"/> Hipertensión
	<input type="checkbox"/> Cáncer
	<input type="checkbox"/> Tuberculosis
	<input type="checkbox"/> Enf. Mental
	<input type="checkbox"/> Enf. Infecciosa
	<input type="checkbox"/> Mal Formación
	<input type="checkbox"/> Otros
<b>Comentarios/Diagnóstico:</b>	

*Figura 16.* Interface antecedentes personales Elaborado por: Eduardo Conrado

En la figura 16, la interface permite identificar y seleccionar a cada uno de los antecedentes personales que puede tener un cliente al momento de ingresar sus datos personales a la aplicación.

### 3.1.3.6 Interface tipo entrenamiento



En la figura 17, la interface permite identificar y seleccionar a cada uno de los tipos de entrenamiento que puede tener un cliente al momento de ingresar datos personales a la aplicación.

### 3.1.3.7 Interface pantalla principal

Interface pantalla principal

ID	Nombre	Apellido	Cédula	Edad	Teléfono	Celular	Email	
1	Andrés	Duarte	17922860	25	6003726	6178		Ver/Editar - Borrar
2	Juan	Villafuerte	17922860	32	2345676	7654		Ver/Editar - Borrar
3	Ignacio	Zambrano	17922860	34	2345342	8773		Ver/Editar - Borrar

*Figura 18.* Interface pantalla principal Elaborado por: Eduardo Conrado

En la figura 18, la interface permite visualizar a los clientes en la lista de espera de atención de valoración física.

### 3.1.3.8 Estados de la cita

Tabla 5.  
*Estados de la cita*

<b>F</b>	<b>Finalizado</b>
<b>E</b>	<b>En Espera</b>
<b>C</b>	<b>Cancelado</b>
<b>V</b>	<b>Visita</b>
<b>L</b>	<b>Llegó</b>

Nota. Elaborado por: Eduardo Conrado

En la tabla 5, la lista anterior describe los diferentes estados que puede tener una valoración física antes, durante y después de ser atendida por el entrenador o médico.

### 3.1.3.9 Interface valoración física

Interface crear valoración física

Fecha	Calendario
Hora:	Módulo de Hora
Cliente	Búsqueda - Relacionar
Especialista	Lista Desplegable
Atendido Por:	Recepcionista

*Figura 19.* Interface crear valoración física Elaborado por: Eduardo Conrado

En la figura 19, la interface permite ingresar una valoración física a ser atendida por un especialista (entrenador físico o médico). En esta interface se mostraran listas desplegables que evitarán cometer errores de digitación por parte del usuario.

### 3.1.3.10 Interface signos vitales

Interface signos vitales

Datos de Control:		
Control	Valor	Fecha
Temperatura °C:		<b>Autom</b>
Presión Arterial Sistólica:		
Presión Arterial Diastólica:		
Pulso / min:		
Peso / kg:		
Frecuencia Respiratoria:		
Talla / cm:		

**Guardar**

*Figura 20.* Interface signos vitales Elaborado por: Eduardo Conrado

En la figura 20, la interface permite ingresar los valores correspondientes al estado de salud del cliente en el momento de ingresar a la valoración física con el entrenador físico o médico.

### 3.1.3.11 Interface generar cobro

Interface generar cobro

IdCobro	Descripción	Observación	Valor
1	Valoración Física		10
3	Nutricionista		10
<b>Valor Total</b>			20

**Generar Cobro**

*Figura 21.* Interface generar cobro Elaborado por: Eduardo Conrado

En la figura 21, la interface permite ingresar los diferentes tipos de cobro de acuerdo a la atención brindada al cliente.

### 3.1.3.12 Interface generar factura

Interface generar factura

**Generar Factura**

Id Factura: Autom  
Cliente :  
Fecha: Autom  
Hora: Autom

IdCobro	Descripción	Observación	Valor
1	Valoración Física		10
3	Nutricionista		10
<b>Valor Total</b>			<b>20</b>

**Generar Cobro**

*Figura 22.* Interface generar factura Elaborado por: Eduardo Conrado

En la figura 22, la interface permite visualizar el formato de impresión de la factura. Cada línea describe el cobro. La última línea calcula automáticamente el valor total de la factura.

### 3.1.3.13 Interface ingresar valoración física

Interface ingresar valoración física

**Ingreso de Valoración Física**

Id usuario **Autom = ID** (automático)  
Fecha (automático)  
Motivo de la valoración  
Descripción del problema físico / salud

*Figura 23.* Interface ingresar valoración física Elaborado por: Eduardo Conrado

En la figura 23, el entrenador físico o médico mediante esta interface ingresa la información principal de la Valoración Física. El motivo de la Valoración Física y la descripción del problema actual físico / salud.

### 3.1.3.14 Interface área físico regional

Interface área físico regional

Orden	Descripción	(selección)
1	Cabeza	
2	Cuello	X
3	Tórax	
4	Abdomen	X
5	Pelvis	
6	Extremidades	

*Figura 24.* Interface área físico regional Elaborado por: Eduardo Conrado

En la figura 24, mediante esta interface el entrenador físico o médico selecciona el o las áreas problemas del cuerpo del cliente. Mediante la selección de las diferentes opciones se escoge las diferentes partes del cuerpo del cliente.

### 3.1.3.15 Interface diagnóstico

Interface diagnóstico

Orden	Descripción	Diagnóstico
1		
2		
3		
4		

*Figura 25.* Interface diagnóstico Elaborado por: Eduardo Conrado

En la figura 25, esta interface permite visualizar el o los posibles diagnósticos encontrados al cliente. Presuntivamente se puede anotar un pre diagnóstico para confirmarlo en una posterior visita al entrenador físico o médico con los resultados de los exámenes médicos enviados a realizarse.

### 3.1.3.16 Interface plan de tratamiento

Interface plan de tratamiento

Orden	Tratamiento - Observaciones
1	Diagnostico: Dolor muscular Prescripción: Ibuprofeno cada 6 horas Terapia Física: compresas con agua

**Guardar**

*Figura 26.* Interface plan de tratamiento Elaborado por: Eduardo Conrado

En la figura 26, a través de esta interface el entrenador físico o médico ingresa el tratamiento de acuerdo a la valoración física. Queda almacenada la información de medicamentos y/o terapia física según sea el caso.

### 3.1.3.17 Interface plan de entrenamiento

Interface plan de entrenamiento

Orden	Entrenamiento - Observaciones
1	Rutinas de gimnasio para pecho: Press de banca. Rutinas para pecho: Press de banca, banca inclinada, banca declinada, Pectoral contractor

**Guardar**

*Figura 27.* Interface plan de entrenamiento Elaborado por: Eduardo Conrado

En la figura 27, a través de esta interface el entrenador físico o médico ingresa el plan de entrenamiento de acuerdo a la valoración física.

### 3.1.3.18 Interface reportes del sistema web

Interface reportes del sistema web

Filtro	Inicial	Final	Entrenador	Nutricionista
Cientes atendidos por Entrenador	12/06/2014	20/06/2014	Andres Naveda	
Cientes atendidos por Médico	13/06/2014	30/06/2014		Mario Perez

*Figura 28.* Interface reportes del sistema web Elaborado por: Eduardo Conrado

En la figura 28, mediante esta interface se puede visualizar los diferentes tipos de reportes que puede generar la aplicación web.

### **3.1.4 Evaluación del proceso**

Se presentó a los usuarios el esquema de las pantallas o interfaces a utilizarse en la aplicación web. Las interfaces facilitan la interactividad entre un usuario y la computadora mediante acciones amigables fáciles de entender y comprender. Su funcionamiento fue entendible por los usuarios. Como consecuencia se procedió a implementar el diseño de las interfaces.

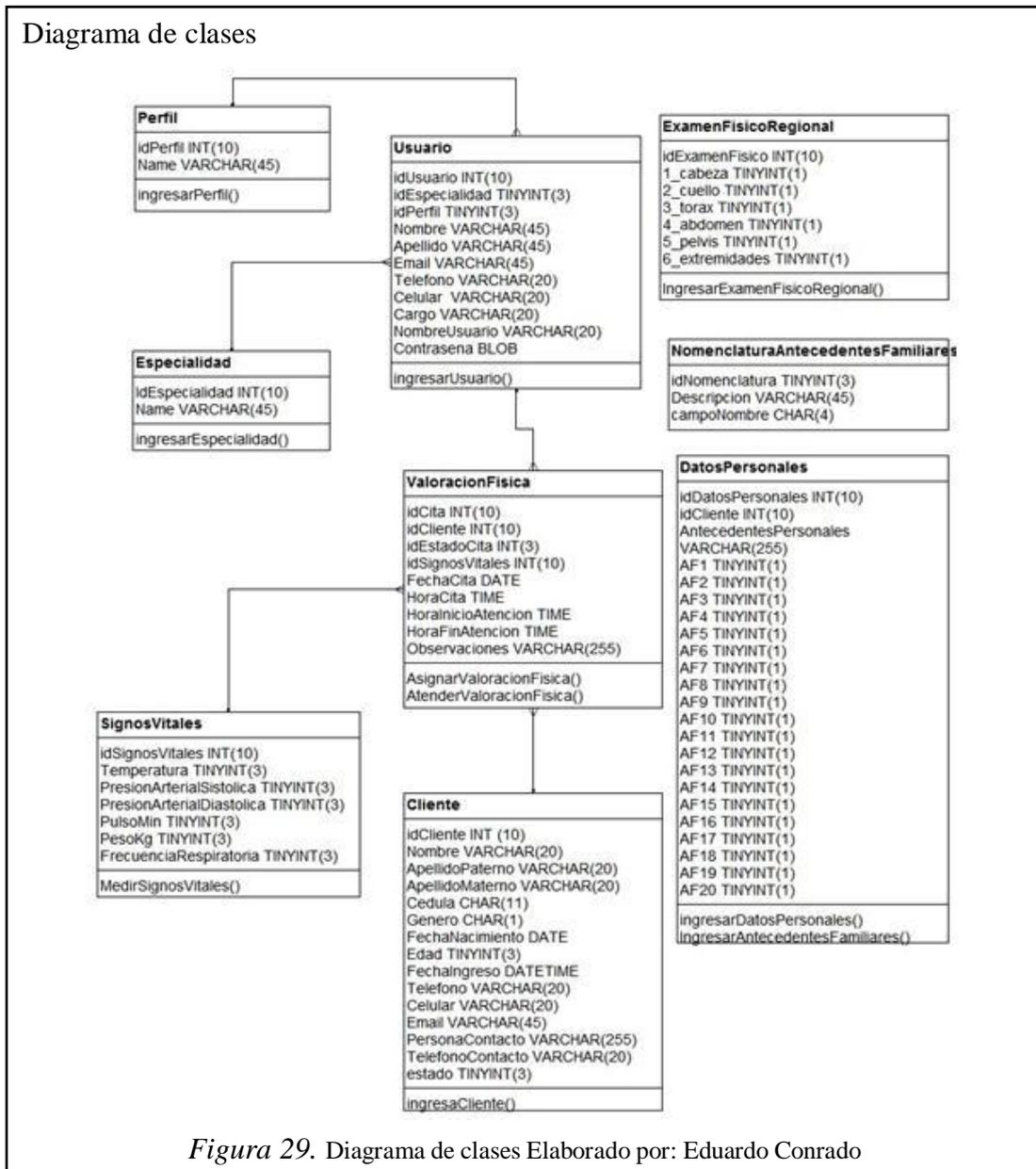
## **3.2 Modelamientos**

### **3.2.1 Modelos aplicados a cada proceso de aplicación web**

La siguiente lista resume los modelos aplicados a cada proceso de aplicación web:

- Diagrama de clases
- Diagrama de casos de uso
- Diagrama de estados
- Diagrama de secuencia
- Diagrama conceptual

### 3.2.1.1 Diagrama de clases



En la figura 29, se visualiza las 9 clases que son un conjunto de objetos o atributos de la clase y métodos u operaciones sobre los objetos de la clase.

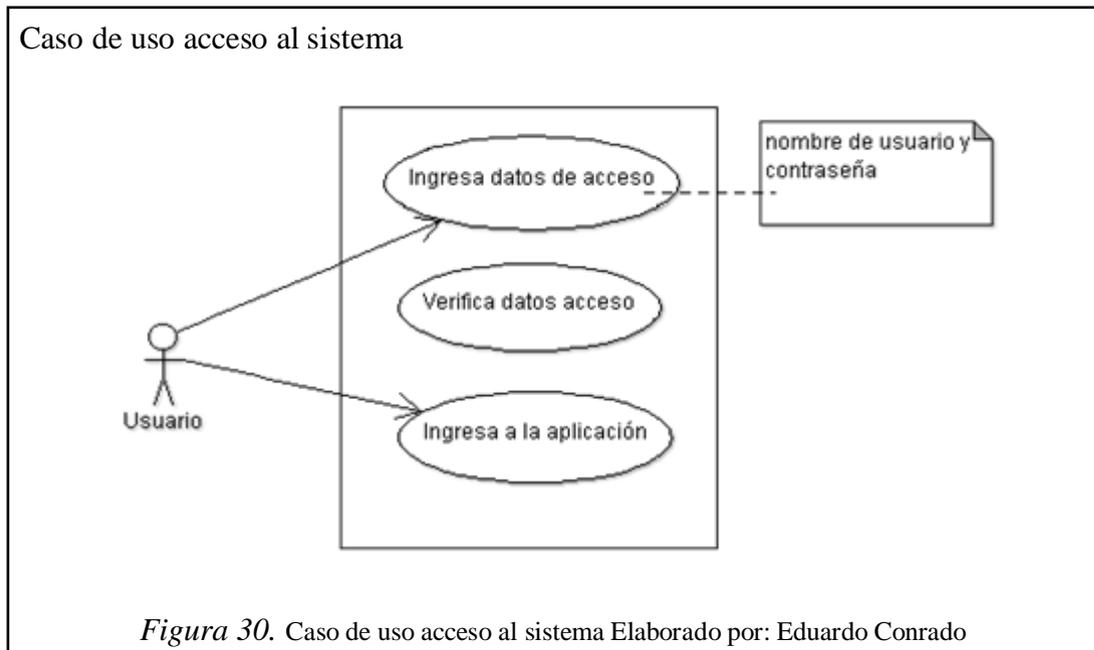
El aplicativo tiene tres clases primarias, donde contiene la información principal del aplicativo web: usuario, valoración física y cliente.

Dos clases secundarias, las cuales complementan a las clases primarias: signos vitales, y datos personales

Y cuatro clases catálogo, donde se consulta información predeterminada en el aplicativo: perfil, especialidad, examen físico regional y nomenclatura antecedentes familiares

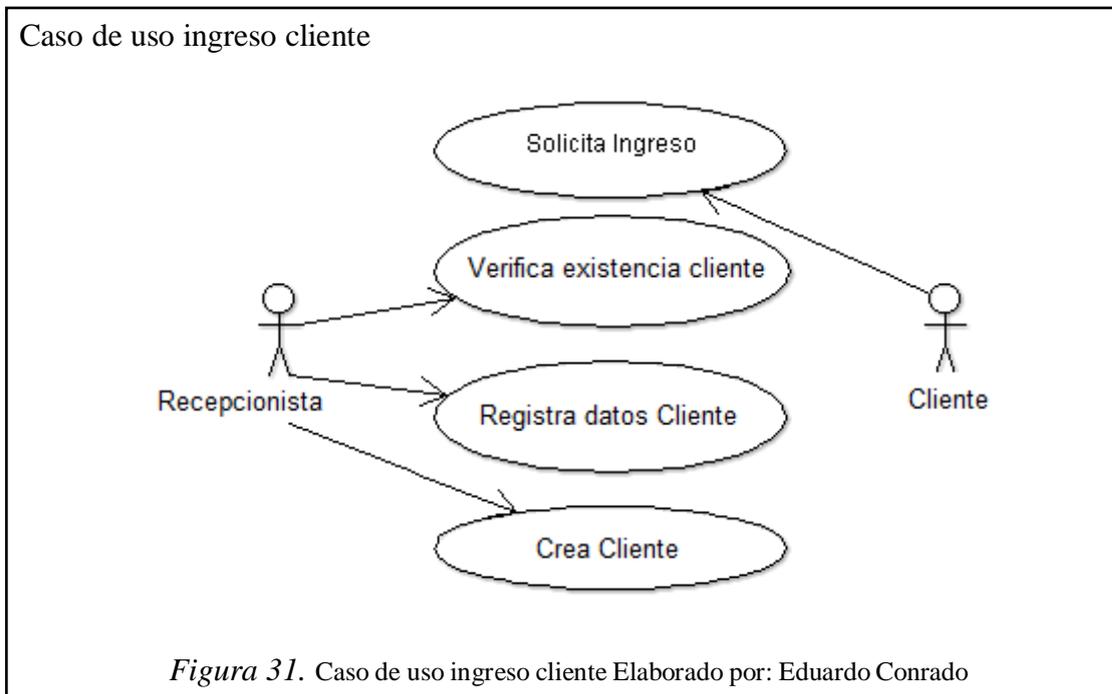
### 3.2.1.2 Diagramas casos de uso

#### a) Caso de uso - acceso al sistema



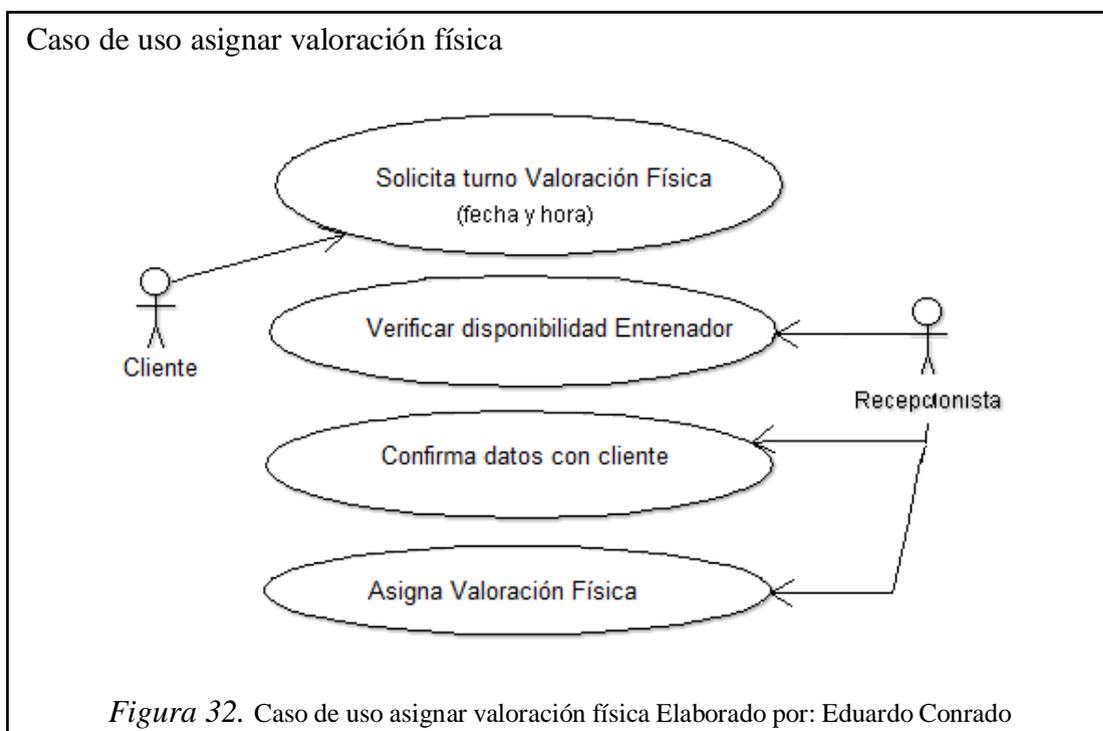
En la figura 30, se muestra al usuario del aplicativo web, con los diferentes escenarios al momento de acceso al sistema.

### b) Caso de uso - ingreso cliente



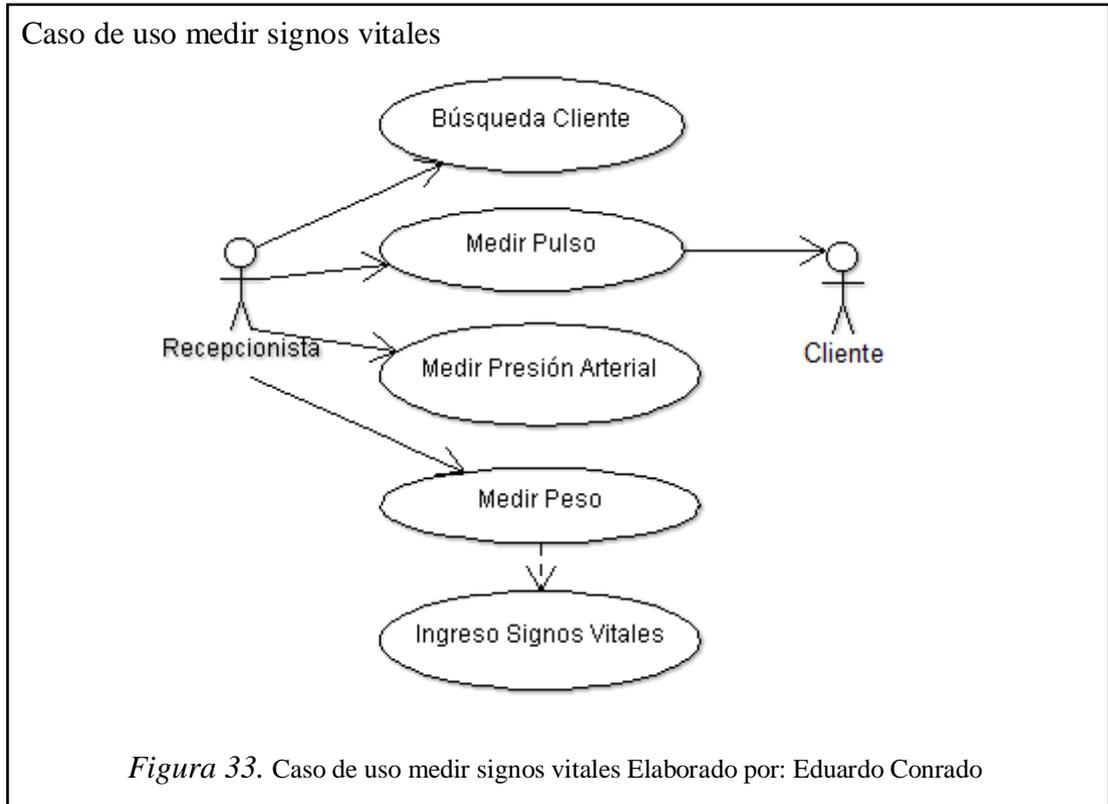
En la figura 31, muestra al cliente solicitando el ingreso al gimnasio, y al usuario recepcionista, verificando la existencia del cliente, registra los datos, o crea al cliente en el sistema.

### c) Caso de uso - asignar valoración física



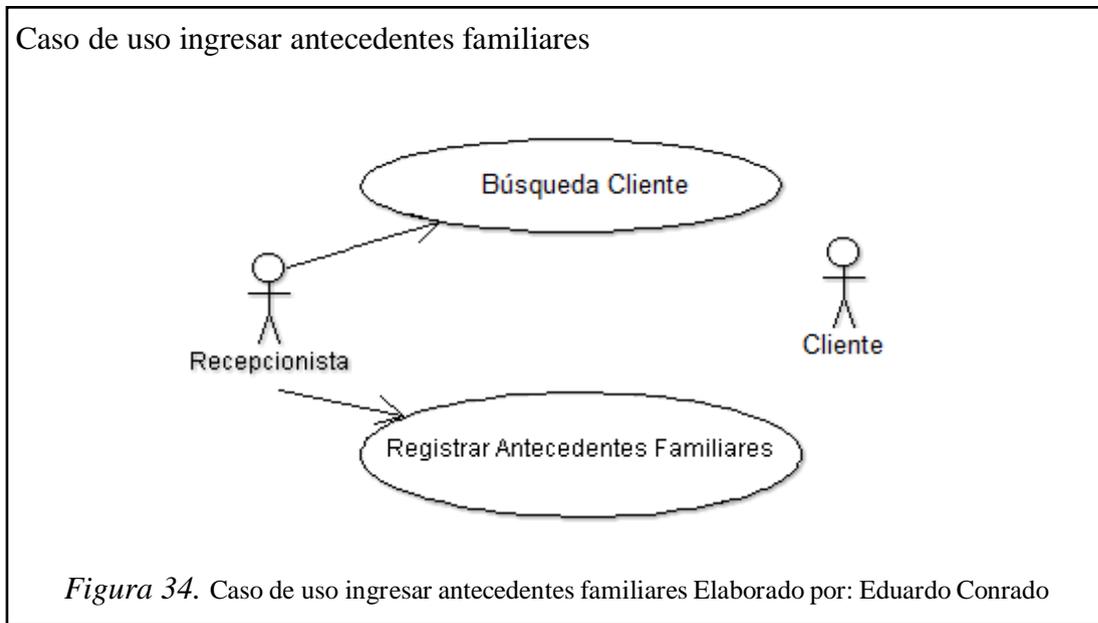
En la figura 3.23, el cliente solicita se le realice una valoración física, el usuario recepcionista, primero verifica disponibilidad del entrenador, después confirma en el sistema los datos del cliente, por último, asigna valoración física.

**d) Caso de uso - medir signos vitales**



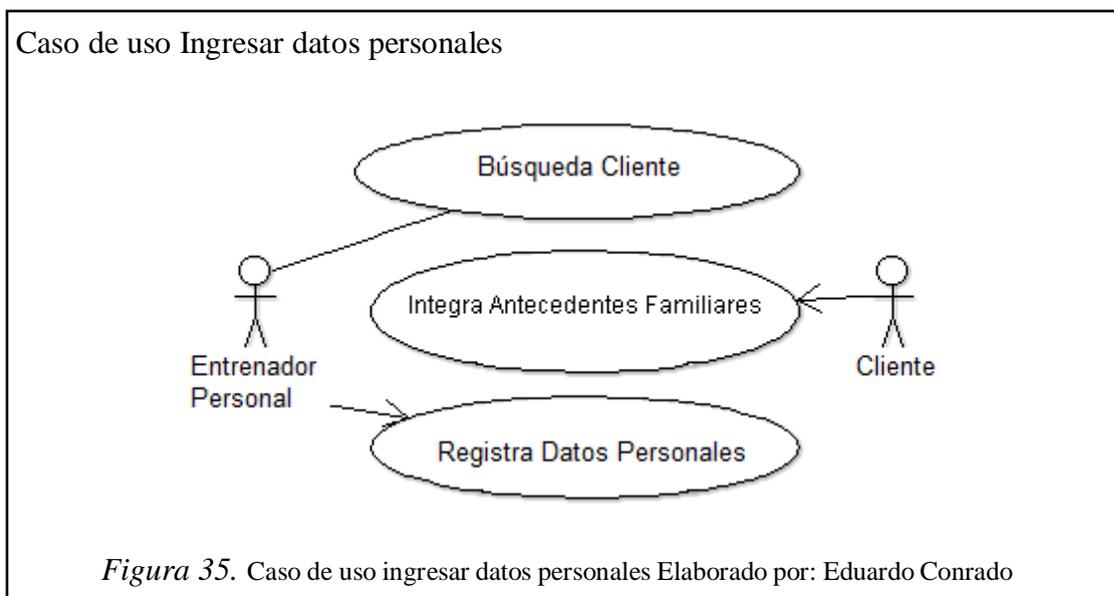
En la figura 33, el usuario recepcionista, busca al cliente en el sistema; después mide pulso, presión arterial, peso; por último, ingresa los valores en el sistema.

**e) Caso de Uso - Ingresar antecedentes familiares**



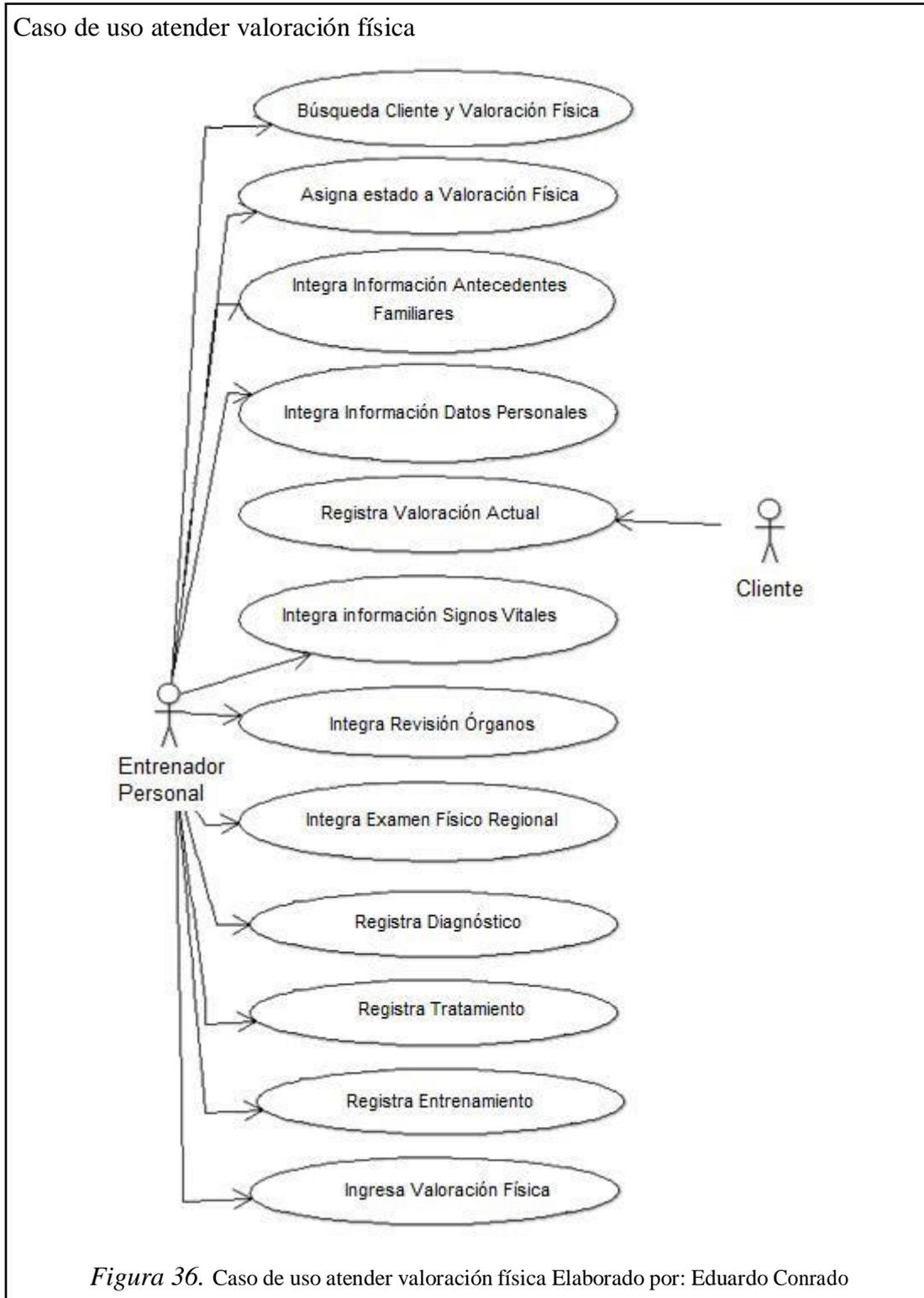
En la figura 34, el usuario recepcionista, busca al cliente en el sistema, después registra los antecedentes familiares del cliente en el sistema.

**f) Caso de uso - Ingresar datos personales**



En la figura 35, el usuario entrenador, busca al cliente en el sistema; el cliente integra antecedentes familiares, por último, el usuarios entrenador registra los datos personales.

**g) Caso de uso - atender valoración física**



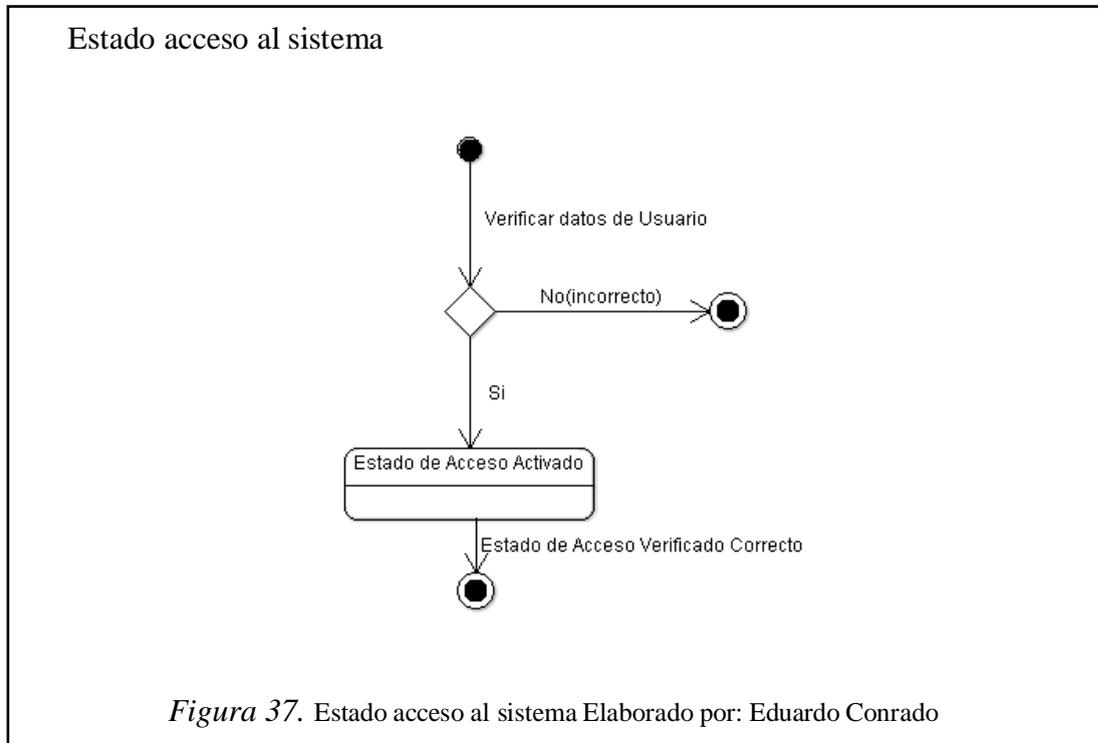
*Figura 36.* Caso de uso atender valoración física Elaborado por: Eduardo Conrado

En la figura 36, el usuario entrenador, busca al cliente e integra los datos obtenidos del sistema; datos personales, signos vitales, revisión de órganos, examen físico

regional; cliente registra la valoración actual; después, el usuario entrenador registra; diagnóstico, tratamiento, entrenamiento; por último, ingresa la valoración física.

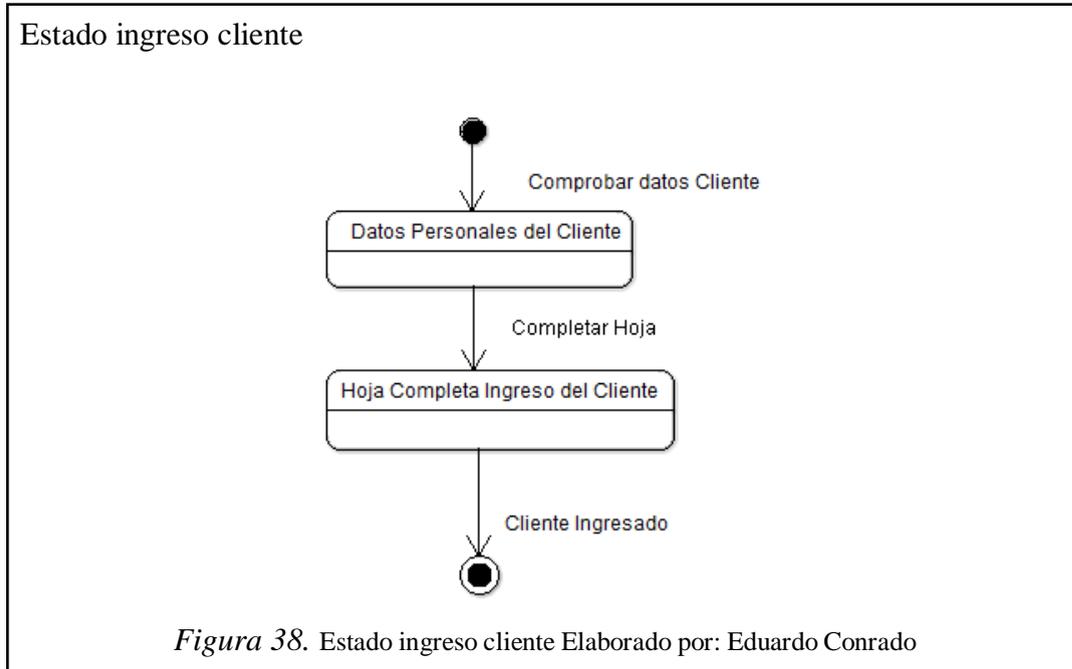
### 3.2.1.3 Diagramas de estado

#### a) Estado - acceso al sistema



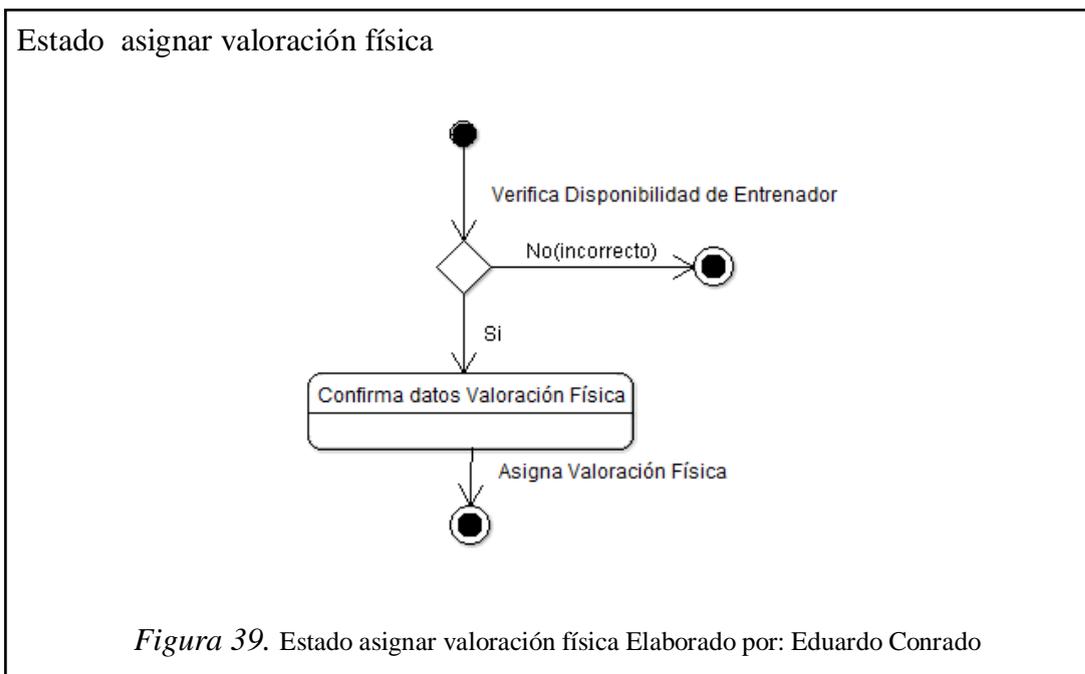
En la figura 3.29, el sistema verifica los datos de ingreso del usuario, No o incorrecto, regresa a la pantalla de ingreso; SI accede al sistema con las opciones de su perfil.

**b) Estado - ingreso cliente**



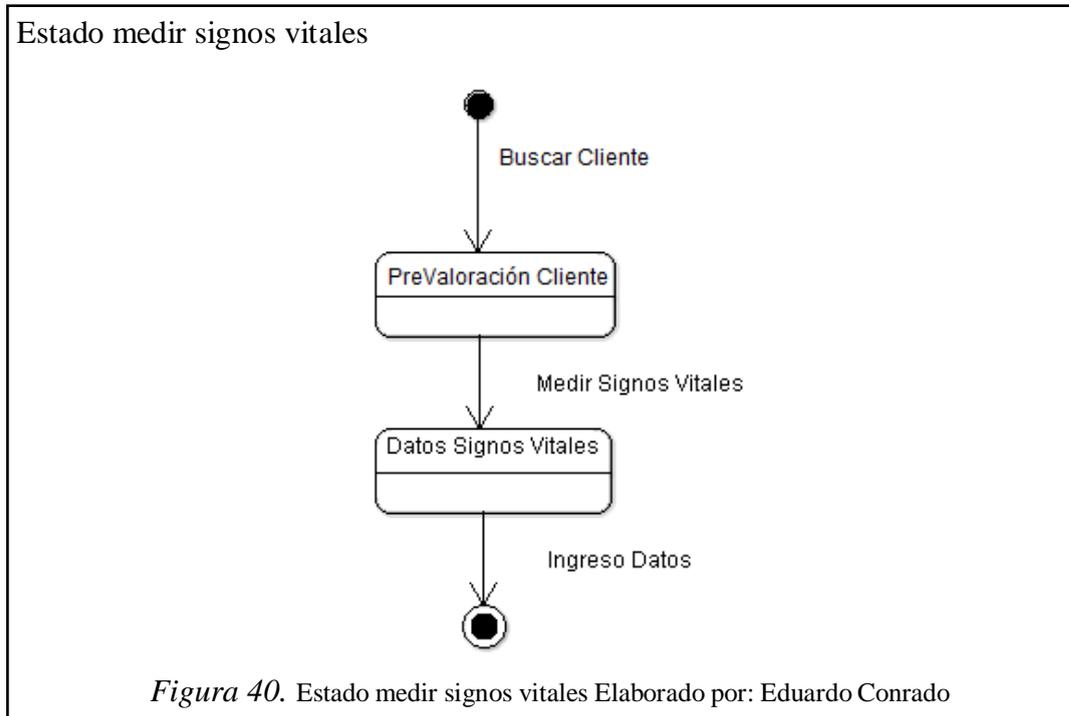
En la figura 38, se muestra el ingreso de los datos del cliente. Se comprueba si el cliente ya existe en el sistema, luego se pasa a la pantalla para el ingreso de los datos, por último, se muestra un mensaje de cliente ingresado.

**c) Estado - asignar valoración física**



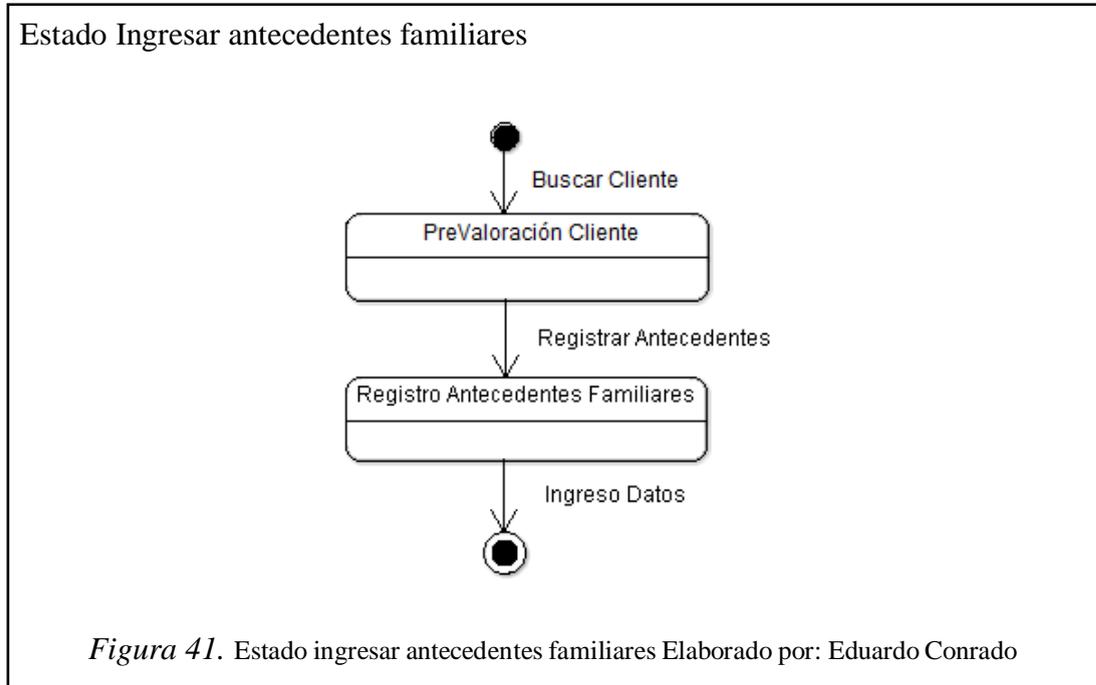
En la figura 39, el sistema verifica la disponibilidad del entrenador; SI, confirma lo datos y asigna la valoración física.

**d) Estado - medir signos vitales**



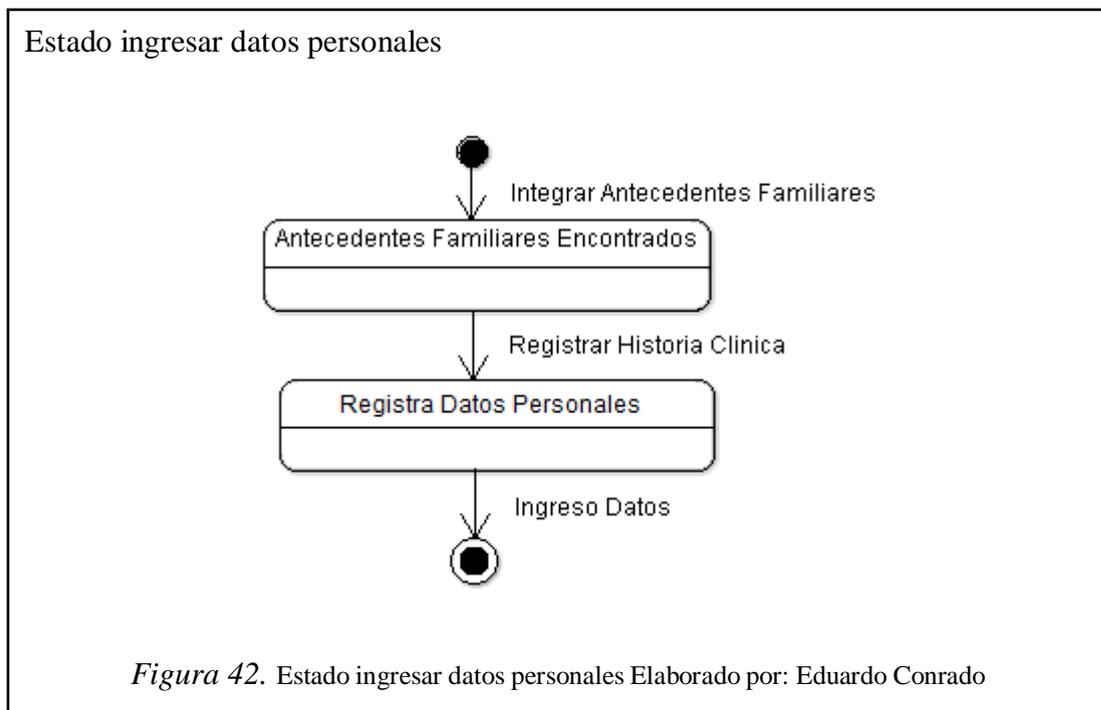
En la figura 40, el sistema busca al cliente; luego se mide los signos vitales del cliente; por último, se ingresa al sistema la información.

**e) Estado - ingresar antecedentes familiares**



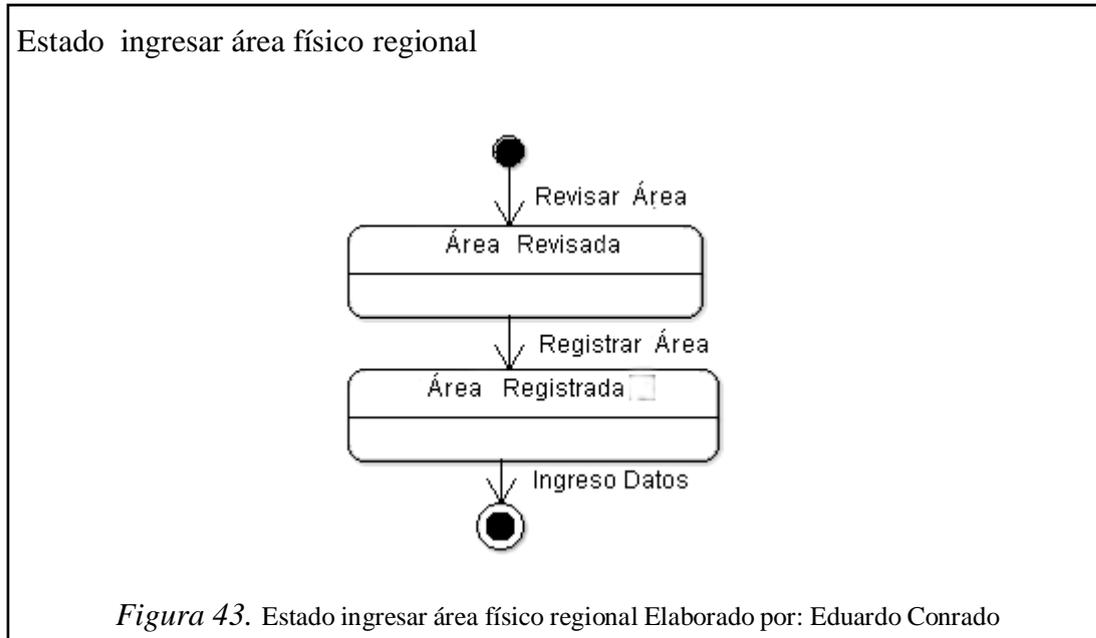
En la figura 41, el sistema busca al cliente; luego se registra los antecedentes familiares; por último se ingresa los datos al sistema.

**f) Estado- ingresar datos personales**



En la figura 42, el sistema integra los antecedentes familiares

**g) Estado - ingresar área física regional**



En la figura 43, se revisan los órganos del cliente; se registra si hay alguna actualización; por último se ingresa la información al sistema.

## h) Estado - atender valoración física

Estado atender valoración física

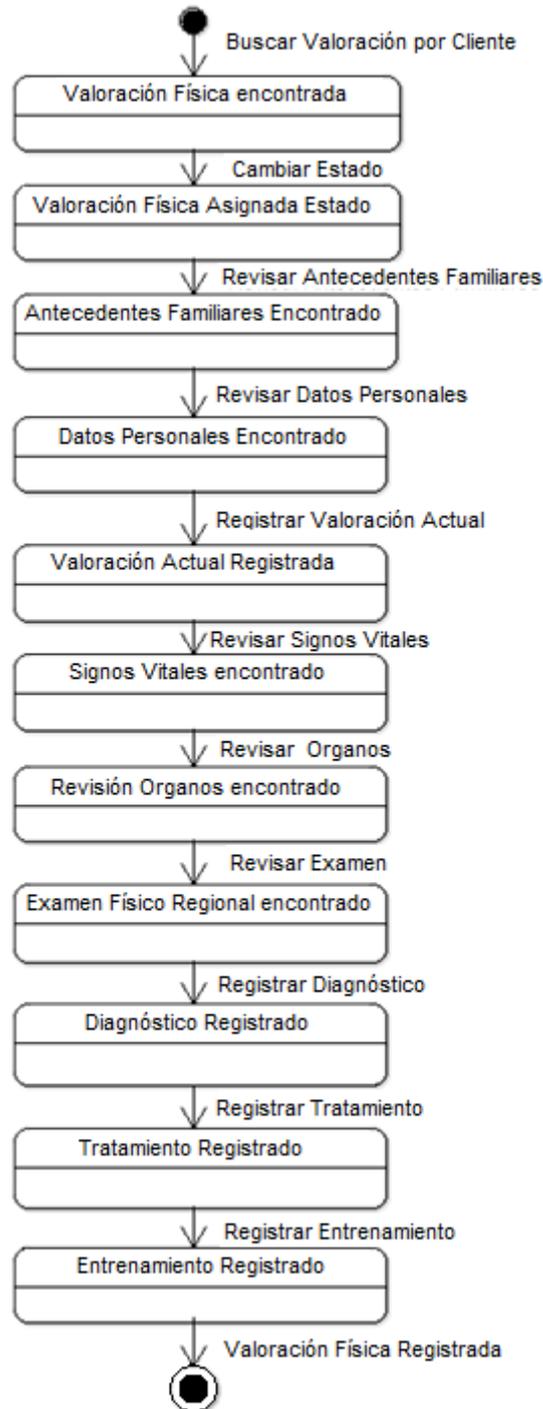
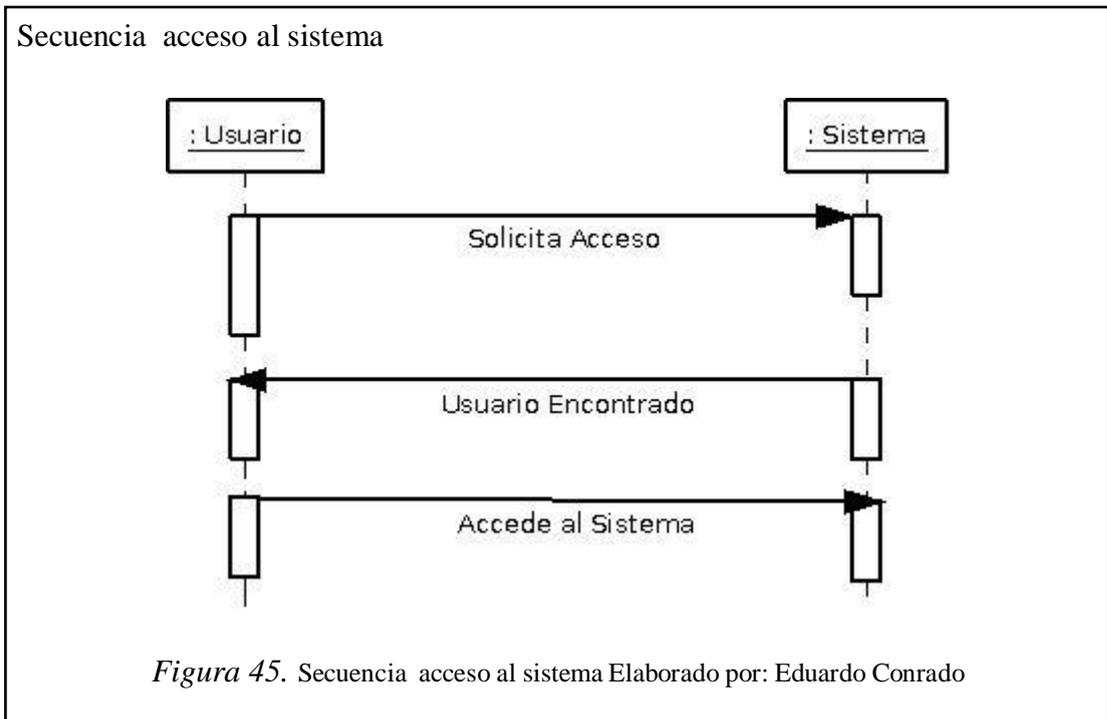


Figura 44. Estado atender valoración física Elaborado por: Eduardo Conrado

En la figura 44, el sistema busca la valoración física del cliente; el sistema la encuentra la valoración; cambia el estado; asigna el estado; se revisa antecedentes familiares, datos personales, valoración física actual, signos vitales, órganos, examen; después se registra el diagnóstico, tratamiento, entrenamiento: por último se registra en el sistema la valoración física final.

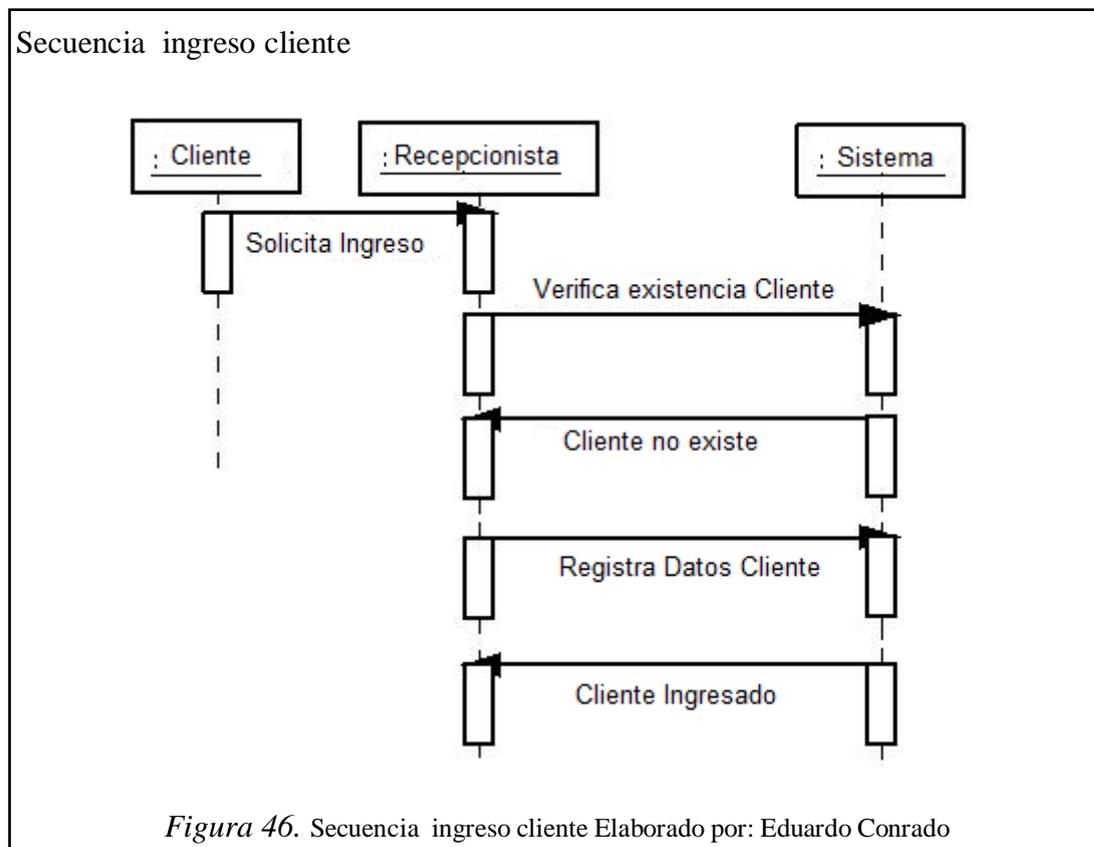
### 3.2.1.4 Diagramas de secuencia

#### a) Secuencia acceso al sistema



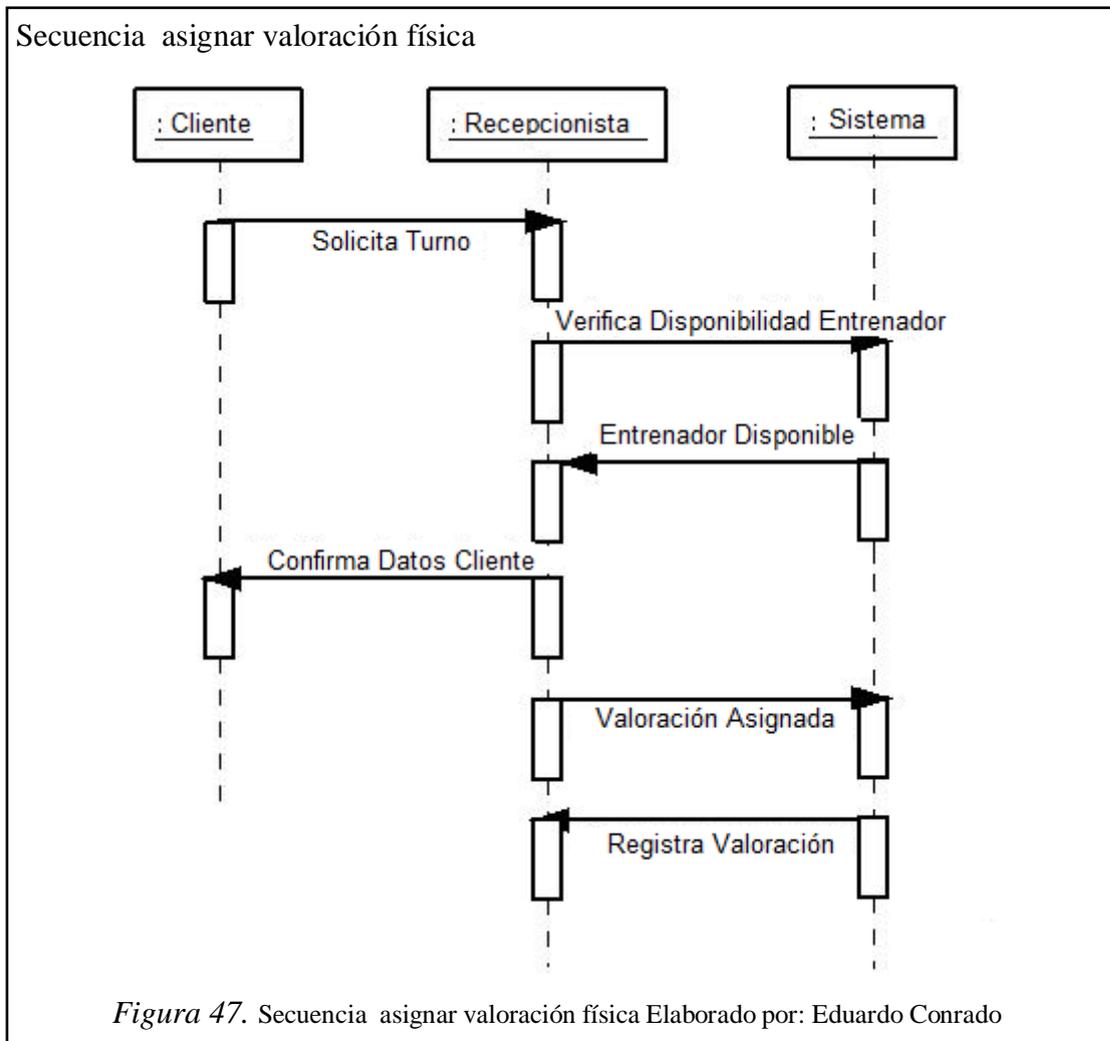
En la figura 45, usuario solicita acceso al sistema; sistema devuelve usuario encontrado; usuario accede al sistema.

## b) Secuencia ingreso cliente



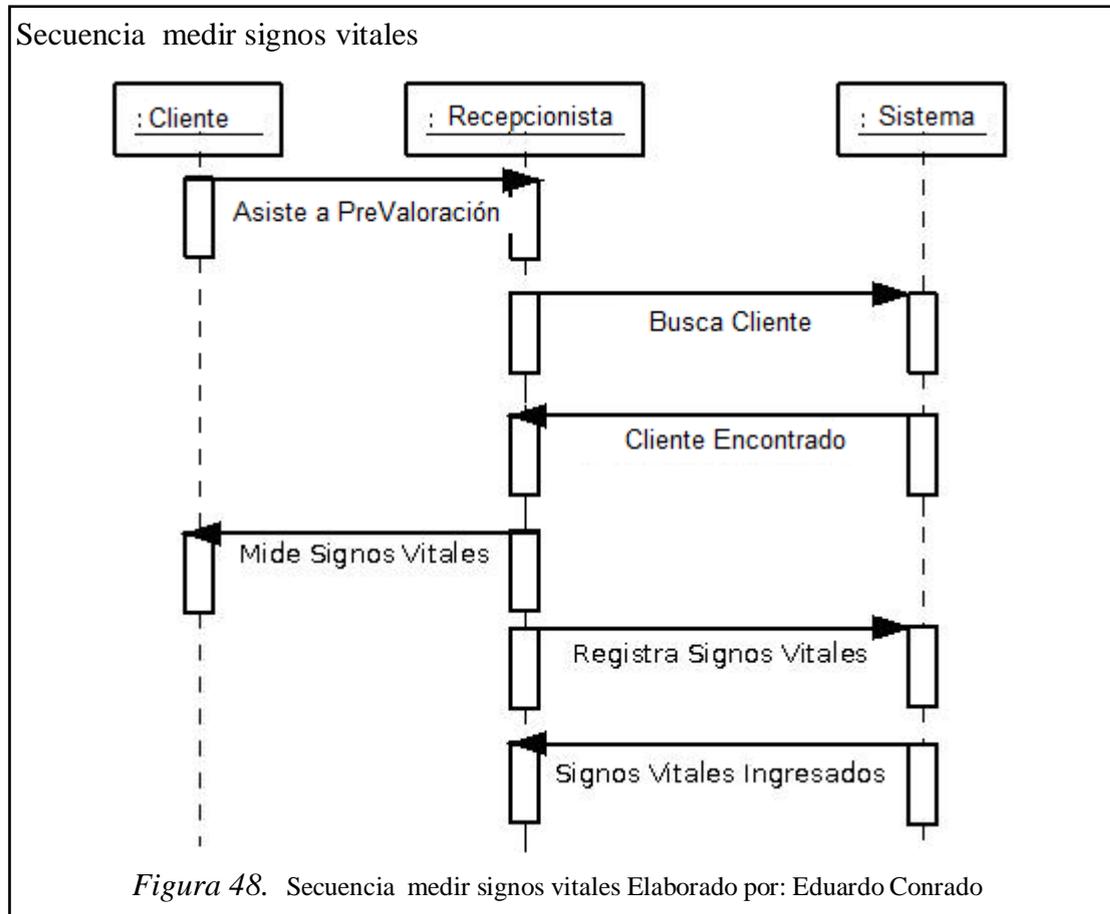
En la figura 46, cliente solicita ingreso al recepcionista; recepcionista verifica existencia del cliente en el sistema; sistema devuelve que cliente no existe; recepcionista registra los datos del nuevo cliente en el sistema; sistema devuelve mensaje, cliente ingresado.

### c) Secuencia asignar valoración física



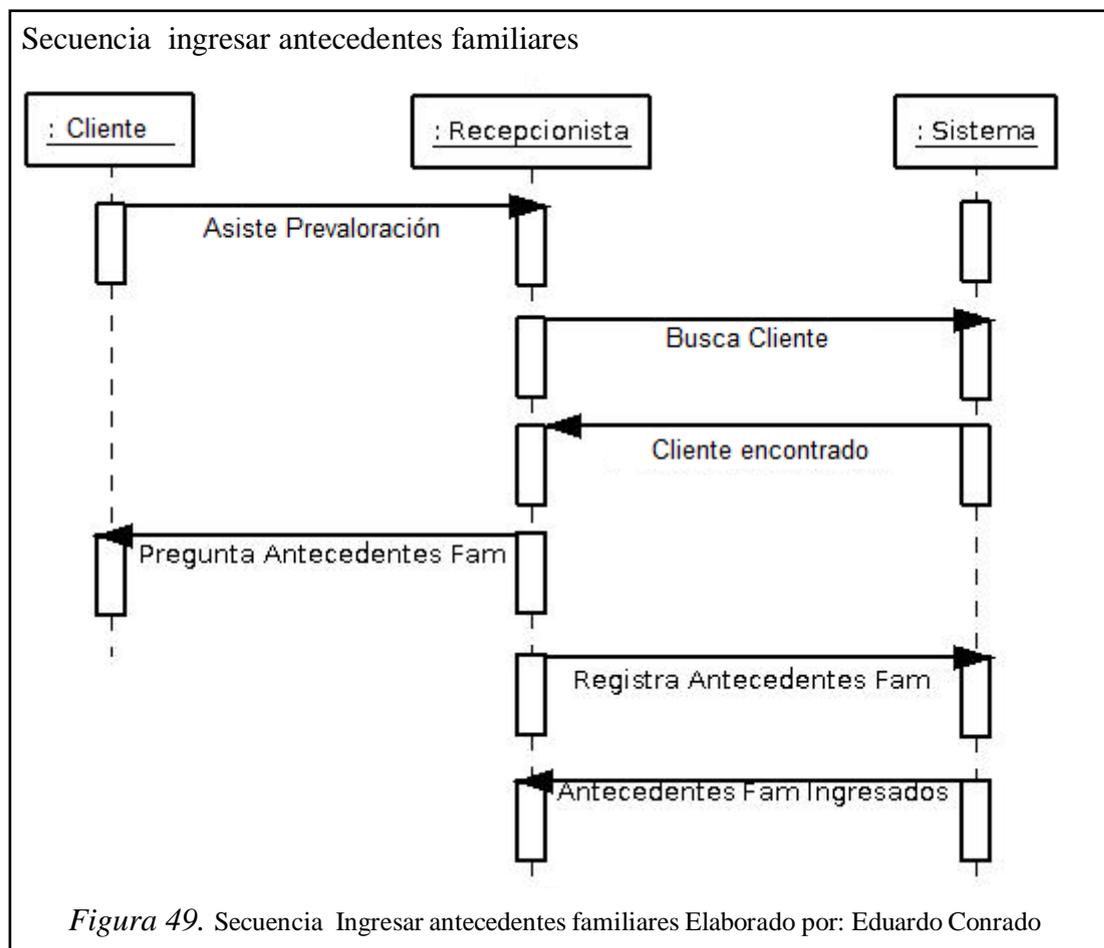
En la figura 47, cliente solicita turno al recepcionista; recepcionista verifica disponibilidad del entrenador en el sistema; sistema devuelve entrenador disponible; recepcionista confirma datos al cliente; recepcionista asigna valoración física; sistema devuelve mensaje, valoración física ingresada.

#### d) Secuencia medir signos vitales



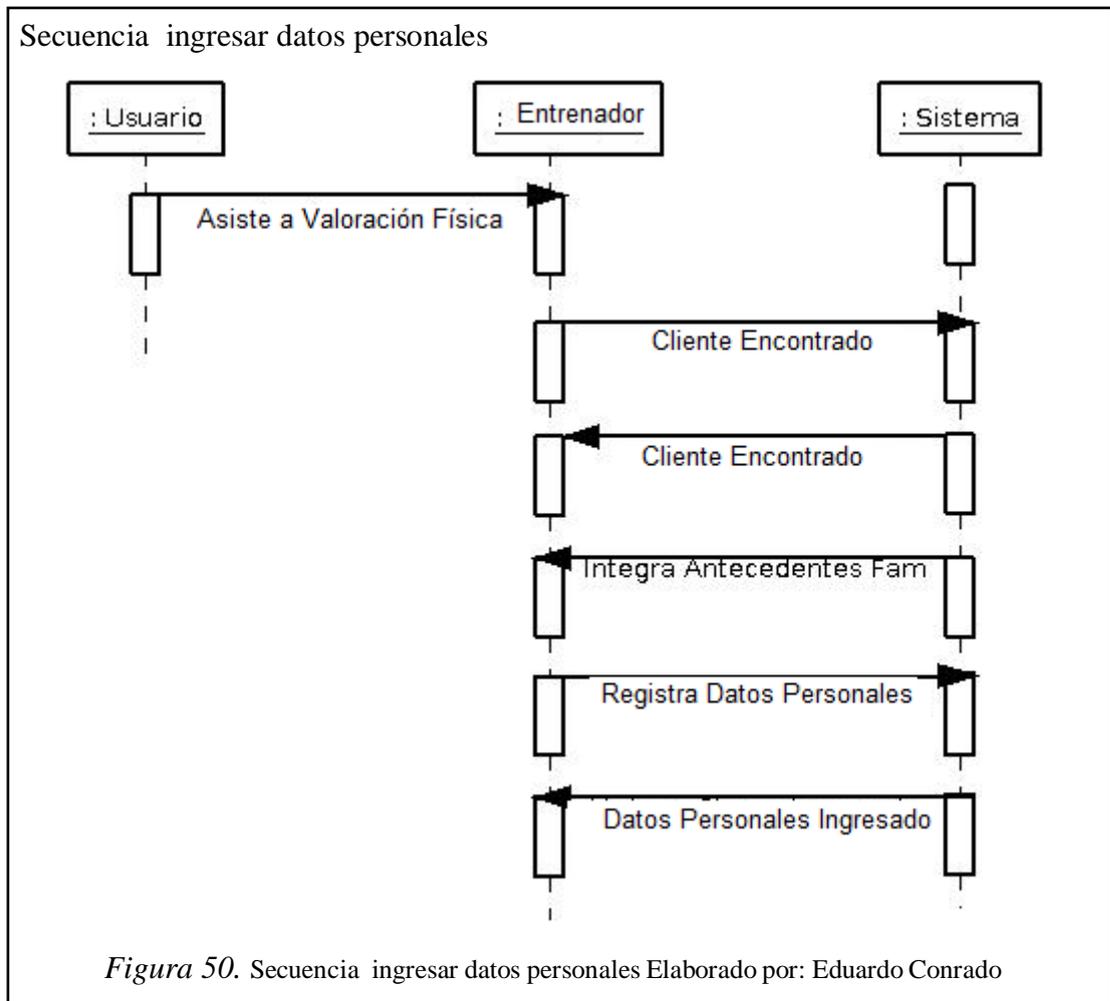
En la figura 48, cliente asiste a prevaloración con recepcionista; recepcionista busca al cliente en el sistema; el sistema encuentra al cliente; recepcionista mide los signos vitales del cliente; recepcionista registra los signos vitales en el sistema; por último el sistema devuelve mensaje de ingreso.

e) Secuencia ingresar antecedentes familiares



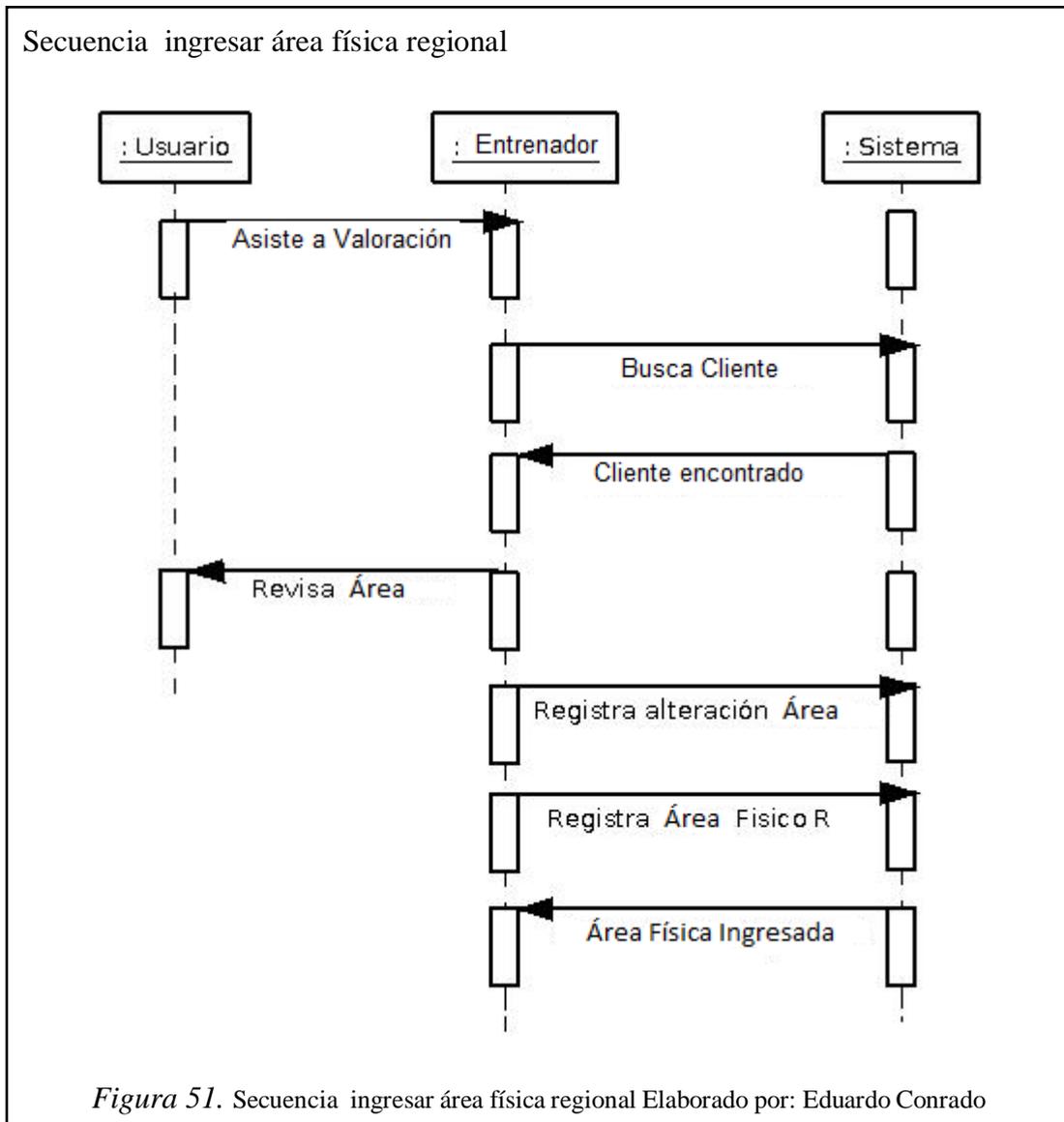
En la figura 49, cliente asiste a prevaloración con recepcionista; recepcionista busca al cliente en el sistema; el sistema encuentra al cliente; recepcionista pregunta antecedentes familiares del cliente; recepcionista registra los signos vitales en el sistema; por último el sistema devuelve mensaje de ingreso.

**f) Secuencia ingresar datos personales**



En la figura 50, cliente asiste a valoración física con entrenador; entrenador busca al cliente en el sistema; el sistema encuentra al cliente e integra antecedentes familiares; entrenador registra datos personales en el sistema; por último el sistema devuelve mensaje de ingreso.

**g) Secuencia ingresar área física regional**



En la figura 51, cliente asiste a valoración física con entrenador; entrenador busca al cliente en el sistema; el sistema encuentra al cliente; entrenador revisa órganos al cliente; el entrenador registra alteración de órganos y el examen físico; por último el sistema devuelve mensaje de ingreso.

## h) Secuencia atender valoración física

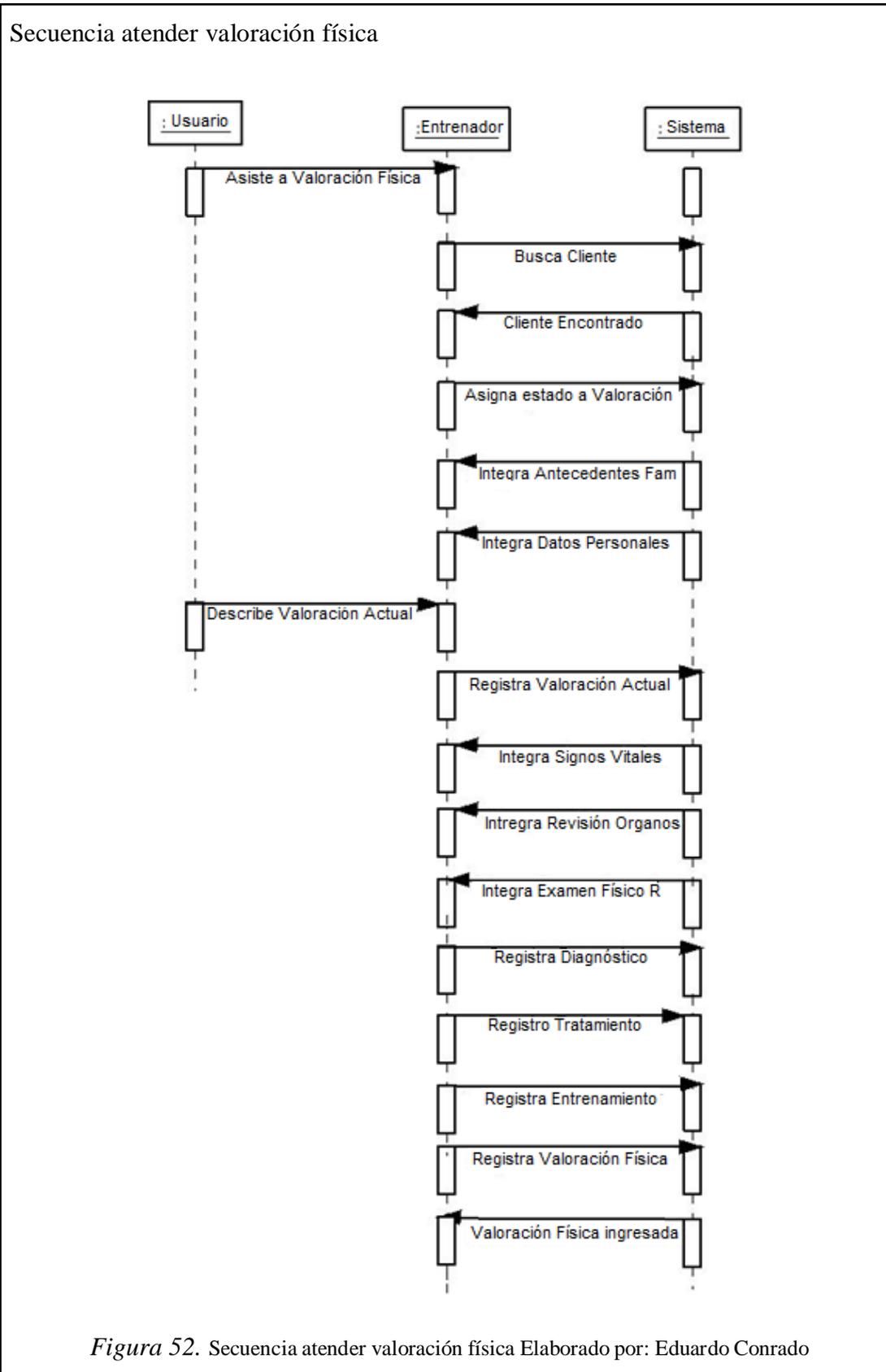


Figura 52. Secuencia atender valoración física Elaborado por: Eduardo Conrado

En la figura 52, cliente asiste a valoración física con entrenador; entrenador busca al cliente en el sistema; el sistema encuentra al cliente; entrenador asigna estado de valoración del cliente; sistema integra antecedentes familiares y datos personales; cliente describe valoración actual a entrenador; entrenador registra valoración actual del cliente; sistemas integra signos vitales, revisión área física; entrenador registra diagnóstico, tratamiento, entrenamiento, valoración física; por último el sistema devuelve mensaje de ingreso.

### 3.2.1.5 Diagrama conceptual

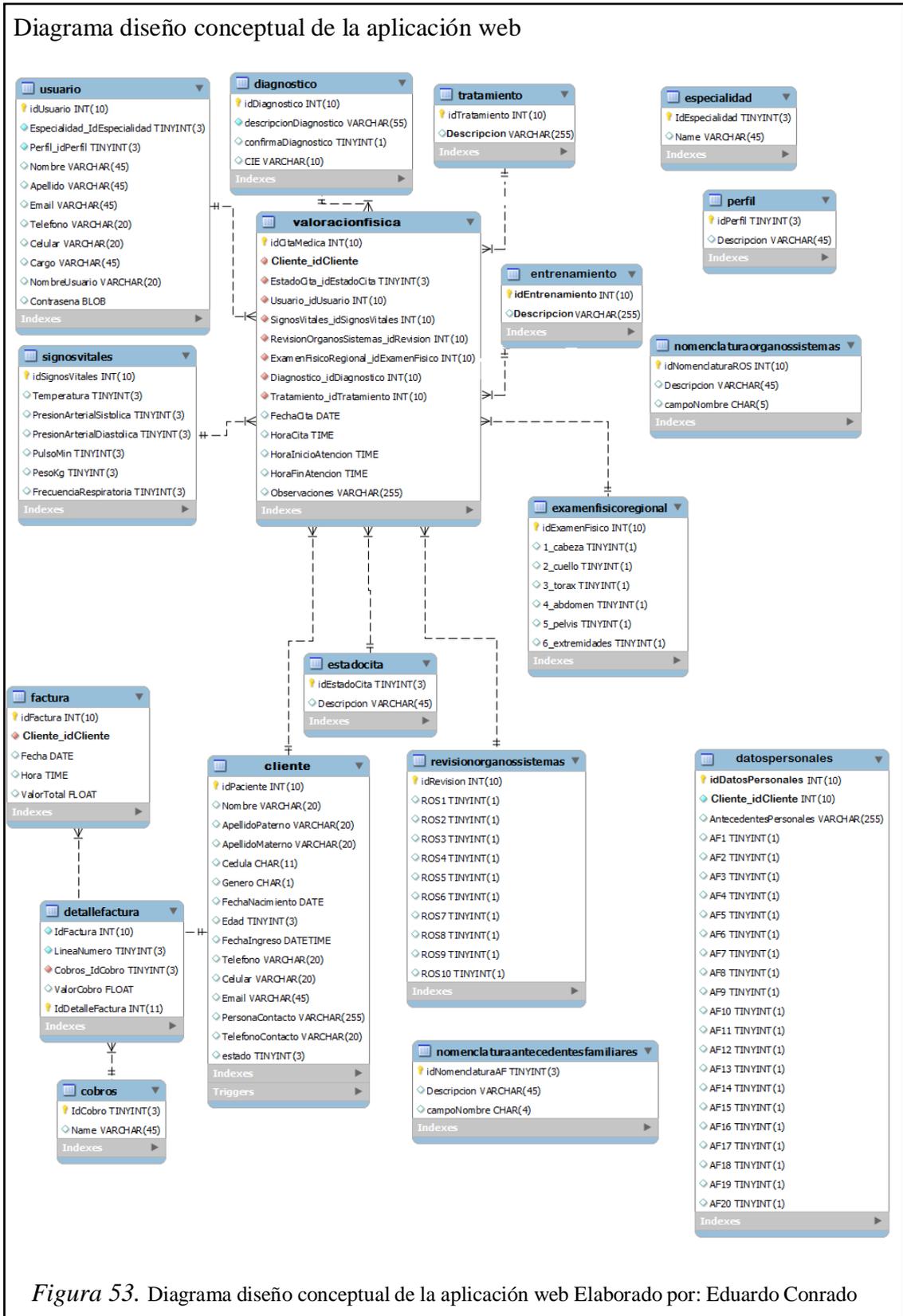


Figura 53. Diagrama diseño conceptual de la aplicación web Elaborado por: Eduardo Conrado

En la figura 53, se muestra la interacción entre las tablas que conforman la base de datos, con sus respectivos campos y el tipo de dato que reciben, las claves primarias, y las relaciones.

## **CAPÍTULO 4**

### **PRUEBAS DEL SISTEMA INFORMÁTICO E IMPLEMENTACIÓN**

#### **4.1 Pruebas de sistema de software**

Son actividades en las cuales un sistema o componente es ejecutado bajo unas condiciones o requerimientos específicos. Los resultados son observados y registrados, y una evaluación es hecha de algún aspecto del sistema o componente.

##### **4.1.1 Flujo trabajo de la prueba**

- Encontrar y documentar los defectos que puedan afectar la calidad del software.
- Validar que el software trabaje como fue diseñado.
- Validar y probar los requisitos que debe cumplir el software.
- Validar que los requisitos fueron implementados correctamente.

##### **4.1.2 Tipo de prueba adecuada al nivel de sistema**

El software ya validado se integra con el resto del sistema donde algunos tipos de pruebas a considerar son:

- Rendimiento: determinan los tiempos de respuesta, el espacio que ocupa el módulo en disco o en memoria, el flujo de datos que genera a través de un canal de comunicaciones, etc.
- Resistencia: determinan hasta donde puede soportar el programa determinadas condiciones extremas.
- Robustez: determinan la capacidad del programa para soportar entradas incorrectas.
- Seguridad: se determinan los niveles de permiso de usuarios, las operaciones de acceso al sistema y acceso a datos.

- Usabilidad: se determina la calidad de la experiencia de un usuario en la forma en la que éste interactúa con el sistema, se considera la facilidad de uso y el grado de satisfacción del usuario.
- Instalación: se determinan las operaciones de arranque y actualización del software.

### 4.1.3 Rendimiento

El rendimiento es el tiempo de respuesta del flujo de datos de un canal de comunicaciones.

Las pruebas de rendimiento son realizadas sobre computadoras, redes, software u otros dispositivos. Estas son utilizadas para determinar la velocidad y eficiencia de los mismos. Este procedimiento puede involucrar tanto *tests* cuantitativos como por ejemplo: medir tiempos de respuesta o cantidad en gran cantidad de líneas de código; como pruebas cualitativas, en los cuales se evalúa fiabilidad, escalabilidad e interoperabilidad. Estas pruebas de rendimiento pueden ser realizadas a través de herramientas que proveen pruebas de estrés que permiten determinar la estabilidad del sistema.

Las limitaciones en los tiempos de respuesta de un sitio web y una aplicación de escritorio son similares, y no han cambiado en el transcurso de los años. Cabe aclarar que en la caso de los sitios web el tiempo está muy relacionado a la velocidad del enlace donde se esté navegando.

Según el autor Jakob Nielsen, en el libro “Usability Engineering” existen tres límites importantes en el tiempo de respuesta:

- 0,1 segundo: es el límite en el cual el usuario siente que esta “manipulando” los objetos desde la interfaz de usuario.
- 1 segundo: es el límite en el cual el usuario siente que está navegando libremente sin esperar demasiado una respuesta del servidor.

- 10 segundos: es el límite en el cual se pierde la atención del usuario, si la respuesta tarda más de 10 segundos se deberá indicar algún mecanismo por el cual el usuario pueda interrumpir la operación.

En el proyecto, el tiempo está condicionado a los siguientes puntos:

- El servidor testeado se encuentra en la misma red en la cual se realizaron las pruebas.
- Velocidad de conexión del servidor.
- Velocidad de conexión del cliente.
- Tiempo en el cual el navegador web tarda para dibujar la página (tiempo muy pequeño).
- Rendimiento de la red en el momento de la prueba.

#### **4.1.4 Características de la prueba**

En esta sección se describirá tanto la herramienta utilizada como las pruebas realizadas.

##### **4.1.4.1 Herramienta Apache JMeter**

Apache JMeter es una herramienta de carga diseñada para realizar pruebas de rendimiento y pruebas funcionales sobre aplicaciones web. Es desarrollado por THE APACHE SOFTWARE FOUNDATION<sup>1</sup>. Originalmente el Apache JMeter fue diseñado para realizar pruebas de estrés sobre aplicaciones web (pruebas web clásicas). Hoy en día su arquitectura ha evolucionado, ahora no sólo puede llevar a cabo pruebas en componentes típicos de Internet (HTTP), sino también puede realizar pruebas sobre Bases de Datos, *scripts Perl*, *servlets*, objetos java, servidores FTP y prácticamente cualquier medio de los que se pueden encontrar en la red. JMeter también permite la ejecución de pruebas distribuidas entre distintos ordenadores, para realizar pruebas de rendimiento.

El Apache JMeter está diseñado para desarrollar diferentes tipos de test, permitiendo diseñar tanto sencillos *tests* que soliciten simples páginas web; así como, complejas

---

<sup>1</sup><http://jakarta.apache.org/jmeter/>

secuencias de requisiciones que permitan evaluar el comportamiento de una aplicación o como la capacidad de carga máxima que pueda tener una aplicación en un servidor (pudiendo llegar a saturar el servidor).

El Apache JMeter incluye una interfaz gráfica de usuario que facilita el diseño de las pruebas. Esta interfaz gráfica además de aportar un entorno cómodo de trabajo también permite guardar y alterar tanto los *tests* desarrollados como los componentes que lo integran. Gracias a esto se pueden reutilizar las pruebas o módulos de las mismas en el desarrollo de nuevas pruebas.

Además de las funcionalidades de prueba antes mencionadas, el Apache JMeter también ofrece la posibilidad de activar un *Proxy web*. Gracias a esto se puede grabar la navegación de un usuario para posteriormente usarla en la generación de una prueba.

#### 4.1.4.2 Realización de la prueba

Para analizar el tiempo de respuesta del servidor, se utilizó la herramienta JMeter. JMeter permite realizar *test* de comportamiento funcional y medir el rendimiento. También se puede utilizar para realizar pruebas de estrés, por ejemplo, en un servidor, y poner a prueba su rendimiento. Para estas pruebas, se configuró un servidor Proxy que provee JMeter, para poder construir un camino de navegación aleatorio y así simular la visita de un usuario. Para poder evaluar los resultados se utilizaron los componentes provistos por la herramienta:

Summary report: permite visualizar los resultados del *test* realizado, en una tabla. Los datos que presenta son:

- *Label*: etiqueta de la muestra
- #Muestras: cantidad de *thread* utilizados para la URL.
- Media: tiempo promedio en milisegundos para un conjunto de resultados.
- Min: tiempo mínimo que demora un *thread* en acceder a una página.
- Max: tiempo máximo que demora un *thread* en acceder a una página
- Rendimiento: rendimiento medido en los requerimiento por segundo / minuto / hora.

- Kb/sec: rendimiento medido en Kbytes por segundo.
- Media en bytes: tamaño medio de respuesta del servidor (en bytes).

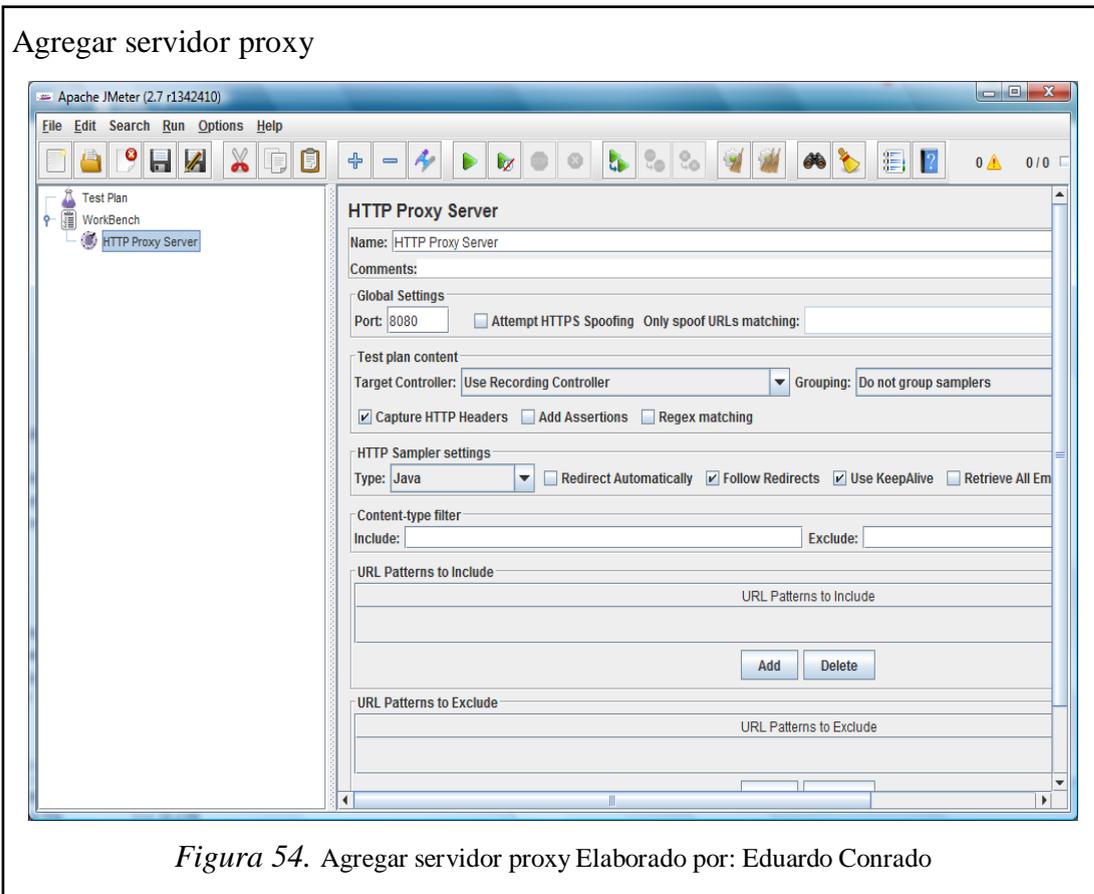
Aggregate graph: este componente es similar al anterior, pero permite obtener resultados más precisos. Utiliza más memoria, ya que calcula la mediana y la línea al 90%, la cual requieren que todos los datos estén almacenados. Los datos que se presentan son:

- URL : etiqueta de la muestra
- #Muestras: cantidad de *thread* utilizados para la URL.
- Media: tiempo promedio en milisegundos para un conjunto de resultados.
- Mediana: valor en tiempo del percentil 50.
- Línea de 90%: máximo tiempo utilizado por el 90% de la muestra, al resto de la misma le llevo más tiempo.
- Min: tiempo mínimo de la muestra de una determinada URL.
- Max: tiempo máximo de la muestra de una determinada URL.
- %Error: porcentaje de requerimientos con errores.
- Rendimiento: rendimiento medido en los requerimiento por segundo / minuto / hora.
- KB/sec: rendimiento medido en Kbytes por segundo.

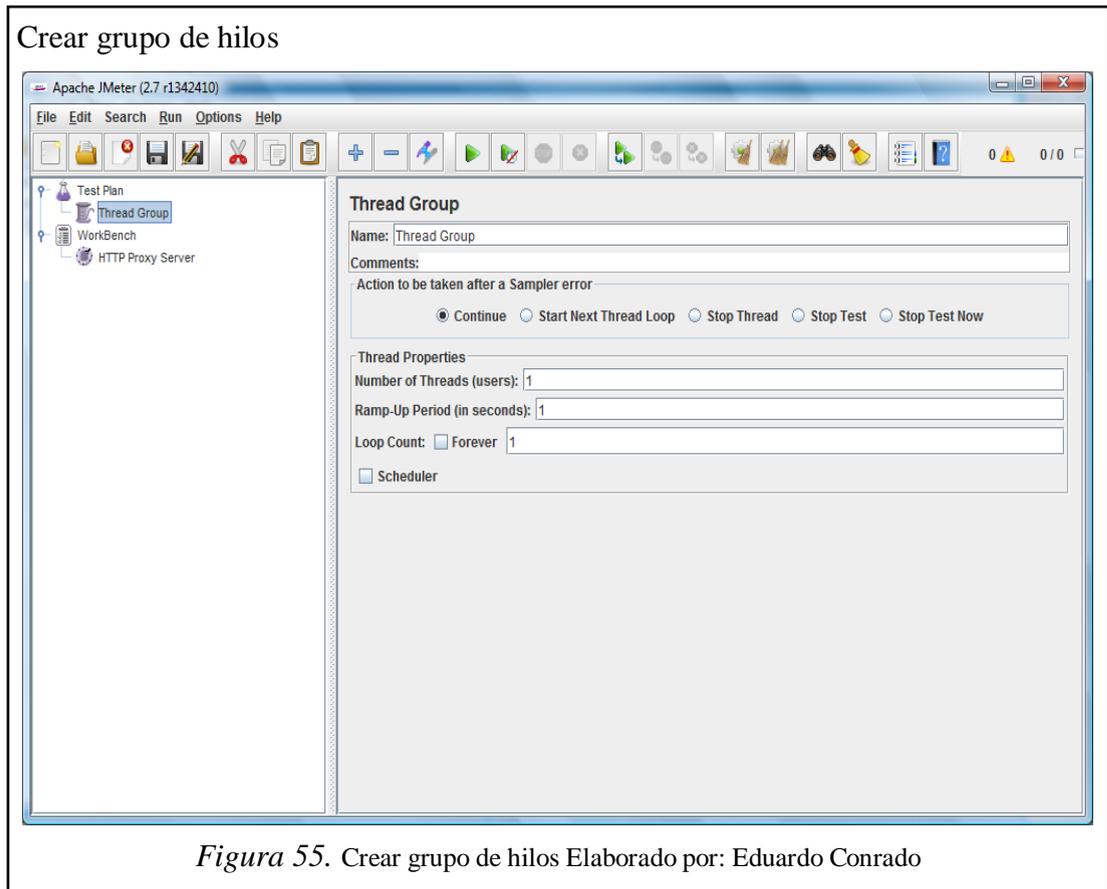
Gráfico de resultados: esta componente permite visualizar gráficamente los siguientes datos: media, mediana, dispersión y el rendimiento (representado como el número actual de requerimientos/minutos que el servidor maneja).

#### 4.1.4.3 Prueba

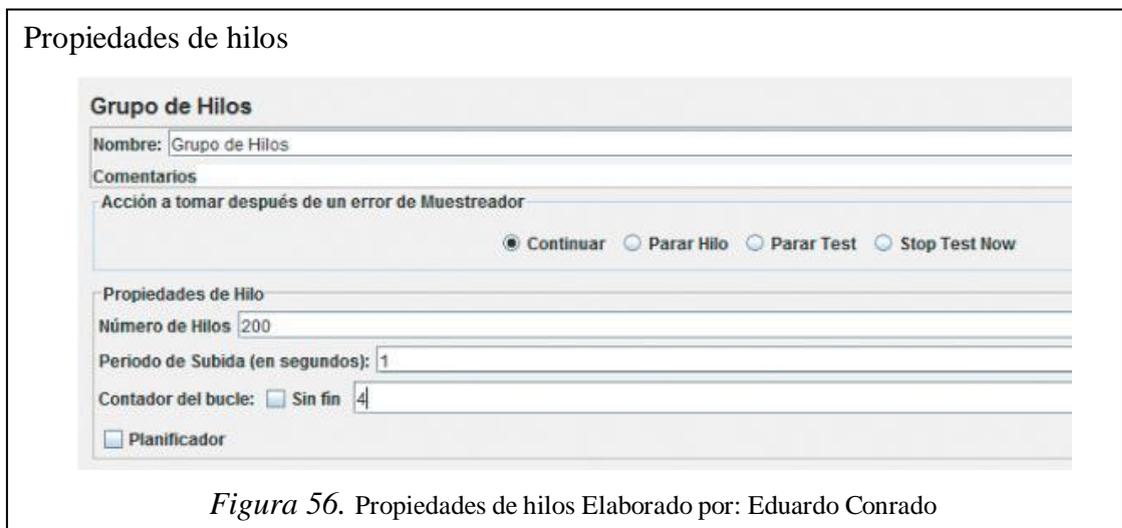
Primero se genera un plan de pruebas desde el menú principal de Apache Jmeter. Seguido se procede a agregar el servidor Proxy para el desarrollo de la prueba.



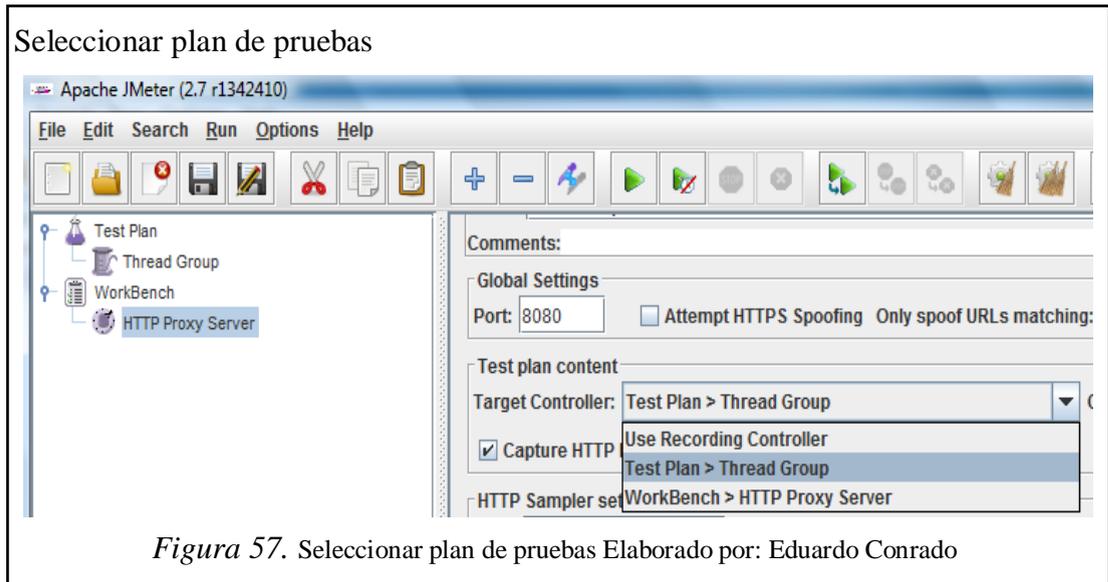
A continuación se procede a crear un grupo de hilos, en el cual se almacenarán todas las peticiones detectadas por el servidor proxy.



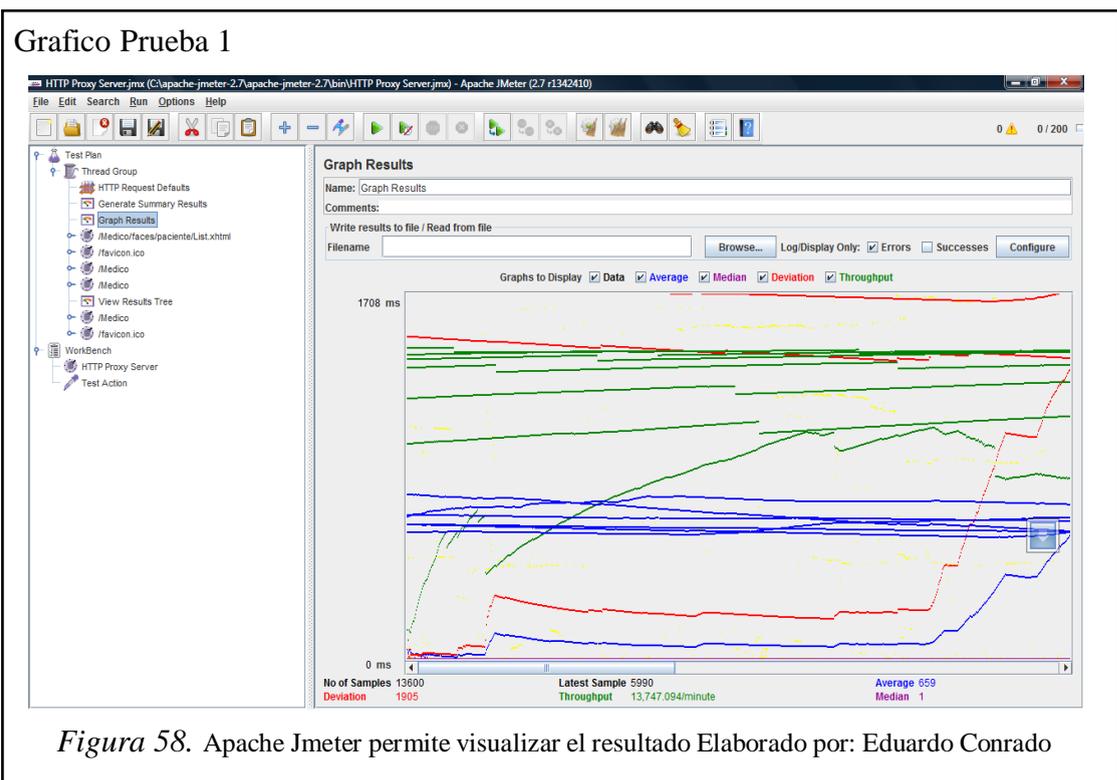
En *Thread Properties* o Propiedades de Hilo se deberá digitar el número de peticiones. En este caso se coloca la cantidad de 200 con periodo de subida de 1 segundo con iteraciones 4 en total.



Una vez creado el grupo de hilos, hay que configurar el servidor proxy para que almacene las peticiones en él.



#### 4.1.4.4 Gráficos de los resultados



#### 4.1.4.5 Resultados obtenidos

La prueba realizada consistió en definir *tests* de 200 *threads* cada uno, los cuales simulan 200 accesos de usuarios respectivamente. Para realizar la prueba se definió una lista de enlaces, a los se simula accesos aleatorios y a partir de ahí, se recolectan los datos necesarios para su interpretación.

Tabla 6.  
*Resumen del Reporte*

<b>Inte r</b>	<b>#Muestr as</b>	<b>Promedi o</b>	<b>Mi n</b>	<b>Max</b>	<b>%erro r</b>	<b>Std.De v</b>	<b>Rendimien to</b>	<b>Kb/se c</b>
Esp	800	1338	3	9600	1.0%	2628.5 5	7.4/sec	23.21
Graf	10400	439	1	2443	1.0%	2032.4 8	62.2/sec	98.04
Inici o	11200	1467	1	3062 1	1.0%	3421.9 3	67.3/sec	187.9 3
<b>Tota l</b>	22400	985	1	3062 1	1.0%	2877.2 3	134.3/sec	300.4 4

Nota. Elaborado por: Eduardo Conrado

## Resultado final en el JMeter

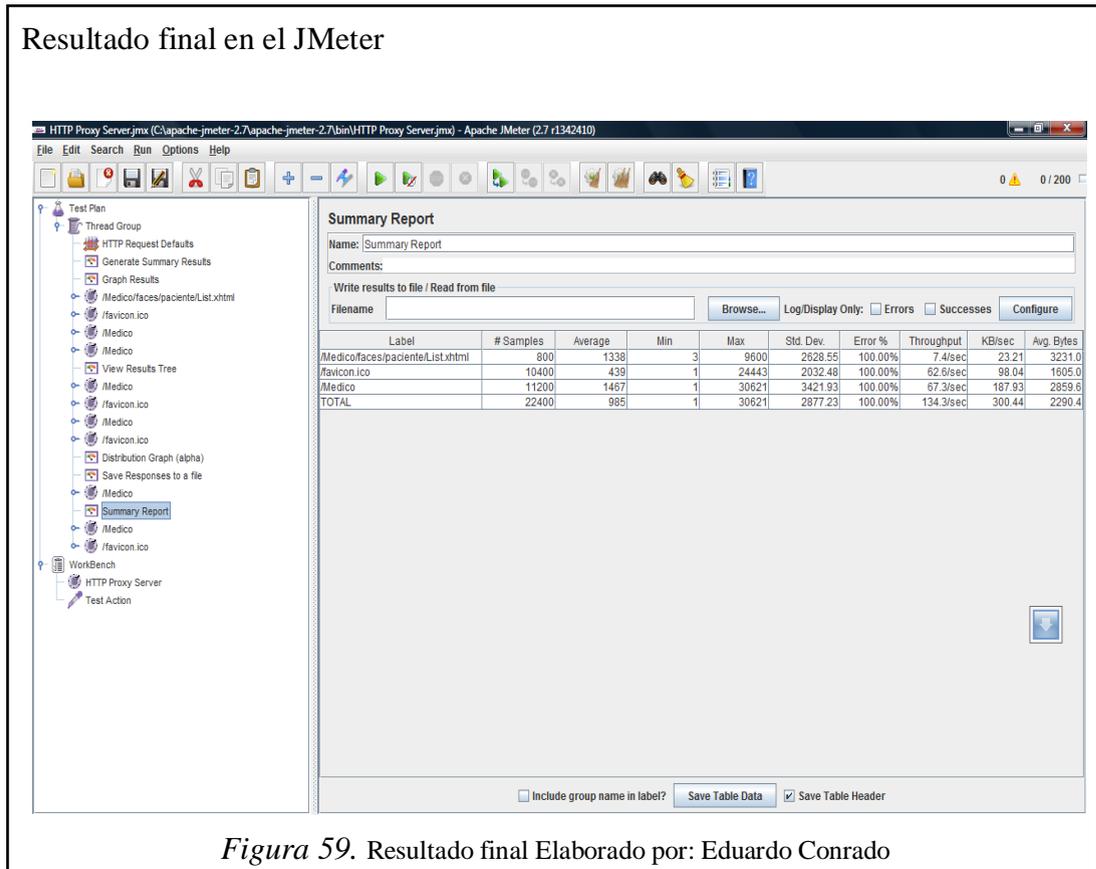


Figura 59. Resultado final Elaborado por: Eduardo Conrado

### 4.1.4.6 Interpretación de resultados

Con 200 usuarios realizando solicitudes, se obtiene un tiempo de respuesta promedio entre 7,4 y 13,2 segundos. Se puede observar cuando al acceder específicamente a una página de ingreso de datos del cliente, la respuesta es corta, porque la conexión se encuentra realizada. Si se accede por primera vez a la aplicación, el tiempo es de 67.3 segundos de donde el gráfico es el que se demora más en cargar con un tiempo de 62.2 segundos. Por la demora en el acceso principal es recomendable cambiar el tipo de archivo de la imagen por una más liviana.

### 4.1.5 Robustez

#### 4.1.5.1 Capacidad del programa para soportar entradas incorrectas

Pruebas de robustez, comúnmente denominadas "*robustness test*", son las encargadas de verificar la capacidad del programa para soportar entradas incorrectas. Por ejemplo, en un sistema donde, el usuario debe ingresar códigos de productos y luego

cantidades es más que factible que en algún momento ingrese un código en el campo de cantidad. Si el programa fue sometido a pruebas de robustez, este valor sería rechazado o grabado como una cantidad inmensa pero que no daría error por desbordamiento de datos.

#### 4.1.5.2 JUnit

JUnit es un conjunto de clases (*framework*) que permite realizar la ejecución de clases Java de manera controlada para poder evaluar si el funcionamiento de cada uno de los métodos de la clase se comporta como se espera. (Wikipedia, Septiembre 2011).

Para crear los JUnit *Tests*, primero se selecciona la clase que se desea analizar mediante clic derecho se selecciona *Tools / Create JUnit Tests*.

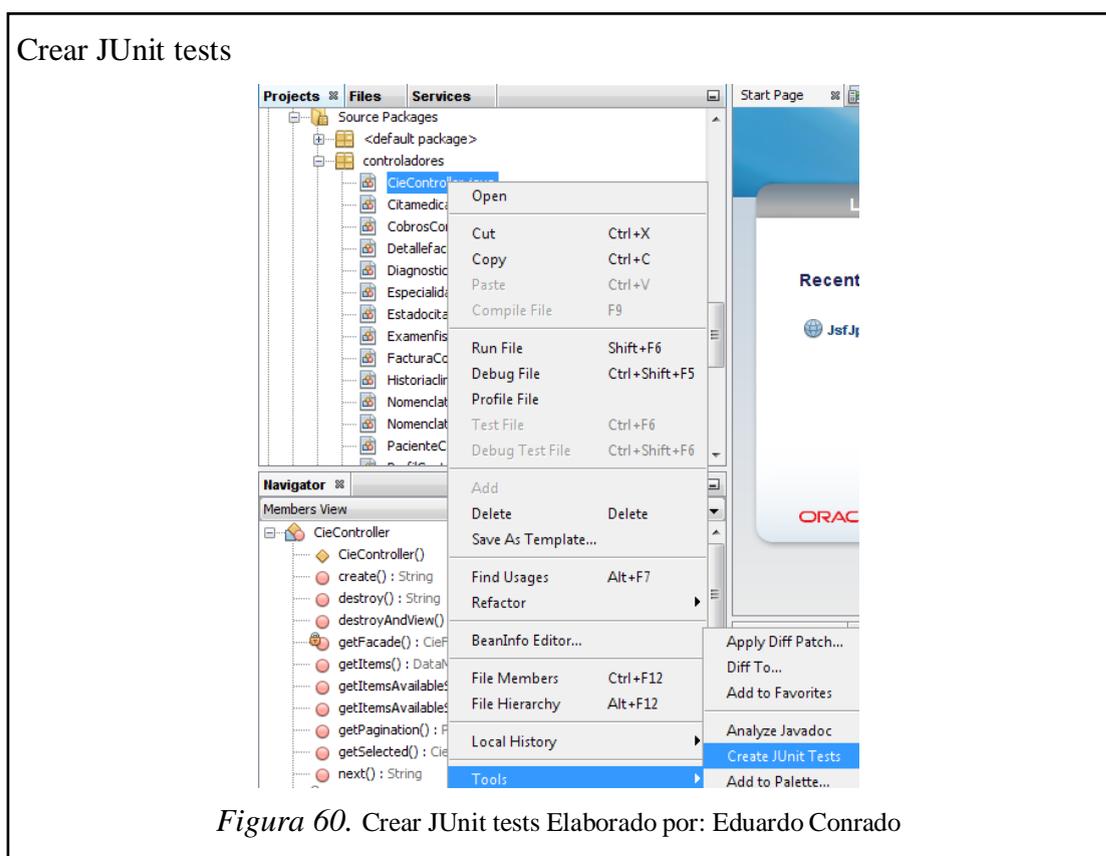


Figura 60. Crear JUnit tests Elaborado por: Eduardo Conrado

Inmediatamente aparece una ventana en donde se selecciona de una lista de verificación la generación del código.

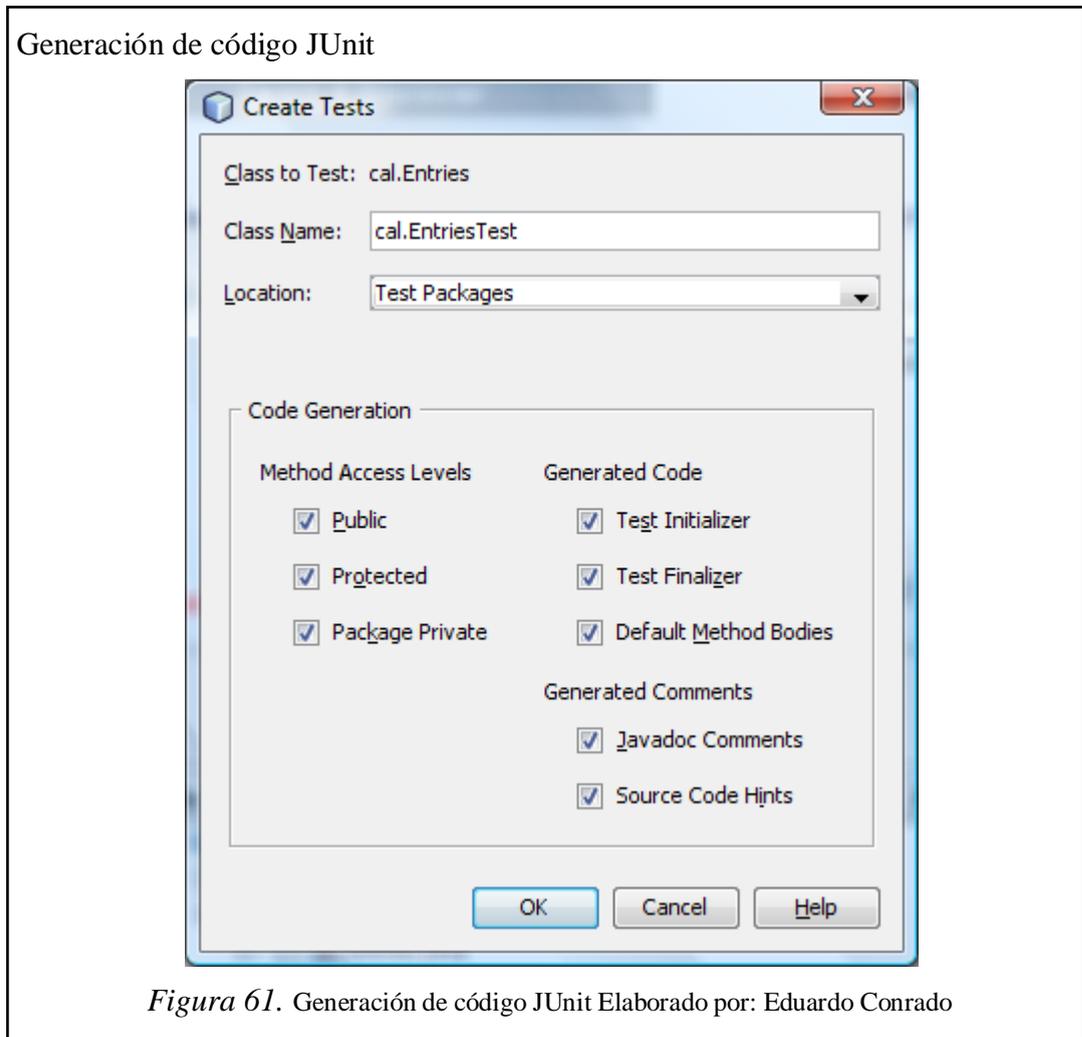
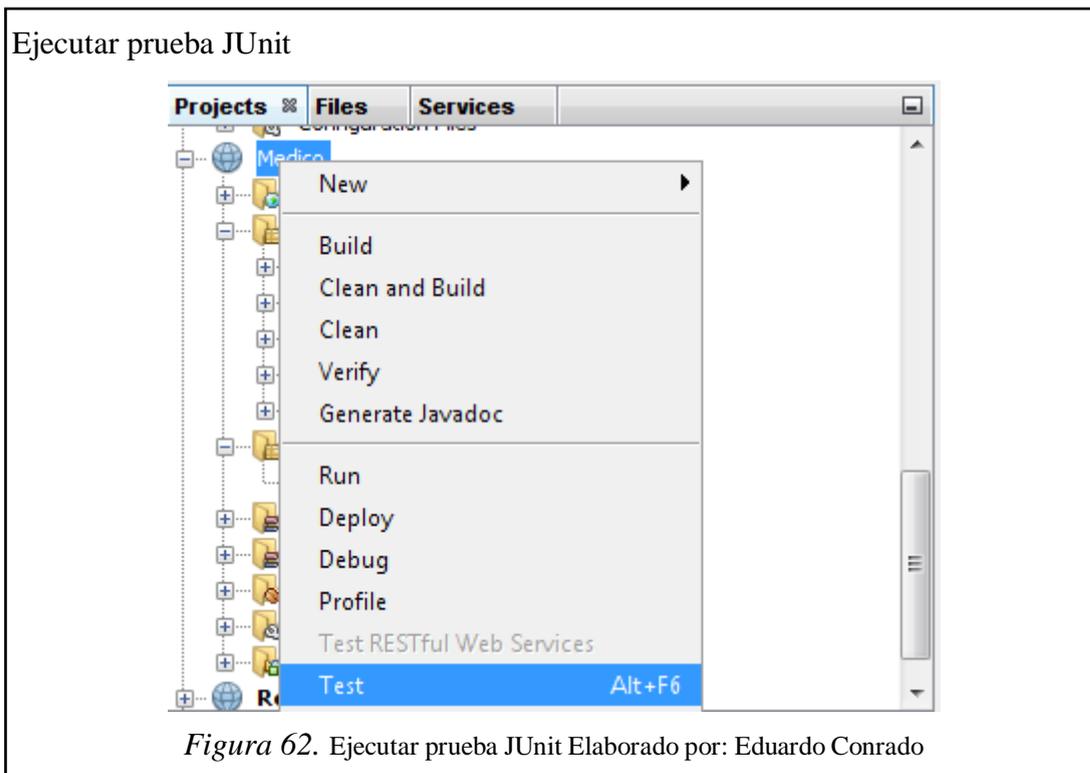


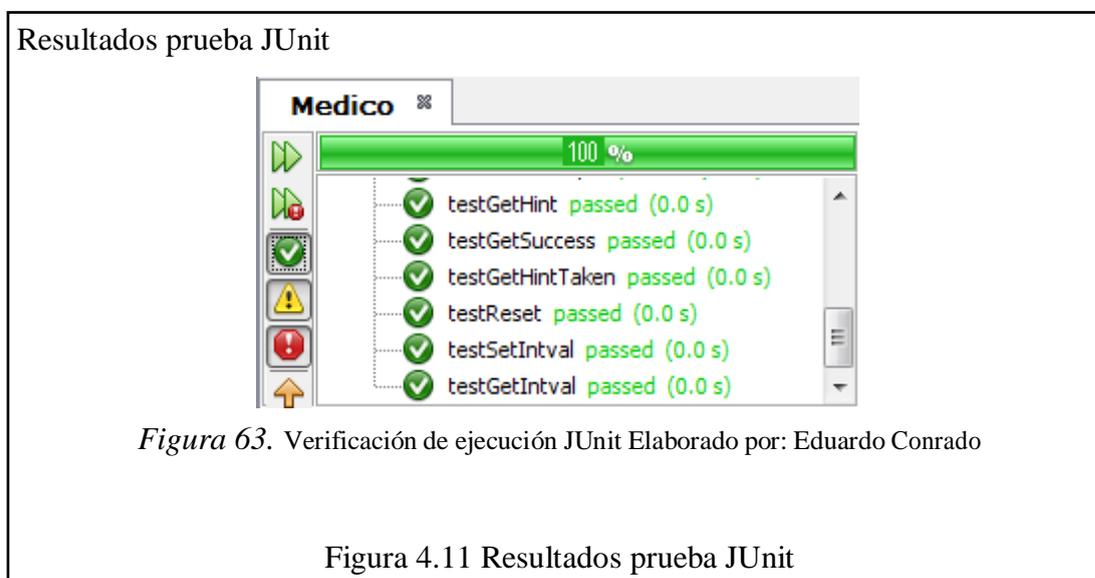
Figura 61. Generación de código JUnit Elaborado por: Eduardo Conrado

JUnit genera automáticamente paquetes de prueba para su análisis. La herramienta permite escoger el ejecutar una prueba unitaria o una general del proyecto. El IDE ejecuta todas las clases de prueba en la carpeta *test packages*.

Para ejecutar la prueba se selecciona el proyecto y se hace un clic derecho en test.



Cuando se ejecuta el test, se verá los siguientes resultados del test de JUnit.



## **4.1.6 Seguridad**

### **4.1.6.1 Niveles de permisos de usuarios, y los accesos a los datos**

De acuerdo a la seguridad que se desea, el nivel de seguridad de la aplicación se asegura que a los usuarios se les restrinjan las funciones específicas o casos de uso, o que están limitados en los datos disponibles para ellos. El objetivo es proteger la base de datos contra accesos no autorizados. Asignando niveles de seguridad, los cuales permitan al usuario solo acceder a la información que tenga permiso. Primeramente se deberá identificar y enumerar cada tipo de usuario, funciones, datos a los que cada podrá tener acceso o permiso de visualizarlo, modificarlo o eliminarlo.

Al momento de realizar las pruebas de funcionamiento se deberá realizarlo con cada uno de tipos de usuario. Con esto se podrá verificar cada permiso tratando de realizar alguna actividad dentro del sistema. Ya que habrá transacciones específicas o exclusivas para cada tipo de usuario.

### **4.1.6.2 Flujo de trabajo de la prueba de seguridad**

- Verificar que los usuarios que tengan privilegios no tengan acceso al sistema
- Verificar que los usuarios que no sean tipo administrador no visualicen los datos financieros del gimnasio
- Verificar que los usuarios de sólo Valoración Física solo vean la información a la que tienen privilegio

### **4.1.6.3 Tipos de usuarios**

Administrador – súper administrador – DBA: están permitidas todas las operaciones, se conceden privilegios y crea o establece nuevos usuarios. Es un usuario que tiene derecho a crear, borrar y modificar objetos y que además puede conceder privilegios a otros usuarios sobre los objetos que ha creado.

Oficinista – recepción: usuario con derecho a valoración física o actualizar y sin derecho a crear o borrar objetos. Privilegios sobre los ciertos datos, añadir nuevos datos, etc.

#### **4.1.7 Usabilidad**

Experiencia del usuario cuando interactúa con el sistema, considerando la facilidad de uso y el grado de satisfacción del usuario

La organización internacional para la estandarización (ISO) ofrece dos definiciones de usabilidad<sup>2</sup>:

ISO/IEC 9126: "La usabilidad se refiere a la capacidad de un software de ser comprendido, aprendido, usado y ser atractivo para el usuario, en condiciones específicas de uso"

Esta definición hace énfasis en los atributos internos y externos del producto, los cuales contribuyen a su funcionalidad y eficiencia. La usabilidad depende no sólo del producto sino también del usuario. Por ello un producto no es en ningún caso intrínsecamente usable, sólo tendrá la capacidad de ser usado en un contexto particular y por usuarios particulares. La usabilidad no puede ser valorada estudiando un producto de manera aislada. (Bevan, 1999)

ISO/IEC 9241: "Usabilidad es la eficiencia y satisfacción con la que un producto permite alcanzar objetivos específicos a usuarios específicos en un contexto de uso específico". Es una definición centrada en el concepto de calidad en el uso, es decir, se refiere a cómo el usuario realiza tareas específicas en escenarios específicos con efectividad. (Bevan, 1999)

##### **4.1.7.1 Principios básicos**

A partir de la conceptualización llevada a cabo por la ISO, se infieren los principios básicos en los que se basa la usabilidad:

Facilidad de Aprendizaje: facilidad con la que nuevos usuarios desarrollan una interacción efectiva con el sistema o producto. Está relacionada con la predicibilidad, sintetización, familiaridad, la generalización de los conocimientos previos y la consistencia.

---

<sup>2</sup> Enciclopedia, Usabilidad

Flexibilidad: relativa a la variedad de posibilidades con las que el usuario y el sistema pueden intercambiar información. También abarca la posibilidad de diálogo, la multiplicidad de vías para realizar la tarea, similitud con tareas anteriores y la optimización entre el usuario y el sistema.

Robustez: es el nivel de apoyo al usuario que facilita el cumplimiento de sus objetivos. Está relacionada con la capacidad de observación del usuario, de recuperación de información y de ajuste de la tarea al usuario.

## **4.2 Implementación del software en el gimnasio**

El manual del administrador y usuario del sistema contienen especificaciones para la instalación del software que se deberán seguir para su correcto funcionamiento. La prueba de la instalación tiene dos propósitos: El primero es asegurarse de que el software se puede instalar bajo diversas condiciones tales como una nueva instalación, una actualización, e instalaciones personalizadas o completas bajo condiciones normales o anormales. Las condiciones anormales incluyen insuficiente espacio en disco, escasos privilegios de crear directorios, etcétera. El segundo propósito es verificar que, una vez que esté instalado, el software funciona correctamente. Se deberá realizar una instalación sobre cada configuración de hardware requerida bajo las condiciones específicas para observar y registrar los cambios del estado del comportamiento y de la configuración de la instalación del registro.

### **4.2.1 Condiciones específicas**

- Nueva instalación: una máquina donde no se ha instalado previamente el software.
- Actualización: una máquina instalada previamente, la misma actualización, misma versión.

Esta técnica soporta la prueba de la instalación del producto desarrollado en unas o más configuraciones de la instalación.

## CAPÍTULO 5

### ELABORACIÓN DE LAS GUÍAS DEL SOFTWARE

#### 5.1 Guía del administrador

##### 5.1.1 Requerimientos de los equipos

Las características mínimas de los equipos son las siguientes:

###### 5.1.1.1 Servidor de base de datos

- Procesador Pentium IV de 1.5 Ghz.
- Disco duro de 40 Gb.
- 256 mb en memoria RAM

###### 5.1.1.2 Estaciones de trabajo

- Procesador Pentium III
- Disco duro de 20 Gb.
- 128 mb en memoria RAM

###### 5.1.1.3 Software

- Navegador actualizado (Mozilla Firefox, Google Chrome, IE 8 )
- Sun Java Runtime Environment (JRE)
- MySQL 5.0 (base de datos)

##### 5.1.2 Creación de la base de datos

Se deberá crear la base de datos mediante el script proporcionado en CD del manual de instalación de la aplicación web. Se ejecutará el *script* mediante la consola de MySQL con el comando *source* y la ubicación física del *script*. La base de datos se

ubicará en el servidor de base de datos y la aplicación se accederá desde cada estación de trabajo.

### **5.1.3 Ingreso de información tablas catálogo**

Igual que le proceso anterior se deberá ejecutar el *script* para el ingreso de datos de las tablas catálogo.

### **5.1.4 Funcionamiento**

En el momento de la instalación, se debe definir de antemano quien deberá ser la persona que se encargue de la administración del sitio. El administrador de la aplicación web es el encargado de manejar la aplicación, se recomienda que el administrador sea la persona que registre a los usuarios asignándoles permisos de acuerdo a las actividades que realice el usuario.

## **5.2 Guía del usuario**

Para acceder a la aplicación web es procederá a abrir un navegador de internet con la siguiente URL:

<http://localhost:8080/Gimnasio>

### 5.2.1 Acceso al sistema web interno

En la figura 64, se muestra el formulario de inicio para ingresar el nombre de usuario y contraseña.

Acceso al sistema web interno

## ACCESO AL SISTEMA



Usuario:

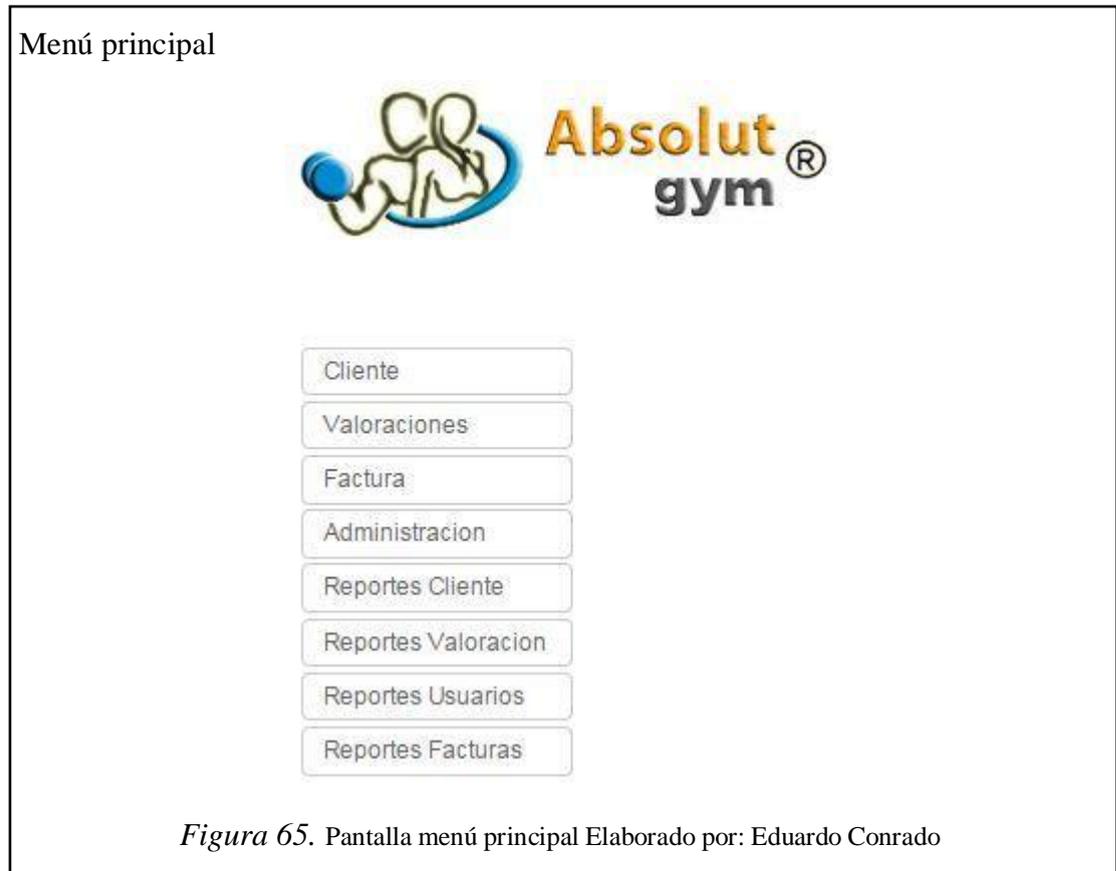
Contraseña:

viernes 11 de julio de 2014 19:06

*Figura 64.* Acceso al sistema web interno Elaborado por: Eduardo Conrado

### 5.2.2 Menú principal

En la figura 65, en la parte izquierda, se despliega toda la lista de opciones del menú.



De acuerdo al nivel de permisos del usuario, se activa o desactiva las opciones del menú principal de la aplicación web.

### 5.2.3 Tipos de usuarios

Solo el usuario administrador tendrá acceso a todas las opciones del menú principal. El usuario podrá dar de alta, de baja modificar cualquier información de la aplicación web.

## 5.2.4 Información de usuario

En la figura 66, por medio del menú principal, submenú administración, se accede a la lista de todos los usuarios del centro.

Lista de usuarios ingresados

Lista de Usuarios ingresados			
ID	Nombres y Apellidos	Perfil	Acción
1	Paola Andrade	RECEPCIONISTA	<a href="#">Modific</a>
2	Pablo Perez	ENTRENADOR PERSONAL	<a href="#">Modific</a>
3	Manuel Landazuri	MEDICO	<a href="#">Modific</a>
4	Amelia Vega	MEDICO	<a href="#">Modific</a>
5	Eduardo Conrado	SUPERADMINISTRADOR	<a href="#">Modific</a>
6	David Ballesteros	ADMINISTRADOR	<a href="#">Modific</a>
7	Angelica Davila	RECEPCIONISTA	<a href="#">Modific</a>
9	Juan Villacis	ENTRENADOR PERSONAL	<a href="#">Modific</a>
10	Laura Ramirez	ENTRENADOR PERSONAL	<a href="#">Modific</a>
11	Jerson Rodriguez	ENTRENADOR PERSONAL	<a href="#">Modific</a>
12	Miguel Huertas	ENTRENADOR PERSONAL	<a href="#">Modific</a>
13	Ricky Glamour	ENTRENADOR PERSONAL	<a href="#">Modific</a>

< ||| >

*Figura 66.* Mostrar lista de usuarios Elaborado por: Eduardo Conrado

### 5.2.5 Crear usuario

En la figura 67, en la parte inferior derecha de la figura anterior, se accede al enlace crear nuevo usuario.

Crear usuario

**Datos del Usuario**

Nombre:(*)	<input type="text"/>
Apellido:(*)	<input type="text"/>
Email:	<input type="text"/>
Telefono:(*)	<input type="text"/>
Celular:	<input type="text"/>
Cargo:(*)	<input type="text"/>
Perfil:(*)	<input type="text" value="Seleccionar"/>
Especialidad:(*)	<input type="text" value="Seleccionar"/>
NombreUsuario:(*)	<input type="text" value="admin"/>
Contraseña:(*)	<input type="password" value="....."/>

[Mostrar todos Usuarios](#)

*Figura 67. Crear usuario Elaborado por: Eduardo Conrado*

## 5.2.6 Perfiles de usuarios

En la figura 68, por medio del menú principal, submenú administración, se accede a la lista de todos los perfiles de usuario del gimnasio.

Mostrar lista de perfiles de usuarios

Lista de Perfiles ingresados		
ID	Descripcion	Accion
1	SUPERADMINISTRADOR	<a href="#">Modificar</a>
2	ADMINISTRADOR	<a href="#">Modificar</a>
3	ENTRENADOR PERSONAL	<a href="#">Modificar</a>
4	RECEPCIONISTA	<a href="#">Modificar</a>
5	ADMINISTRACION	<a href="#">Modificar</a>
6	MEDICO	<a href="#">Modificar</a>

*Figura 68.* Mostrar Lista de Perfiles de Usuarios Elaborado por: Eduardo Conrado

### 5.2.6.1 Crear nuevo perfil

En la figura 69, en la parte inferior derecha de la figura anterior, se accede al enlace crear nuevo perfil.

Crear nuevo perfil

**Datos del Perfil**

IdPerfil:

Descripcion:(\*)

[Mostrar todos Perfiles](#)

*Figura 69.* Crear nuevo perfil Elaborado por: Eduardo Conrado

### 5.2.7 Información del cliente

En la figura 70, por medio del menú principal, se accede al submenú clientes desde donde se puede visualizar la lista de todos los clientes del gimnasio.

Mostrar lista de clientes

Modificar Clientes Ingresados			
ID	Nombres y Apellidos	Cedula	Accion
1	Mario Reyes Perez	1718903429	<a href="#">Modificar</a>
2	Sara Salazar Perez	1710542588	<a href="#">Modificar</a>
3	Cristina Segovia Tulipe	1710542589	<a href="#">Modificar</a>
4	Pedro Guerrero Santos	1710542599	<a href="#">Modificar</a>
5	Isabella Guerrero Santos	1718903429	<a href="#">Modificar</a>

*Figura 70.* Mostrar lista de clientes Elaborado por: Eduardo Conrado

En la figura 71, al final de la lista existen opciones para modificar la información del cliente.

Opciones de lista

Modificar Clientes Ingresados		
Apellidos	Cedula	Accion
	1718903429	<a href="#">Modificar</a>
	1710542588	<a href="#">Modificar</a>
	1710542589	<a href="#">Modificar</a>
	1710542599	<a href="#">Modificar</a>
	1718903429	<a href="#">Modificar</a>

*Figura 71.* Modificar clientes ingresados Elaborado por: Eduardo Conrado

### 5.2.7.1 Crear cliente

En la figura 72, por medio del enlace situado [Crear Nuevo Cliente](#) en la parte inferior derecha de la figura 72, se procede a ingresar los datos para crear un nuevo cliente del gimnasio.

Crear cliente

**Datos del Cliente**

Nombre:(*)	<input type="text"/>
Apellido Paterno:(*)	<input type="text"/>
Apellido Materno:	<input type="text"/>
Cedula:(*)	<input type="text"/>
Genero:(*)	<input checked="" type="radio"/> Masculino <input type="radio"/> Femenino
Fecha Nacimiento:(*)	<input type="text" value="dd/mm/aaaa"/>
Fecha Ingreso	12/07/2014
Telefono:(*)	<input type="text"/>
Celular:	<input type="text"/>
Email:	<input type="text"/>
Persona Contacto:(*)	<input type="text"/>
Telefono Contacto:(*)	<input type="text"/>

*Figura 72.* Ingreso de nuevo cliente Elaborado por: Eduardo Conrado

## 5.2.8 Información de especialidades

En la figura 5.10, por medio del menú principal, submenú administración, se accede a la información de las especialidades existentes en el gimnasio.

Mostrar lista de especialidades

Lista de Especialidades ingresadas		
ID	Descripcion	Accion
1	Ninguna	<a href="#">Modificar</a> <a href="#">Eliminar</a>
2	Medico General	<a href="#">Modificar</a> <a href="#">Eliminar</a>
3	Deportólogo	<a href="#">Modificar</a> <a href="#">Eliminar</a>
4	Traumatólogo	<a href="#">Modificar</a> <a href="#">Eliminar</a>
5	Cardiologo	<a href="#">Modificar</a> <a href="#">Eliminar</a>
6	Terapeuta Físico	<a href="#">Modificar</a> <a href="#">Eliminar</a>
7	Ortopedia	<a href="#">Modificar</a> <a href="#">Eliminar</a>
8	Nutriologo	<a href="#">Modificar</a> <a href="#">Eliminar</a>
9	Otro	<a href="#">Modificar</a> <a href="#">Eliminar</a>

*Figura 73.*Mostrar lista de especialidades Elaborado por: Eduardo Conrado

### 5.2.8.1 Crear nueva especialidad

En la figura 74, en la parte inferior derecha de la figura anterior, se accede al enlace crear nueva especialidad.

Crear nueva especialidad

Descripcion:(\*)

[Mostrar todos Especialidad](#)

[Guardar](#)

*Figura 74.* Crear nueva especialidad Elaborado por: Eduardo Conrado

### 5.2.9 Estados de una valoración física

En la figura 75, por medio del menú principal, submenú administración, se accede a la información de los tipos de estado que tiene una valoración física dentro del centro.

Mostrar estados de una valoración física

Lista de Estados de una Valoracion Fisica	
ID	Descripcion
1	Finalizado
2	Espera
3	Cancelado
4	Visita
5	Llego

*Figura 75.* Mostrar estados de una valoración física Elaborado por: Eduardo Conrado

### 5.2.10 Información de cobros

En la figura 76, por medio del menú principal, submenú administración, se accede a la información de los cobros que se hacen las facturas.

Mostrar lista de cobros

Lista de Cobros ingresados		
id	Descripcion	Accion
1	Valoracion	<a href="#">Modificar</a> <a href="#">Eliminar</a>
2	Entrenamiento	<a href="#">Modificar</a> <a href="#">Eliminar</a>
3	Terapia Fisica	<a href="#">Modificar</a> <a href="#">Eliminar</a>
4	Vendas	<a href="#">Modificar</a> <a href="#">Eliminar</a>
5	Otros	<a href="#">Modificar</a> <a href="#">Eliminar</a>
6	Extras	<a href="#">Modificar</a> <a href="#">Eliminar</a>

*Figura 76.*Mostrar lista de cobros Elaborado por: Eduardo Conrado

### 5.2.11 Crear nuevo tipo de cobro

En la figura 77, en la parte inferior derecha de la figura anterior, se accede al enlace crear nuevo cobro.

Crear nuevo tipo de cobro

**Datos del Cobro**

Descripcion:(\*)

[Mostrar todos Cobros](#)

*Figura 77. Crear nuevo tipo de cobro Elaborado por: Eduardo Conrado*

### 5.2.12 Información de signos vitales

En la figura 78, cuando se genera una valoración física, se ingresa la información de signos vitales.

Ingresar signos vitales

Atender Valoracion Fisica	
Hora inicio atencion:	<input type="text" value="20:31"/>
Signos Vitales	
Temperatura:	<input type="text" value="35"/>
SYS:	<input type="text" value="120"/>
DIA:	<input type="text" value="80"/>
Pulso:	<input type="text" value="70"/>
Peso:	<input type="text" value="120"/>
Frecuencia Respiratoria:	<input type="text" value="20"/>

*Figura 78. Ingresar signos vitales Elaborado por: Eduardo Conrado*

### 5.2.13 Información de área física

En la figura 79, dentro de una valoración física, se ingresa la información del área física que se va a entrenar.

Determinar área física

Examen Físico Regional	<input type="checkbox"/> Cabeza <input type="checkbox"/> Cuello <input type="checkbox"/> Torax <input type="checkbox"/> Abdomen <input type="checkbox"/> Pelvis <input type="checkbox"/> Extremidades
------------------------	--

*Figura 79.* Determinar área física Elaborado por: Eduardo Conrado

### 5.2.14 Información de tipo entrenamiento

En la figura 80, dentro de una valoración física, se ingresa la información del tipo de entrenamiento que va a realizar el cliente.

Tipo de entrenamiento

Tipo Entrenamiento	<input type="checkbox"/> AEROBICO <input type="checkbox"/> ANAEROBICO <input type="checkbox"/> RESISTENCIA <input type="checkbox"/> FUERZA <input type="checkbox"/> VELOCIDAD
--------------------	---

*Figura 80.* Tipo de entrenamiento Elaborado por: Eduardo Conrado

### 5.2.15 Ingresar diagnóstico

En la figura 81, en una valoración física, se ingresa la información de diagnóstico (confirmado o por confirmar) de la valoración física realizada al cliente.

Ingresar diagnóstico

Diagnostico:	<input type="text"/>
--------------	----------------------

*Figura 81. Ingresar diagnóstico Elaborado por: Eduardo Conrado*

### 5.2.16 Ingresar tratamiento

En la figura 82, dentro de una valoración física, se ingresa la información del tratamiento de la valoración física realizada al cliente.

Ingresar tratamiento

Tratamiento:	<input type="text"/>
--------------	----------------------

*Figura 82. Ingresar tratamiento Elaborado por: Eduardo Conrado*

### 5.2.17 Ingresar plan de entrenamiento

En la figura 83, dentro de una valoración física, se ingresa la información del tratamiento de la valoración física realizada al cliente.

Ingresar plan de entrenamiento

Plan de Entrenamiento	<input type="text"/>
-----------------------	----------------------

*Figura 83. Ingresar plan de entrenamiento Elaborado por: Eduardo Conrado*

### 5.2.18 Información de datos personales – nomenclatura antecedentes familiares

En la figura 84, dentro de la información de datos personales del cliente del gimnasio, se encuentra la información de los antecedentes familiares del cliente. La nomenclatura y descripción se puede visualizar accediendo desde submenú administración.

Lista de antecedentes familiares

Lista de Antecedentes Personales ingresados			
ID	Descripcion	Cod	Accion
1	CARDIOPATIA	AF1	<a href="#">Modificar</a>
2	DIABETES	AF2	<a href="#">Modificar</a>
3	ENF. C MUSCULAR	AF3	<a href="#">Modificar</a>
4	HIPERTENSION	AF4	<a href="#">Modificar</a>
5	CANCER	AF5	<a href="#">Modificar</a>
6	TUBERCOLOSIS	AF6	<a href="#">Modificar</a>
7	ENF. MENTAL	AF7	<a href="#">Modificar</a>
8	ENF. INFECCIOSA	AF8	<a href="#">Modificar</a>
9	MAL FORMACION	AF9	<a href="#">Modificar</a>
10	OTRO	AF10	<a href="#">Modificar</a>

*Figura 84.*Lista de antecedentes familiares Elaborado por: Eduardo Conrado

### 5.2.19 Nomenclatura tipo de entrenamiento

En la figura 85, en cada valoración física del gimnasio, se procede a realizar la revisión de los tipos de entrenamiento. Se registra la forma en que va a trabajar el cliente. La nomenclatura y descripción se puede visualizar accediendo desde submenú administración.

Lista tipo entrenamiento

Lista Tipo de Entrenamiento			
ID	Descripción	Código	Acción
1	AEROBICO	ROS1	<a href="#">Modificar</a>
2	ANAEROBICO	ROS2	<a href="#">Modificar</a>
3	RESISTENCIA	ROS3	<a href="#">Modificar</a>
4	FUERZA	ROS4	<a href="#">Modificar</a>
5	VELOCIDAD	ROS5	<a href="#">Modificar</a>

*Figura 85.* Lista tipo entrenamiento Elaborado por: Eduardo Conrado

### 5.2.20 Reportes

Mediante la aplicación web, se puede acceder a los reportes generados con los datos ingresados dentro la aplicación web.

#### 5.2.20.1 Tipos de reportes

##### 5.2.20.1.1 Con información del cliente



En la figura 87, para acceder a la información del cliente, se procede a realizarla la búsqueda del cliente por su id.

Buscar información por id de cliente

**Informacion Cliente**

Parametros de busqueda

Id cliente  
 Cedula  
 Nombre

Ingresar valor:(\*)

*Figura 87.* Buscar información por id de cliente Elaborado por: Eduardo Conrado

En la figura 88, se despliega la información del cliente.

Reporte búsqueda de cliente

 **Gimnasio Absolut Gym**

Resultado Busqueda de Usuario

**RECEPCIONISTA**

Ninguna

idUsuario	<b>1</b>
Nombre	<b>Paola</b>
Apellido	<b>Andrade</b>
Email	<b>paola@hotmail.com</b>
Telefono	<b>2232951</b>
Celular	<b>095754590</b>
Cargo	<b>Recepcionista</b>
NombreUsuario	<b>paola</b>

*Figura 88.* Reporte búsqueda de cliente Elaborado por: Eduardo Conrado

### 5.2.20.1.2 Con información del usuario

En la figura 89, para acceder a la información del usuario, se procede a realizarla la búsqueda del usuario por su id.

Buscar información por id del usuario

**Desplegar información de Usuario**

Parametros de búsqueda

Id usuario  
 Nombre

Ingresar valor:(\*)

*Figura 89.* Buscar información por id del usuario Elaborado por: Eduardo Conrado

En la figura 90, se despliega la información del usuario.

Reporte búsqueda de usuario

 **Gimnasio Absolut Gym**

Resultado Búsqueda de Usuario

**RECEPCIONISTA**

Ninguna

idUsuario	<b>1</b>
Nombre	<b>Paola</b>
Apellido	<b>Andrade</b>
Email	<b>paola@hotmail.com</b>
Telefono	<b>2232951</b>
Celular	<b>095754590</b>
Cargo	<b>Recepcionista</b>
NombreUsuario	<b>paola</b>

*Figura 90.* Reporte búsqueda de usuario Elaborado por: Eduardo Conrado

## CONCLUSIONES

En el proceso de desarrollo del sistema web conseguí incrementar mis habilidades para la recopilación y modelado de datos utilizando el sistema de gestión de bases de datos relacional MySQL.

El sistema web interno del gimnasio fue realizado en Netbeans. La herramienta permite el desarrollo de soluciones a los requerimientos presentados por los usuarios. Permittiéndose así satisfacer las necesidades de registro de información de los clientes.

A través del uso de la metodología XP y las historias de usuario se estableció las prioridades de las características que necesita el cliente para conseguir el producto final de software.

El software libre proporciona independencia tecnológica en el desarrollo de nuevos productos. El uso software libre usado permite reducir costes por licencia y uso. La independencia tecnológica del software libre permite mayor flexibilidad en el manejo de herramientas de desarrollo.

Las herramientas para testear aplicaciones web como Apache JMeter proveen de componentes que permiten visualizar los datos obtenidos en forma gráfica.

A través del uso del sistema web interno el gimnasio se mejora el desempeño de los procesos de registro de clientes, valoraciones físicas, el servicio médico cuenta con herramientas tecnológicas que facilitan su trabajo diario.

## **RECOMENDACIONES**

Es necesario capacitar a cada uno de los usuarios sobre el manejo del sistema web de tal manera que se realice un buen uso del software que permita garantizar los datos que se almacenan en la base de datos y la información que se presenta en el sistema.

El análisis de los requerimientos del usuario permitió definir el flujo de procesos. Las mejoras en el futuro deberán ser realizadas y basadas en el análisis realizado. Cada cambio deberá ser desarrollado y testeado para lograr la compatibilidad con lo desarrollado en el presente proyecto.

Las herramientas para testear aplicaciones web deben ser utilizadas más frecuentemente entre los desarrolladores para evaluar que los procesos se realicen correctamente. Los desarrolladores necesitarán acceder datos específicos esenciales para ejecutar los casos de prueba relacionados.

Se recomienda desarrollar y crear métricas para observar la calidad del software en sus estados posteriores y comparar los perfeccionamientos con versiones previas. Esto ayudará a aumentar el valor y conocimiento de los procesos que realiza el software para así lograr su mejora continua.

Para mantener la seguridad de la información y la operación del sistema se recomienda realizar respaldo periódico de la base de datos.

## LISTA DE REFERENCIAS

- Letelier, P. (2006). *eXtreme Programming (XP)*. *Departamento de Sistemas Informáticos y Computación (DSIC)*, 2.
- Apache. (2014). *Sitio oficial de Apache*.
- Bevan, N. (1999). *Quality in Use: Meeting User Needs for Quality*. *Journal of System and Software*. Serco Usability Services.
- Catalan, M. (2001). Metodologías de evaluación de Interfaces Gráficas de Usuario. *RCLIS*, 36.
- Conrado, E. (2014). *Información obtenida para el Sistema*.
- DNTI. (2011). Plan de Contingencia. *Dirección Nacional de Tecnologías de la Información*, 7.
- DNX. (01 de 01 de 2005). *microsoft.com*. Obtenido de Usabilidad y Experiencia de Usuario:  
[http://www.microsoft.com/spain/empresas/guias/usabilidad/experiencia\\_usuario.m](http://www.microsoft.com/spain/empresas/guias/usabilidad/experiencia_usuario.m)
- GC. (2014). *Redes Locales*. Canarias: Gobierno de las Canarias.
- Gym, A. (2014). *Información empresarial*. Quito.
- Joskowicz, J. (10 de 02 de 2008). *Reglas y prácticas en eXtreme Programming*. Obtenido de <http://iie.fing.edu.uy/>: <http://iie.fing.edu.uy/~josej/docs/XP%20-%20Jose%20Joskowicz.pdf>
- Lamarca, M. (2014). *DHTML*. Madrid.
- López, V. (2007). Soluciones Software Libre. *Universidad Autónoma*, 13.
- Magaña, M. (2014). *Tecnología JSP (Java Server Pages)*.
- Montero, Y., & Fernández, F. (2 de Septiembre de 2013). *Nosolousabilidad*. Recuperado el 04 de mayo de 2014, de Diseño Inclusivo: Marco Metodológico para el Desarrollo de Sitios Web Accesibles: [http://www.nosolousabilidad.com/articulos/diseno\\_inclusivo.htm](http://www.nosolousabilidad.com/articulos/diseno_inclusivo.htm)
- Netbeans. (2014). *Sitio web oficial*.

- Oracle. (25 de 05 de 2014). *Java API Specifications*. Recuperado el 25 de 05 de 2014, de Oracle Technology Network:  
<http://www.oracle.com/technetwork/java/api-141528.html>
- SENA. (2011). *Manual TI*. Medellin: Secretaria Nacional de Aprendizaje.
- Torrijos , R. (2014). *Desarrollo de Aplicaciones Web con JSP y XML*. Programacion Net.
- UTS. (2013). MySQL. *Universidad Tecnológica del Sur de Sonora*, 2.

# ANEXOS

## Anexo 1. Formulario para valoración física

<b>1 MOTIVO DE VALORACIÓN FÍSICA</b>												
<b>2 ANTECEDENTES PERSONALES</b>												
<b>3 ANTECEDENTES FAMILIARES</b>												
1. CARDIOPATÍA	2. DIABETES	3. ENF. C. VASCULAR	4. HIPER. TENSION	5. CÁNCER	6. TUBERCULO. SIS	7. ENF. MENTAL	8. ENF. INFECCIOSA	9. MAL. FORMACIÓN	10. OTRO			
<b>4 ESTADO VALORACIÓN ACTUAL</b>												
<b>5 REVISIÓN ACTUAL DE ÓRGANOS Y SISTEMAS</b>												
CP = CON EVIDENCIA DE PATOLOGÍA: MARCAR "X" Y DESCRIBIR ABAJO ANOTANDO EL NÚMERO Y LETRA.						SP = SIN EVIDENCIA DE PATOLOGÍA: MARCAR "X" Y NO DESCRIBIR.						
1	C P	S P	3	C P	S P	5	C P	S P	7	C P	S P	
ÓRGANOS DE LOS SENTIDOS			CARDIO VASCULAR			GENTAL			MÚSCULO ESQUELÉTICO			
2			4			6			8			
RESPIRATORIO			DIGESTIVO			URINARIO			ENDOCRINO			
									10			
									HEMO LINFÁTICO			
									NERVIOSO			
<b>5 REVISIÓN ACTUAL DE ÓRGANOS Y SISTEMAS</b>												
CP = CON EVIDENCIA DE PATOLOGÍA: MARCAR "X" Y DESCRIBIR ABAJO ANOTANDO EL NÚMERO Y LETRA.						SP = SIN EVIDENCIA DE PATOLOGÍA: MARCAR "X" Y NO DESCRIBIR.						
1	C P	S P	3	C P	S P	5	C P	S P	7	C P	S P	
ÓRGANOS DE LOS SENTIDOS			CARDIO VASCULAR			GENTAL			MÚSCULO ESQUELÉTICO			
2			4			6			8			
RESPIRATORIO			DIGESTIVO			URINARIO			ENDOCRINO			
									10			
									HEMO LINFÁTICO			
									NERVIOSO			
<b>6 SIGNOS VITALES Y ANTROPOMETRIA</b>												
FECHA DE MEDICION												
TEMPERATURA °C												
PRESION ARTERIAL	//			//			//			//		
PULSO / min	//			//			//			//		
FRECUENCIA RESPIRATORIA	//			//			//			//		
PESO / Kg	//			//			//			//		
TALLA / cm	//			//			//			//		
<b>7 EXAMEN FISICO REGIONAL</b>												
CP = CON EVIDENCIA DE PATOLOGÍA: MARCAR "X" Y DESCRIBIR ABAJO ANOTANDO EL NÚMERO Y LETRA CORRESPONDIENTES.						SP = SIN EVIDENCIA DE PATOLOGÍA: MARCAR "X" Y NO DESCRIBIR.						
1	C P	S P	2	C P	S P	3	C P	S P	4	C P	S P	
CABEZA			CUELLO			TÓRAX			ABDOMEN			
									5			
									PELVIS			
									6			
									EXTREMIDADES S			
<b>8 DIAGNOSTICO</b>												
1				3								
2				4								
<b>9 PLANES DE TRATAMIENTO</b>												
PLANES DE DIAGNOSTICO, TERAPÉUTICO Y EDUCACIONAL												
<b>10 PLANES DE ENTRENAMIENTO</b>												
PLANES DE DIAGNOSTICO, TERAPÉUTICO Y EDUCACIONAL												

**Anexo 2. Factura Absolut Gym**

<p><b>Fitness Center ABSOLUT GYM</b></p> <p>Dir. Calle Pedro Capiro S/N y calle Lexo Bruis Tel. 02-3284-182 Quito - Ecuador</p>	<p><b>R.U.C. 1790848522001</b> Aut. S.R.I. 1110098976</p> <p><b>FACTURA</b> S001-001 <b>00000726</b></p> <p><b>SR. DAVID BALLESTEROS</b></p>										
<p>Fecha: _____ RUC/C.C.: _____</p> <p>Sr.(s): _____</p> <p>Dirección: _____ Tel: _____</p>											
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%;">DETALLE</th> <th style="width: 10%;">CANT.</th> <th style="width: 20%;">V. UNITARIO</th> <th style="width: 20%;">V. TOTAL</th> </tr> </thead> <tbody> <tr> <td style="height: 150px;"> </td> <td> </td> <td> </td> <td> </td> </tr> </tbody> </table>				DETALLE	CANT.	V. UNITARIO	V. TOTAL				
DETALLE	CANT.	V. UNITARIO	V. TOTAL								
		SUBTOTAL \$									
		IVA 0 % \$									
		IVA 12 % \$									
		TOTAL \$									
FIRMA AUTORIZADA	FIRMA CLIENTE										
CONVE GRAF Los Freanos De10-56 y 6 de Diciembre / Tel.: 02 3400 751 - R.U.C.: 1711901437001 AUT.SRI 5702											

### **Anexo 3. Glosario de términos**

#### **A**

- ActiveX** Son controles que sirven para definir componentes de software reusables de forma independiente del lenguaje de programación. Las aplicaciones de software son diseñadas para proveer funcionalidades a la página del programa sin tener que escribir código completo para la función, inclusive puede ser incluido dentro de otras aplicaciones.
- Apache** Apache es programa de servidor HTTP Web de código abierto (open source). Su desarrollo empezó en 1995 y actualmente es uno de los servidores web más utilizados en la red. Usualmente corre en UNIX, Linux, BSD y Windows. Es un poderoso paquete de servidor web con muchos módulos que se le pueden agregar y que se consiguen gratuitamente en el Internet. Uno de sus competidores es Microsoft IIS.  
<http://www.apache.org>
- Apache Derby** Apache Derby es un sistema gestor de base de datos relacional escrito en Java que puede ser empotrado en aplicaciones Java y utilizado para procesos de transacciones online. Tiene un tamaño de 2 MB de espacio en disco. Inicialmente distribuido como IBM Cloudscape, Apache Derby es un proyecto open source licenciado bajo la Apache 2.0 License. Actualmente se distribuye como Sun Java DB.
- Apache Software Foundation** Apache Software Foundation (ASF) es una organización no lucrativa (en concreto, una fundación) creada para dar soporte a los proyectos de software bajo la denominación Apache, incluyendo el popular servidor HTTP Apache. La ASF se formó a partir del llamado Grupo Apache y fue registrada en Delaware (Estados Unidos), en junio de 1999.

- Apache Tomcat** Apache Tomcat (también llamado Jakarta Tomcat o simplemente Tomcat) funciona como un contenedor de *servlets* desarrollado bajo el proyecto Jakarta en la Apache Software Foundation. Tomcat implementa las especificaciones de los *servlets* y de JavaServer Pages (JSP) de Sun Microsystems. Tomcat es un servidor web con soporte de *servlets* y JSPs. Tomcat no es un servidor de aplicaciones, como JBoss o JOnAS. Incluye el compilador Jasper, que compila JSPs convirtiéndolas en *servlets*. El motor de *servlets* de Tomcat a menudo se presenta en combinación con el servidor web Apache.
- API Java** Es una interfaz de programación de aplicaciones que consiste en un conjunto de clases, funciones y procedimientos de la programación orientada a objetos para desarrollar aplicaciones Java. Utiliza librerías que pueden ser utilizadas por otro software como una capa de abstracción.
- Applets Java** Los applets generalmente están embebidos en páginas web y pueden ser ejecutados directamente desde un navegador con soporte para Java utilizando la Java Virtual Machine (JVM), o en el AppletViewer de Sun. El applet se descarga en el navegador web y comienza a ejecutarse. Esto permite crear programas que cualquier usuario puede ejecutar con tan solo cargar la página web en su navegador.
- ASP** Active Server Pages es una tecnología de Microsoft que utiliza lenguajes de programación que se ejecutan del lado del servidor del lado del servidor, mediante páginas web generadas dinámicamente utilizando Internet Information Services (IIS) u otro servidor web compatible.

## C

### **Catalina**

Catalina es el nombre del contenedor de servlets del Jakarta Tomcat desde la versión 4x. Fue desarrollado bajo el Proyecto Jakarta de la Apache Software Foundation. Tomcat implementa las especificaciones de Sun Microsystems para servlets y Java Server Pages (JSP), las cuales son importantes tecnologías web basadas en Java.

### **CGI**

Common Gateway Interface o Interfaz de entrada común es una especificación que permite la transferencia de información la tecnología World Wide Web y una aplicación CGI. Estándar para transferir, aceptar y devolver los datos entre un servidor web y una aplicación externa cuyo resultado final de la ejecución son objetos MIME. Las aplicaciones pueden estar escritas en cualquier lenguaje que soporte el servidor web, si se ejecutan en el servidor reciben el nombre de CGIs.

### **Código abierto**

Código abierto es el término con el que se conoce al software distribuido y desarrollado libremente. El código abierto tiene un punto de vista más orientado a los beneficios prácticos de poder acceder al código, que a las cuestiones éticas y morales las cuales se destacan en el software libre.

### **CSS**

Cascading style sheets (CSS) son hojas de estilo en cascada utilizados para dar estilo y formato de múltiples a documentos HTML y XHTML, puede usarse dentro del documento html o en un documento separado que define la estructura de la presentación. La regla tiene dos partes: un selector y la declaración. La declaración está compuesta por una propiedad y el valor que se le asigne. El Consortium W3C regula las especificaciones de las hojas de estilo que se usan

como estándar en los navegadores.

## **D**

### **DCU**

Es una metodología que sigue el Modelo de Proceso de la Ingeniería de la Usabilidad y de la Accesibilidad (MPIu+a). Al usar esta metodología hace involucrar a los usuarios desde el inicio del desarrollo. Es una metodología que implica una estrecha colaboración entre los diseñadores y programadores y los usuarios a los que va destinado el sistema. Esta metodología trata de garantizar que la aplicación desarrollada sea usable.

## **E**

### **Enterprise JavaBeans**

Los Enterprise JavaBeans (también conocidos por sus siglas EJB) son una de las API que forman parte del estándar de construcción de aplicaciones empresariales J2EE (ahora JEE 6.0) de Oracle Corporation (inicialmente desarrollado por Sun Microsystems).

### **Embedded**

Son scripts que permiten hacer más potente a un lenguaje de programación, en las páginas web se desarrollan en lenguaje HTML que puede ser embeberse códigos JavaScript o PHP para aumentar las posibilidades ya que son diseñados para cubrir necesidades específicas.

## **F**

### **Framework**

La palabra inglesa "framework" define, en términos generales, un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar. Representa una arquitectura de software que

modela las relaciones generales de las entidades del dominio, y provee una estructura y una especial metodología de trabajo, la cual extiende o utiliza las aplicaciones del dominio.

## **G**

### **Glassfish**

GlassFish es un servidor de aplicaciones de software libre desarrollado por Sun Microsystems, compañía adquirida por Oracle Corporation, que implementa las tecnologías definidas en la plataforma Java EE y permite ejecutar aplicaciones que siguen esta especificación. La versión comercial es denominada Oracle GlassFish Enterprise Server (antes Sun GlassFish Enterprise Server). Es gratuito y de código libre, se distribuye bajo un licenciamiento dual a través de la licencia CDDL y la GNU GPL. GlassFish está basado en el código fuente donado por Sun y Oracle Corporation, éste último proporcionó el módulo de persistencia TopLink. GlassFish tiene como base al servidor Sun Java System Application Server de Oracle Corporation, un derivado de Apache Tomcat.

### **GNU**

La Licencia Pública General de GNU o más conocida por su nombre en inglés GNU General Public License o simplemente sus siglas del inglés GNU GPL, es una licencia creada por la Free Software Foundation en 1989 (la primera versión), y está orientada principalmente a proteger la libre distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

## **I**

**ISO** Es el organismo encargado de promover el desarrollo de normas internacionales de fabricación (tanto de productos como de servicios), comercio y comunicación para todas las ramas industriales a excepción de la eléctrica y la electrónica. Su función principal es la de buscar la estandarización de normas de productos y seguridad para las empresas u organizaciones (públicas o privadas) a nivel internacional.

## **J**

**Java** Java es un lenguaje de programación de alto nivel orientado a objetos, desarrollado por James Gosling en 1995. El lenguaje en sí mismo toma mucha de su sintaxis de C, Cobol y Visual Basic, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria. La memoria es gestionada mediante un recolector de basura.

**Java EE** Es una plataforma de programación parte de la Plataforma Java para desarrollar y ejecutar software de aplicaciones en el lenguaje de programación Java con arquitectura de N capas distribuidas y que se apoya ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones.

**Java SE** Java Platform, Standard Edition o Java SE es una colección de APIs del lenguaje de programación Java útiles para muchos programas de la Plataforma Java. La Plataforma Java 2, Enterprise Edition incluye todas las clases en el Java SE, además de algunas de las cuales son útiles para programas que se ejecutan en servidores sobre workstations.

**JasperReports Server** Es un servidor de gran alcance para presentación de informes flexible y ligera. Genera, organiza, entrega informes y dashboards interactivos a los usuarios.

**Javascript** Es un lenguaje interpretado en el cliente por el navegador al momento de cargarse la página, es multiplataforma, orientado a eventos con manejo de objetos, cuyo código se incluye directamente en el mismo documento HTML. La principal característica es que permite modificar páginas web en tiempo real sin usar CGI's y construir sitios Web más interactivos, utilizando contenido dinámico.

**JDBC** Java Database Connectivity, más conocida por sus siglas JDBC , es una API que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java, independientemente del sistema operativo donde se ejecute o de la base de datos a la cual se accede, utilizando el dialecto SQL del modelo de base de datos que se utilice.

**JSF** JavaServer Faces (JSF) es una tecnología y framework para aplicaciones Java basadas en web que simplifica el desarrollo de interfaces de usuario en aplicaciones Java EE. JSF usa JavaServer Pages (JSP) como la tecnología que permite hacer el despliegue de las páginas, pero también se puede acomodar a otras tecnologías como XUL.

**JSP** JavaServer Pages (JSP) es una tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML. Las JSP permiten la utilización de código Java mediante scripts. La tecnología JSP permite crear aplicaciones

web para que sean ejecutados en varios tipos de servidores web. Un JSP es una página Web que contiene etiquetas especiales y código Java incrustado, mientras que un servlet es un programa que recibe peticiones y genera a partir de ellas una página web.

## **JUnit**

JUnit es un conjunto de bibliotecas creadas por Erich Gamma y Kent Beck que son utilizadas en programación para hacer pruebas unitarias de aplicaciones Java. JUnit es un conjunto de clases (framework) que permite realizar la ejecución de clases Java de manera controlada, para poder evaluar si el funcionamiento de cada uno de los métodos de la clase se comporta como se espera. Es decir, en función de algún valor de entrada se evalúa el valor de retorno esperado; si la clase cumple con la especificación, entonces JUnit devolverá que el método de la clase pasó exitosamente la prueba; en caso de que el valor esperado sea diferente al que regresó el método durante la ejecución, JUnit devolverá un fallo en el método correspondiente.

## **L**

### **Licencia Apache**

La licencia Apache (Apache License o Apache Software License para versiones anteriores a 2.0) es una licencia de software libre creada por la Apache Software Foundation (ASF). La licencia Apache requiere la conservación del aviso de copyright y el disclaimer, pero no es una licencia copyleft, ya que no requiere la redistribución del código fuente cuando se distribuyen versiones modificadas.

## **M**

**Microsoft IIS** Es un servidor web y un conjunto de servicios para el sistema operativo Microsoft Windows. El servicio convierte a una PC en un servidor web para Internet o una intranet, lo que permite publicar páginas web tanto local como remotamente. Los módulos procesan distintos tipos de páginas como Active Server Pages (ASP) y ASP.NET

**Middleware** Middleware es un software que asiste a una aplicación para interactuar o comunicarse con otras aplicaciones, software, redes, hardware y/o sistemas operativos. Éste simplifica el trabajo de los programadores en la compleja tarea de generar las conexiones que son necesarias en los sistemas distribuidos. De esta forma se provee una solución que mejora la calidad de servicio, seguridad, envío de mensajes, directorio de servicio, etc.

## N

**Netbeans** NetBeans es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo. NetBeans IDE es un producto libre y gratuito sin restricciones de uso. NetBeans es un proyecto de código abierto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento, y con cerca de 100 socios en todo el mundo. Sun Microsystems fundó el proyecto de código abierto NetBeans en junio de 2000 y continúa siendo el patrocinador principal de los proyectos.

## O

**Open Source Software** Software de código abierto es software cuyo código de programación subyacente está a disposición de los usuarios para que puedan leerlo, modificarlo y crear nuevas versiones del software que incorpora los cambios. Hay muchos tipos de software de código abierto, difieren en el plazo de concesión de licencias en virtud del cual sea (modificado) y las copias del código fuente puedan (o deban ser) redistribuido.

## P

**Plataforma Java** La plataforma Java es el nombre de un entorno o plataforma de computación originaria de Sun Microsystems, capaz de ejecutar aplicaciones desarrolladas usando el lenguaje de programación Java u otros lenguajes que compilen a bytecode y un conjunto de herramientas de desarrollo. En este caso, la plataforma no es un hardware específico o un sistema operativo, sino más bien una máquina virtual encargada de la ejecución de las aplicaciones, y un conjunto de bibliotecas estándar que ofrecen una funcionalidad común.

**Plugin Java** Es una tecnología de Java Runtime Environment que permite establecer la conexión entre los exploradores más conocidos y la plataforma Java. La conexión permite que los applets de sitios Web se ejecuten en el explorador en el escritorio.

## S

**Script** Son archivos de texto que contienen un conjunto de instrucciones que son interpretados línea por línea para su ejecución, no son programas compilados que se convierten en un archivo binario para ejecutarse. El soporte de scripts de HTML es independiente del lenguaje de scripts. Los scripts son usados para realizar diversas tareas como combinar componentes, interactuar con el sistema operativo o con el usuario.

**Servidor Apache** **HTTP** El servidor HTTP Apache es un servidor web HTTP de código abierto, para plataformas Unix (BSD, GNU/Linux, Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.12

**Servlets** Es un programa realizado en lenguaje Java el cual es ejecutado en un servidor Web. Los Servlets pueden responder a cualquier tipo de solicitudes, recibiendo y atendiendo solicitudes de los clientes Web a través de HTTP, ejecutados en forma dinámica a partir de los parámetros de la petición que envíe el navegador web.

**Sun Microsystems** Sun Microsystems fue una empresa informática que se dedicaba a vender ordenadores, componentes informáticos, software y servicios informáticos. Fue adquirida en el año 2009 por Oracle Corporation, anteriormente parte de Silicon Valley, fabricante de semiconductores y software.

## T

**Thread** Un hilo es básicamente una tarea que puede ser ejecutada en paralelo

con otra tarea. Los hilos de ejecución que comparten los mismos recursos, sumados a estos recursos, son en conjunto conocidos como un proceso. En sistemas operativos, un hilo de ejecución, hebra o subproceso es la unidad de procesamiento más pequeña que puede ser planificada por un sistema operativo. La creación de un nuevo hilo es una característica que permite a una aplicación realizar varias tareas a la vez (concurrentemente). Los distintos hilos de ejecución comparten una serie de recursos tales como el espacio de memoria, los archivos abiertos, situación de autenticación, etc. Esta técnica permite simplificar el diseño de una aplicación que debe llevar a cabo distintas funciones simultáneamente.

## **X**

### **XML**

(eXtensible Markup Language), lenguaje de marcas extensible, es un lenguaje de marcas desarrollado por el World Wide Web Consortium (W3C). XML proviene de un lenguaje inventado por IBM en los años setenta, llamado GML (Generalized Markup Language), que surgió por la necesidad que tenía la empresa de almacenar grandes cantidades de información. Este lenguaje gustó a la ISO, por lo que en 1986 trabajaron para normalizarlo, creando SGML (Standard Generalized Markup Language), capaz de adaptarse a un gran abanico de problemas. A partir de él se han creado otros sistemas para almacenar información. A diferencia de otros lenguajes XML da soporte a bases de datos, siendo útil cuando varias aplicaciones se deben comunicar entre sí o integrar información.

### **XUL**

XUL (acrónimo de XML-based User-interface Language, lenguaje basado en XML para la interfaz de usuario) es la aplicación de XML a la descripción de la interfaz de usuario en el navegador Mozilla. XUL no es un estándar. La mejor fuente para encontrar material de referencia sobre XUL son páginas especializadas así como libros

técnicos. La principal ventaja de XUL es que aporta una definición de interfaces GUI simple y portable. Esto reduce el esfuerzo empleado en el desarrollo de software. Otras aplicaciones aparte de Mozilla usan este lenguaje para sus interfaces de usuario. Algunas de ellas usan JavaScript para su lógica. Las aplicaciones XUL tienen la ventaja de poder correr en distintos sistemas operativos.