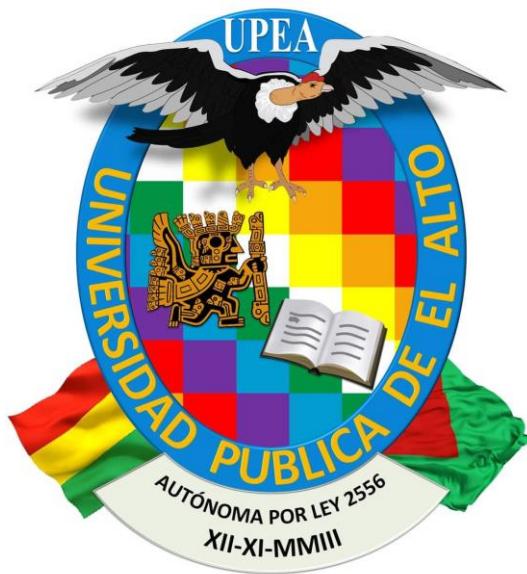


UNIVERSIDAD PÚBLICA DE EL ALTO

CARRERA INGENIERÍA DE SISTEMAS



PROYECTO DE GRADO

**“SISTEMA DE INFORMACIÓN WEB DE CONTROL
ADMINISTRATIVO”**

CASO: “GIMNASIO WINNER”

Para Optar al Título de Licenciatura en Ingeniería de Sistemas

MENCIÓN: INFORMÁTICA Y COMUNICACIONES

Postulante: Estanislao Yahuasi Apaza

Tutor Metodológico: Ing. Dionicio Henry Pacheco Rios

Tutor Revisor: M.Sc. Juan Marcos Miranda Nina

Tutor Especialista: Lic. Freddy Salgueiro Trujillo

EL ALTO – BOLIVIA

2020

INDICE

	Pág.
1 MARCO PRELIMINAR.....	1
1.1 INTRODUCCION.....	1
1.2 ANTECEDENTES	1
1.2.1 ANTECEDENTES INSTITUCIONALES	1
1.2.2 ANTECEDENTES INTERNACIONALES	2
1.2.3 ANTECEDENTES NACIONALES	3
1.3 PLANTEAMIENTO DEL PROBLEMA	4
1.3.1 PROBLEMA PRINCIPAL.....	4
1.3.2 PROBLEMAS SECUNDARIOS	4
1.4 OBJETIVOS.....	5
1.4.1 OBJETIVO GENERAL.....	5
1.4.2 OBJETIVO ESPECIFICO.....	5
1.5 JUSTIFICACION	6
1.5.1 JUSTIFICACIÓN TÉCNICA	6
1.5.2 JUSTIFICACIÓN ECONÓMICA	6
1.5.3 JUSTIFICACIÓN SOCIAL.....	6
1.6 METODOLOGIA	6
1.6.1 METODOLOGIA UWE	6
1.7 HERRAMIENTAS.....	8
1.7.1 LENGUAJE DE PROGRAMACIÓN PHP	8
1.7.1.1 Características.....	9
1.7.1.2 Ventajas	9
1.7.1.3 Desventajas	9
1.7.2 HTML	10

1.7.2.1	Ventajas	10
1.7.2.2	Desventajas	11
1.7.3	FRAMEWORK BootStrap	11
1.7.3.1	Ventajas	11
1.7.3.2	Desventajas	12
1.7.4	FRAMEWORK PHALCON.....	12
1.7.4.1	Características.....	13
1.7.5	GESTOR DE BASES DE DATOS.....	13
1.7.5.1	Características.....	14
1.7.5.2	Ventajas	14
1.7.5.3	Desventajas	15
1.8	LIMITES Y ALCANCES	15
1.8.1	LIMITES	15
1.8.2	ALCANCES	15
1.9	APORTES	16
2	MARCO TEÓRICO	17
2.1	INTRODUCCIÓN.....	17
2.2	DEFINICIÓN DE TÉRMINOS WEB	17
2.2.1	SISTEMA	17
2.2.2	SISTEMA DE INFORMACIÓN	18
2.2.2.1	Características de Sistema de Información.....	19
2.2.2.2	Componentes de un Sistema de Información.	20
2.2.3	SISTEMA WEB	20
2.2.4	SISTEMA DE CONTROL.....	21
2.2.5	ADMINISTRACIÓN	21

2.2.6	SISTEMA WEB DE ADMINISTRACIÓN	22
2.2.7	SISTEMA WEB	22
2.3	INGENIERÍA DE REQUERIMIENTO.....	23
2.3.1	REQUISITOS FUNCIONALES Y NO FUNCIONALES.....	26
2.4	METODOLOGÍA DE DESARROLLO.....	26
2.5	METODOLOGÍA UWE	27
2.5.1	CARACTERÍSTICAS UWE	30
2.5.2	MODELOS DE UWE	30
2.5.2.1	Modelo de Requerimientos	31
2.5.2.2	Modelo Conceptual	32
2.5.2.3	Modelo de Navegación.....	33
2.5.2.4	Modelo de Presentación	35
2.5.2.5	Modelo de Proceso.....	37
2.5.3	FACES O ETAPAS DE UWE	37
2.5.3.1	Análisis de Requisitos	37
2.5.3.2	Diseño del Sistema.....	38
2.5.3.3	Codificación del Software.....	38
2.5.3.4	Pruebas	38
2.5.3.5	Instalación o Fase de Implementación	38
2.5.3.6	Mantenimiento	38
2.5.4	DIAGRAMAS DE UWE	38
2.5.4.1	Diagrama de Clase	38
2.5.4.2	Diagrama de Objetos.....	39
2.5.4.3	Diagramas de Caso de Uso	39
2.5.4.4	Diagramas de Secuencia	40

2.5.4.5 Diagramas de Colaboración	41
2.5.4.6 Diagrama de Estado	42
2.5.4.7 Diagrama de Actividad	42
2.5.4.8 Diagrama de Componentes	43
2.6 INGENIERIA DE SOFTWARE	44
2.6.1 PROCESO DE SOFTWARE	46
2.7 HERRAMIENTAS.....	48
2.7.1 LENGUAJE DE PROGRAMACIÓN	48
2.7.1.1 Php	48
2.7.1.2 Html	49
2.7.2 FRAMEWORK	49
2.7.2.1 BootStrap.....	49
2.7.2.2 Jquery	51
2.7.2.3 Phalcon	53
2.7.3 GESTOR DE BASES DE DATOS MYSQL	54
2.7.4 SERVIDOR APACHE.....	55
2.7.5 ARQUITECTURA MVC	56
2.7.5.1 Modelo.....	57
2.7.5.2 Vista.....	57
2.7.5.3 Controlador.....	57
2.8 CALIDAD DE SOFTWARE	57
2.8.1 MÉTRICAS DE MCCALL.....	58
2.8.1.1 Operación del Producto	58
2.8.1.2 Revisión del Producto.....	59
2.8.1.3 Transición del Producto.....	59

2.9	ESTIMACIÓN DE COSTOS DEL PROYECTO	59
2.9.1	MODELO DE COSTO COCOMO II.....	59
2.9.2	CARACTERÍSTICAS PRINCIPALES.....	61
2.10	SEGURIDAD DE SOFTWARE.....	70
2.10.1	LA TRIADA DE LA SEGURIDAD DE LA INFORMACIÓN	71
2.11	PRUEBAS DEL SISTEMA.....	72
3	DISEÑO METODOLOGICO	74
3.1	DIAGNÓSTICO DEL SISTEMA ACTUAL	74
3.2	ANÁLISIS DEL SISTEMA.....	74
3.2.1	INGENIERÍA DE REQUERIMIENTOS.	74
3.2.2	REQUERIMIENTOS FUNCIONALES	74
3.2.3	REQUERIMIENTO NO FUNCIONALES	76
3.2.4	MODELO DEL NEGOCIO.....	76
3.3	DISEÑO DEL SISTEMA.....	94
3.3.1	DIAGRAMA DE CLASES	94
3.3.2	MODELO RELACIONAL	95
3.3.3	CODIFICACIÓN	96
3.3.4	MODELO DE NAVEGACIÓN.....	98
3.3.5	MODELO DE PRESENTACIÓN.....	98
4	MARCO APLICATIVO	99
4.1	IMPLEMENTACIÓN DEL SISTEMA	99
4.2	SITIO WEB (FRONT END).....	99
4.3	INGRESO AL SISTEMA (BACKEND)	100
4.4	EVALUACIÓN DE CALIDAD	101
4.4.1	EVALUACIÓN DE CALIDAD DEL PORTAL WEB	101

4.5	ANÁLISIS DE COSTOS.....	107
4.5.1	COCOMO II.....	107
4.6	SEGURIDAD.....	112
4.6.1	ENcriptación	112
4.7	PRUEBAS DE SOFTWARE.....	113
4.7.1	PRUEBAS DE CAJA BLANCA	113
4.7.2	PRUEBA DE CAJA NEGRA.....	116
4.8	CONCLUSIONES Y RECOMENDACIONES.....	119
4.8.1	CONCLUSIONES.....	119
4.8.2	RECOMENDACIONES	120
	BIBLIOGRAFÍA.....	121

ÍNDICE DE TABLAS

	Pág.
Tabla 2.1: Distribución del Mercado de Software Actual y Futuro	60
Tabla 2.2.: Productividad para el modelo Composición de Aplicación.....	62
Tabla 2.3.: Ponderación para Puntos de Función.....	65
Tabla 2.4.: Factores de Complejidad	66
Tabla 2.5: Factores de Escala	68
Tabla 2.6.: Valores para los Factores de Escala	68
Tabla 2.7.: Multiplicadores de Esfuerzo	69
Tabla 2.8: Valores para Multiplicadores de Esfuerzo	70
Tabla 3.1.: Requerimientos Funcionales.....	75
Tabla 3.2.: Requerimientos no funcionales.....	76
Tabla 3.3.: Descripción de Actores de Casos de Uso del Gimnasio Winner.	77
Tabla 3.4.: Caso de Uso Ingresar al Sistema (Usuario).	78
Tabla 3.5.: Flujo de Eventos: Caso de Uso Ingresar al Sistema (Usuario),.....	78
Tabla 3.6.: Caso de Uso Registrarse (Usuario).	79
Tabla 3.7.: Flujo de Eventos: caso de Uso Registrarse por parte del Usuario. ..	79
Tabla 3.8.: Caso de Uso Ingresar al Sistema (Administrador).	80
Tabla 3.9.: Flujo de Eventos: caso de Uso Ingresar al Sistema (Administrador).	80
Tabla 3.10.: Caso de Uso Administrar Instructores.....	81
Tabla 3.11.: Flujo de Eventos: Administrar Instructores.....	82
Tabla 3.12.: Caso de Uso Ingresar al Sistema (Instructor).....	83
Tabla 3.13.: Flujo de Eventos: Caso de Uso Ingresar al Sistema (Instructor). ...	83
Tabla 3.14.: Caso de Uso Cerrar Sesión (Usuario, Administrador e Instructor). 84	84

Tabla 3.15.: Flujo de Eventos: Caso de Uso Cerrar Sesión (Usuario, Administrador e Instructor).....	84
Tabla 4.1.: Factor Operación del Producto.	101
Tabla 4.2.: Factor de Revisión del Producto.	103
Tabla 4.3.: Factor de Transición del Producto.	104
Tabla 4.4.: Puntos de Función del Sistema.	107
Tabla 4.5.: Factor de Complejidad.	108
Tabla 4.6.: Factores de Escala.	110
Tabla 4.7.: Multiplicadores de Esfuerzo.	111
Tabla 4.8.: Escala de Salarios.	112
Tabla 4.10.: Valores límites – inicio de sesión.....	116
Tabla 4.11.: Prueba de Caja Negra – Inicio de Sesión.....	117
Tabla 4.12.: Valores Límites – Reserva.....	118
Tabla 4.13.: Prueba de Caja Negra – Reserva.....	119

ÍNDICE DE FIGURAS

	Pág.
Figura 1.1.: Organigrama Gimnasio WINNER	2
Figura 2.1.: Grafica General de Sistema.....	18
Figura 2.2.: Actividades de un Sistema de Información.....	19
Figura 2.3.: Vista General De Modelos UWE.....	31
Figura 2.4.: Estereotipos de Casos de Uso.....	31
Figura 2.5.: Modelo de Caso de Uso.....	32
Figura 2.6.: Estereotipo de Diagrama de Contenido.....	33
Figura 2.7.: Diagrama de Contenido.....	33
Figura 2.8.: Diagrama de Navegación UML.....	34
Figura 2.9.: Nombre y Símbolo de Estereotipos – Modelo de Navegación.....	35
Figura 2.10.: Nombre y Símbolo de Estereotipos – Modelo de Presentación.	36
Figura 2.11.: Diseño de Presentación UWE.	36
Figura 2.12.: Capa de la ingeniería de software.....	45
Figura 2.13.: Descripción de uso PHP.....	48
Figura 2.14.: Estructura básica de HTML.	49
Figura 2.15.: Composición MVC.	56
Figura 2.16.: Modelo de calidad de Software McCALL.....	58
Figura 2.17.: Triada de la Información.....	71
Figura 3.1.: Diagrama de Casos de Uso del Gimnasio Winner.....	76
Figura 3.2.: Caso de Uso, Ingresar al Sistema (Usuario).	77
Figura 3.3.: Caso de Uso Registrarse (Usuario).	78
Figura 3.4.: Caso de Uso Ingresar al Sistema (Administrador).	79
Figura 3.5.: Caso de Uso Administrar instructores.	81
Figura 3.6.: Caso de Uso Ingresar al Sistema (Instructor).	82
Figura 3.7.: Caso de Uso Cerrar Sesión (Usuario, Administrador e Instructor).	83
Figura 3.8.: Diagrama de Secuencia al Ingresar al Sistema (Usuario).	84
Figura 3.9.: Diagrama de Secuencia Registrarse.....	85
Figura 3.10.: Diagrama de Secuencia Ingresar al sistema (Administrador).....	85
Figura 3.11.: Diagrama de Frecuencia Administrar Instructores.....	86
Figura 3.12.: Diagrama de Secuencia Ingresar al Sistema (Instructor).	87

Figura 3.13.: Diagrama de Secuencia Cerrar Sesión (Usuario, Administrador e Instructor).....	87
Figura 3.14.: Diagrama de Colaboración Ingresar al Sistema (Usuario).	88
Figura 3.15.: Diagrama de Colaboración Registrarse (Usuario).	88
Figura 3.16.: Diagrama de Colaboración Ingresar al sistema (Administrador)...	89
Figura 3.17.: Diagrama de Colaboración Administrar Instructores.....	89
Figura 3.18.: Diagrama de Colaboración Ingresar al Sistema (Instructor).....	90
Figura 3.19.: Diagrama de colaboración Finalizar Sesión.....	90
Figura 3.20.: Diagrama de Estado Ingresar al Sistema (Usuario).....	91
Figura 3.21.: Diagrama de Estado Registrarse (Usuario).....	91
Figura 3.22.: Diagrama de Estado Ingresar al Sistemas (Administrador).....	92
Figura 3.23.: Diagrama de Estado Administrar instructores.....	92
Figura 3.24.: Diagrama de Estado Ingresar al Sistema (Instructor).....	93
Figura 3.25.: Diagrama de Estado Cerrar Sesión (Usuario, Administrador e Instructor).....	93
Figura 3.26.: Modelo de Presentación.....	94
Figura 3.27.: Modelo de Presentación.....	95
Figura 3.28.: Codificación Login.....	96
Figura 3.29.: Control de Acceso Usuario.	97
Figura 3.30.: Modelo de Navegación.	98
Figura 3.31.: Modelo de Presentación.....	98
Figura 4.1.: Sitio web.	99
Figura 4.2.: Ingreso al Sistema.	100
Figura 4.3.: Sistema Administrativo (Back end).	101
Figura 4.4.: Seguridad de Contraseña.....	113
Figura 4.5.: Representación Prueba de Caja Blanca.	113
Figura 4.6.: Prueba de Caja Blanca – Inicio de Sesión.....	114
Figura 4.7.: Prueba de aja Negra – inicio sesión.	116
Figura 4.8.: Prueba de Caja Negra – Reserva.	118

DEDICATORIA

Mi Proyecto de Grado la dedico primeramente a Dios por haberme guardado y cuidado durante todo el proceso de mi carrera.

A mi Madre Lucia Apaza vda. De Yahuasi y Como también a mi padre que en paz descanse a Luis Felipe Yahuasi por haberme enseñado el valor del sacrificio y por su apoyo incondicional en cada etapa de mi vida.

A mis hermanos Jesus Felix Yahuasi, Edwin Yahuasi, Elizabeth Yahuasi y Vannesa Yahuasi porque siempre han apoyado para poder lograr y cumplir mis sueños y metas.

AGRADECIMIENTO

Agradezco a Dios por hacer realidad mi sueño y cumplir una de mis metas.

A la Universidad Pública de El Alto UPEA por haberme aceptado ser parte de ella y abierto las puertas en su seno para poder estudiar mi Carrera Ingeniería de Sistemas, así como también a los diferentes docentes que brindaron sus conocimientos y apoyo para seguir adelante día a día.

A mi madre, hermanos y hermanas que me acompañaron a lo largo de este camino, brindándome apoyo y ánimo para salir adelante.

*Gracias, de corazón, a mis tutores, **Ing. Dionicio Henry Pacheco Ríos, Lic. Freddy Salgueiro Trujillo y M. Sc. Juan Marcos Miranda Nina.** Gracias por su paciencia, dedicación, motivación, criterio y aliento. Han hecho fácil lo difícil, ha sido un privilegio poder contar con su guía y ayuda.*

*Gracias a las personas que, de alguna manera u otra, han sido un apoyo para el avance de mi proyecto, gracias a mis amigos, **Ing. Edwin Eduardo Silisqui Aramayo, Noemy Mamani Chuyma y Jose Luis Rondo Mamani,** muchas gracias por todo el apoyo y por estar en el momento más necesitado.*

RESUMEN

En este proyecto se desarrolló el Sistema de información Web de Control Administrativo. Todo el sistema se desarrolla bajo el sistema de Software Libre utilizando como entorno de desarrollo Phalcon y el lenguaje de programación de Php para el Sistema Web.

El objetivo principal fue desarrollar un Sistema Web para la Administración del Gimnasio Winner, para poder reducir el tiempo de registro e información, así mismo poder brindar una mejor información de todos los servicios, productos, rutinas que ofrece el gimnasio y dar una buena información a los clientes inscritos para poder brindar una buena atención al cliente. También se habilitará un formulario de reservas para que el cliente pueda realizar toda su duda que tenga sobre los servicios que ofrece el Gimnasio y poder mandar sus consultas.

Empleamos la metodología UWE (UML-Based Web Engineering), porque es una herramienta para modelar aplicaciones web y es utilizada en la ingeniería web. Además, UWE, es especializada en la especificación de aplicaciones adaptativas que hace hincapié en las características de personalización. UWE, se basa en el proceso unificado (UML) pero adaptados al desarrollo web.

1 MARCO PRELIMINAR

1.1 INTRODUCCION

Tras la creación del Internet y su estandarización, hoy en día las páginas web son herramientas muy importantes que nos ayudan a poder interactuar con los usuarios o clientes, ver las necesidades de la sociedad y de esa manera poder ofrecer servicios. Al mismo tiempo las plataformas o aplicaciones web de los gimnasios sean convertido en una de las opciones más aceptables en la red, por ello se plantea el proyecto sistema de información de control administrativo de esa manera podrán acceder a la información del gimnasio WINNER.

El proyecto propuesto realizará la adecuada administración de todos la información y control del gimnasio WINNER, dado que llega un punto donde se dificulta el manejo de la información, si es necesario hacer una adecuada modificación. Así para poder manejar esta información de una manera más fácil y simplificada, se realizará la creación del sistema que controle toda la información.

El presente proyecto propuesto tiene como fin reducir el tiempo en los procesos administrativos dentro del gimnasio WINNER con la implementación de un SISTEMA DE INFORMACION WEB DE COTROL ADMINISTRATIVO que proporciona mayor información, actualizada, mediante una herramienta útil para la asistencia a los procesos administrativos, consultas, nutrición, publicidad y otros.

El gimnasio WINNER no cuenta con un sistema de información web es por tal motivo que se ve la necesidad de implementar un sistema que facilite una buena administración y haga de esta entidad un buen gimnasio competitivo y eficiente.

Se plantea usar la metodología UWE (UML-BASED WEB ENGINEERING) ya que cumple y cubre todo el ciclo de vida de las plataformas y aplicaciones web. Las herramientas a emplear son: (Framework, Phalcon, lenguaje de programación PHP, gestor de base de datos).

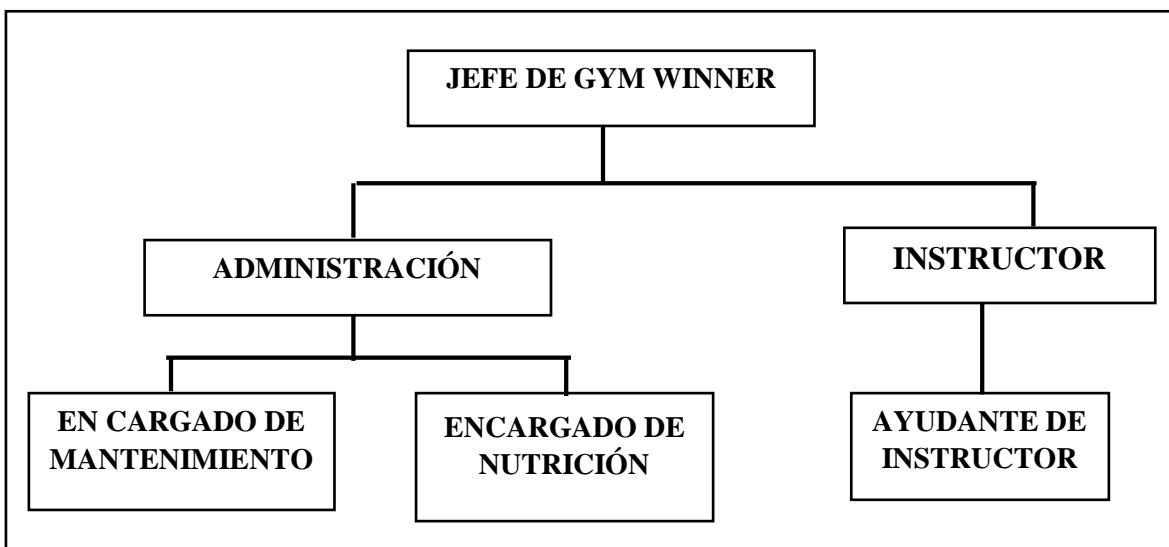
1.2 ANTECEDENTES

1.2.1 ANTECEDENTES INSTITUCIONALES

El gimnasio fitness WINNER fue fundada el 10 de septiembre de 2004, se encuentra ubicada, en la ciudad de La Paz - El Alto Zona. Villa Tunari, AV. Juan pablo II, Edificio San Juan de Dios 4to piso, además cuenta con N° NIT 6139906018.

El objetivo del proyecto a realizar del Gimnasio WINNER es brindar un buen servicio donde toda la persona esté satisfecha con el Gimnasio y así poder llegar hacer un Gimnasio reconocido y de calidad.

Figura 1.1.: Organigrama Gimnasio WINNER.



Fuente: Gimnasio Winner.

- **Misión**

Brindar a nuestros miembros una buena salud física y mental, para ayudarles a alcanzar sus objetivos individuales; con nuestra amplia experiencia les proveemos bienestar en base a un esmerado servicio como un ambiente agradable y con un personal entrenado en los últimos conocimientos disponibles.

- **Visión**

Para el 2021 abrir un nuevo ambiente y ofrecer nuestros servicios a las personas y así poder ser el gimnasio líder de la ciudad, brindando bienestar a nuestros miembros, generando valor a nuestro gimnasio, a nuestros colaboradores y a nuestra comunidad.

1.2.2 ANTECEDENTES INTERNACIONALES

Se desarrollan diversos Proyectos de Grado, existen algunos proyectos que se relacionan de alguna manera con el actual proyecto de grado tales son:

- **Título: “DESARROLLO DE UNA APLICACIÓN WEB PARA UN GIMNASIO”.**

Autor: Santiago Martínez Pera

Año: 2014

Diseñar, desarrollar una aplicación web para un gimnasio. La aplicación web tendrá un sistema de autenticación de usuarios, distinguiendo entre el administrador, usuarios registrados, Universidad de Navarra (España).

Herramientas

Este proyecto es el completo desarrollo de una aplicación web en HTML5, con la ayuda de la librería JavaScript JQuery Mobile, y con el sistema de gestión de datos desarrollado a partir de PHP y MySql.

- **Título: “Gestión de Horarios y Accesos de Un Gimnasio”**

Autor: Héctor Martínez García

Año: 2011

El presente proyecto se desarrolló un software en ambiente web para el fácil manejo y administración de horarios. Escuela técnica superior de ingeniería informática. Universidad de Málaga (España).

Herramienta

Para desarrollar usar el lenguaje ASP o SP.NET, aunque también puede extenderse mediante extensiones para que sea capaz de procesar ficheros PHP o Perl.

1.2.3 ANTECEDENTES NACIONALES

- **Título: “Complejo de fitness Integral Formación y Salud Física”**

Autor: Álvaro Cruz López

Año: 2017

El proyecto un COMPLEJO DE FITNESS INTEGRAL, el cual se diseñó con el fin de reducir el crecimiento en la población de la ciudad del El Alto, trayendo beneficios de salud, evitando enfermedades por medio de la formación y educación física.

- **Título: “CONTROL E INFORMACION NUTRICIONAL BAJO TECNOLOGIAS ANDROID”**

Autor: Alexandra Ivette López Chuquimia

Año: 2014

Impartir de forma amplia orientaciones nutricionales que contribuyan a fomentar hábitos saludables, a través del control e información nutricional mediante una aplicación móvil basada en Android.

Metodología XP

Metodología ágil para pequeños y medianos equipos. Desarrollando software cuando los requerimientos son ambiguos o rápidamente cambiantes. XP asume el cambio como algo natural ya que en alguna etapa del proyecto sucede.

Herramienta

La aplicación es desarrollada en Plataforma Android, Entorno de desarrollo Eclipse y Lenguaje de Programación Java.

1.3 PLANTEAMIENTO DEL PROBLEMA

Actualmente el gimnasio WINNER no cuenta con ningún tipo de control, ya que la información se lleva de forma manual mediante planilla por este hecho se presenta problemas en el registro de la administración lo cual causa pérdidas de información en cuanto a los registros entregados diariamente, todos estos factores provocan que existan perdidas de dinero obteniendo durante el día de trabajo sin tener un software que controle la información.

Por esta razón el gimnasio actualmente se vio en la necesidad de obtener un sistema web para almacenar la toda la información necesaria, este tipo de sistema se adapta a su forma de trabajo logrando reducir el tiempo en el manejo de la información y así poder llevar una mejor administración de todos los inscritos que realiza el gimnasio.

1.3.1 PROBLEMA PRINCIPAL

El actual problema del gimnasio WINNER se centra en el manejo de la información ya que lo realiza de forma manual sin ningún orden específico, lo cual genera pérdida de tiempo al momento de registrar y administrar a los clientes que se inscriben dentro del gimnasio y la ausencia de un Sistema de registro e información es muy necesaria.

1.3.2 PROBLEMAS SECUNDARIOS

En el análisis de información que se maneja y genera en la administración del Gimnasio WINNER, se lograron identificar los siguientes problemas:

- El control y registro de la administración es de forma manual, y se emplea demasiado tiempo en ello.
- La actualización periódica de control es manual, lo cual provoca la entrega de información a destiempo.

- La elaboración de información no se lo puede proporcionar en el tiempo establecido, causa demora al realizar los informes.
- La falta de control de la administración de información, lo que dificulta una adecuada toma de decisiones respecto a las distintas operaciones que realiza el gimnasio.
- Carencia de un sistema de control para el almacenamiento de información.
- Los datos de registro no se obtienen de manera rápida y oportuna al no estar centralizados.

Por los problemas citados surge la siguiente interrogante:

¿De qué manera se puede solucionar el correcto control de registro por la administración del gimnasio WINNER?

1.4 OBJETIVOS

1.4.1 OBJETIVO GENERAL

Desarrollar un sistema de información web de control administrativo y genere la información del gimnasio que permita organizar y reducir el tiempo en el control de manera confiable y eficaz para su mejor rendimiento en la administración.

1.4.2 OBJETIVO ESPECIFICO

- Diseñar una base de datos adecuado y confiable según los requerimientos para el almacenamiento de la información que maneja el gimnasio WINNER.
- Implementar un módulo para la administración de la información del gimnasio.
- Reducir tiempo en la inscripción de clientes por parte de la administración del Gimnasio.
- Realizar un óptimo control de información de inscritos por parte de la administración del gimnasio.
- Elaborar un reporte y/o reportes detallados de la información de clientes inscritos en el gimnasio.
- Brindar una información optima, rápida y confiable para la toma de decisiones.
- Publicar en un sitio web de información que el gimnasio tiene sobre el servicio y productos que ofrece.

1.5 JUSTIFICACION

1.5.1 JUSTIFICACIÓN TÉCNICA

Para el desarrollo del presente proyecto se cuentan actualmente con el equipo y las herramientas adecuadas para la implementación del sistema propuesto, también existe los medios necesarios por el gimnasio WINNER para realizar el análisis, diseño e implementación y mantenimiento del sistema para su funcionamiento en internet.

1.5.2 JUSTIFICACIÓN ECONÓMICA

El desarrollo del sistema proporciona un buen control de registros e información a largo plazo disminuyendo el tiempo, permitiendo a la administración del gimnasio una información rápida y confiable en tiempo real. Lo que implica un ahorro económico al ingreso de datos e información, el desarrollo del sistema web de registro e información emplea herramientas que sustituyan los procesos manuales que generan un costo innecesario para el gimnasio.

1.5.3 JUSTIFICACIÓN SOCIAL

Con la implementación del sistema web de registro beneficia al personal encargado de la administración, permitiendo reducir el tiempo y los procesos manuales, a si también permitir que el registro sea rápida y eficaz, además los inscritos y personal perteneciente al gimnasio podrán contar con una información rápida, confiable, oportuna y segura de todo el flujo de información de vital importancia para un amplio campo de actividades referente al crecimiento de inscritos y correspondientes proyecciones que se realiza para la toma de futuras decisiones en beneficio del gimnasio.

En cuanto al Sistema web de registros e información será utilizado como una herramienta que brindará apoyo en las actividades del personal de administración del gimnasio.

1.6 METODOLOGIA

1.6.1 METODOLOGIA UWE

Es un proceso del desarrollo para aplicaciones Web enfocados sobre el diseño sistemático, la personalización y la generación semi-automática de escenarios que guie el proceso de desarrollo de una aplicación web. UWE describe una metodología

de diseño sistemática, basada en la técnica UML, la notación de UML y los mecanismos de extensión de UML.

Es una herramienta que permite modelar aplicaciones web, utilizada en la ingeniería web, prestando especial atención en automatización y personalización (sistemas adaptativos). UWE es una propuesta basada en el proceso unificado UML, pero adaptados al web. En requisitos separa las fases de captura, definición y validación. Hace además una clasificación y un tratamiento especial dependiendo del carácter de cada requisito.

El marco de UWE es necesario la definición de un perfil UML (extensión) basado en estereotipos con este perfil se logra la asociación de una semántica distinta a los diagramas del UML, pero con el propósito de acoplar el UML a un dominio específico, en este caso, las aplicaciones Web. Entre los principales modelos de UWE podemos citar: el modelo lógico-conceptúa, modelo navegación, modelo de presentación, visualización de escenario Web y la interacción temporal, entre los diagramas: diagrama de estado, secuencia, colaboración y actividad.^º

UWE define vistas especiales representadas gráficamente por diagramas en UML. Además, UWE no limita el número de vistas posibles de una aplicación, UML proporciona mecanismo de extensión basada en estereotipos. Estos mecanismos de extensión son los que UWE utiliza para definir estereotipos que son lo que finalmente se utiliza en las vistas especiales para el modelado de aplicaciones Web. De esta manera, se obtiene una notación UML adecuado a un dominio en específico a la cual se le conoce perfil UML.

Además de estar considerado como una extensión de estándar UML también se basa en otros estándares como, por ejemplo: XMI como modelo de intercambio de formato, MOF para la meta-modelado, los principios modelados de MDA el modelo de transformación del lenguaje QVT, XML.

Fases o etapas de la metodología UWE

- **1º Fase: Captura, análisis y especificación de requisitos**

En simple palabras y básicamente, durante esta fase, se adquieren, reúnen y especifican las características funcionales y no funcionales que deberá cumplir la aplicación web.

- **2° Fase: Diseño del sistema**

Se basa en la especificación de requisitos producido por el análisis de los requerimientos (fase de análisis), el diseño define cómo estos requisitos se cumplirán, la estructura que debe darse a la aplicación web.

- **3° Fase: Codificación del software**

Durante esta etapa se realizan las tareas que se conocen como programación; que consiste, esencialmente, en llevar a código fuente, en el lenguaje de programación elegido, todo lo diseñado en la fase anterior.

- **4° Fase: Pruebas**

Las pruebas se utilizan para asegurar el correcto funcionamiento de secciones de código.

- **5° Fase: La Instalación o Fase de Implementación**

Es el proceso por el cual los programas desarrollados son transferidos apropiadamente al computador destino, inicializados, y, eventualmente, configurados; todo ello con el propósito de ser ya utilizados por el usuario final.

Implementación y Lanzamiento: En la implementación de la Pagina Web es recomendable utilizar estándares (HTML, XHTML...) para asegurar la futura compatibilidad y escalabilidad del sitio. Una vez implementada la página web y aprobada su funcionalidad se procede al lanzamiento del sitio.

- **6° Fase: El Mantenimiento**

Es el proceso de control, mejora y optimización del software ya desarrollado e instalado, que también incluye depuración de errores y defectos que puedan haberse filtrado de la fase de pruebas de control.

1.7 HERRAMIENTAS

Para el desarrollo del Sistema de información web que necesita el gimnasio WINNER los recursos a utilizar en cuanto a software, PHP como lenguaje de programación, con framework, Jquery, Phalcon, MySQL para la base de datos y un servidor Apache HTTP Server.

1.7.1 LENGUAJE DE PROGRAMACIÓN PHP

Es un lenguaje de programación de propósito general de código del lado del servidor originalmente diseñado para el desarrollo web de contenidos dinámicos. Fue uno de

los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en un documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genere el HTML resultante.

1.7.1.1 Características

- Utilizado para generar páginas web dinámicas.
- Se ejecuta en el servidor.
- Los usuarios no pueden ver el código PHP únicamente reciben en sus navegadores código HTML.
- Las páginas que genera son visibles para prácticamente cualquier navegador y computadora o dispositivos móviles que pueda interpretar el HTML.
- No se necesita la instalación de PHP en el lado del cliente.
- Versiones resiente permiten la POO.
- Lenguaje de alto nivel.

1.7.1.2 Ventajas

- Su sintaxis es muy similar a otros lenguajes.
- Fácil.
- Es un lenguaje muy popular tiene una comunidad muy grande.
- Rápido.
- Multiplataforma.
- Maneja base de datos.
- Bastante documentado.
- Libre y gratuito.
- Varias funciones.
- No requiere definición de variables.
- Puede ser combinado junto a HTML.
- Tiene muchos frameworks que facilitan el desarrollo en este lenguaje.
- Muchos servicios de alojamiento web tienen PHP

1.7.1.3 Desventajas

- Necesita un servidor para funcionar.

- La POO es deficiente para aplicaciones grandes.
- Todo el trabajo se realiza en el servidor y mucha información o solicitudes pueden ser ineficiente.

1.7.2 HTML

Tras desterrar al flash, el HTML5 se ha convertido en el rey de los lenguajes web por ser semántico, adaptable, flexible, escalable y multiplataforma. En Go Móvil, y dadas sus múltiples ventajas, no dudamos en utilizarlo para el desarrollo de soluciones móviles destinadas a grandes y pequeñas empresas.

1.7.2.1 Ventajas

- Es nativo, y por tanto independiente de plugins de terceros. Es decir, no pertenece a nadie, es opensource.
- Es más semántico, con etiquetas que permiten clasificar y ordenar en distintos niveles y estructuras el contenido. Además, incorpora metadatos de manera más formal, favoreciendo el posicionamiento SEO y la accesibilidad.
- El código es más simple lo que permite hacer páginas más ligeras que se cargan más rápidamente favoreciendo la usabilidad y la indexación en buscadores.
- Ofrece una compatibilidad mayor con los navegadores de dispositivos móviles.
- Incluye la etiqueta de dibujo canvas, que ofrece más efectos visuales.
- Ofrece soporte a codecs específicos.
- Posibilita la inserción de vídeos y audio de forma directa.
- Permite la geolocalización del usuario. Algo muy útil para el marketing móvil.
- Tiene la capacidad de ejecutar páginas sin estar conectado.
- Dispone de nuevas capacidades CSS3 como posibilidad de usar cualquier fuente o tipografía en HTML, columnas de texto, opacidad, transparencia, canales alpha, contraste, saturación, brillo, animaciones de transición y transformación, bordes
- redondeados, gradientes, sombras, etc.
- Permite realizar diseños adaptables a distintos dispositivos (web, tablets, móviles).

1.7.2.2 Desventajas

- Formato de vídeo estándar: por el momento no hay un formato de vídeo compatible con todos los navegadores web y cuando a una plataforma se suben 24 horas de vídeo cada minuto es importante minimizar el número de formatos a soportar.
- Control de streaming: el visor de vídeo en Flash permite acceder a partes en concreto del mismo a través de la manipulación de variables que Flash toma e interpreta via Actionscript y que HTML5 no tiene. La etiqueta video de HTML5 muestra un archivo como quien enlaza una foto sin posibilidad de manejar el concepto streaming.
- HTML5 aún no cuenta con todas las funcionalidades necesarias para poder incrustar un vídeo en la red con todos los elementos que tiene YouTube actualmente: anotaciones, subtítulos, anuncios, relacionados... Además, tampoco permite reproducir vídeos HTML5 a pantalla completa.
- Acceso a cámara y micrófono: La etiqueta video de HTML5 sirve para ver videos, no para interactuar, mientras que Flash lleva una importante ventaja tecnológica al permitir grabar y hacer videoconferencias desde el browser.

1.7.3 FRAMEWORK BootStrap

Bootstrap, es un framework originalmente creado por Twitter, que permite crear interfaces web con CSS y JavaScript, cuya particularidad es la de adaptar la interfaz del sitio web al tamaño del dispositivo en que se visualice. Es decir, el sitio web se adapta automáticamente al tamaño de una PC, una Tablet u otro dispositivo. Esta técnica de diseño y desarrollo se conoce como Responsive Design o Diseño Adaptativo.

1.7.3.1 Ventajas

- Utiliza componentes y servicios creados por la comunidad web, tales como: HTML5 shim, Normalize.css, OOCSS (CSS orientado a objetos), jQuery UI, LESS y GitHub.
- Es un conjunto de buenas prácticas.

- El famoso Grid system, que por defecto incluye 12 columnas fijas o fluidas, dependiendo de si tu diseño. Está orientado siguiente el concepto Mobile First.
- **Soporte:** Hay una enorme comunidad que soporta este desarrollo y cuenta con implementaciones externas como WordPress, Drupal, SASS o jQuery UI.
- **Comodidad y rapidez:** Herramienta sencilla y ágil para construir sitios web e interfaces.
- **Componentes:** Mucha variedad de plantillas y temas.

1.7.3.2 Desventajas

- **Aprendizaje:** Es necesario adaptarse a su forma de trabajo, si bien su curva de aprendizaje es liviana, deberás comprender y familiarizarte con su estructura y nomenclatura.
- **Adaptación:** Debes adaptar tu diseño a un grid de 12 columnas, que se modifican según el dispositivo. Aquí empiezan los problemas, Bootstrap por defecto te trae anchos, márgenes y altos de línea, y realizar cambios específicos es por decir, un poco tedioso.
- **Mantenimiento:** Es complicado, cambiar de versión si has realizado modificaciones profundas sobre el core.
- **Ampliar componentes:** Si necesitas añadir componentes que no existen, debes hacerlos tú mismo en CSS y cuidar de que mantenga coherencia con tu diseño y cuidando el responsive.
- **Pesado:** No es ligero y además, para algunas funcionalidades, será necesario tener que usar JavaScript y jQuery.

1.7.4 FRAMEWORK PHALCON

Hay muchos frameworks para PHP hoy en día, pero ninguno como Phalcon. Casi todos los programadores preferimos usar un framework. Esto debido a que nos proporcionan una gran funcionalidad que está probada y lista para usar, al mismo tiempo no repitiéndonos y reusando código. Sin embargo, los frameworks requieren incluir muchos archivos e interpretar miles de líneas de código en cada petición. Adicionalmente frameworks orientados a objetos agregan una cantidad considerable

de overhead. Todas estas operaciones hacen que las aplicaciones sean más lentas por consiguiente impactando la experiencia de usuario de manera negativa.

1.7.4.1 Características

- Los componentes están libremente acoplados. Con Phalcon, nada está impuesto: tienes la libertad de usar todo el framework, o solo las partes que necesites.
- Optimizaciones de bajo nivel ayudan a reducir la sobrecarga requerida para correr aplicaciones MVC.
- Las operaciones con base de datos se efectúan con la máxima eficiencia al usar un ORM para PHP escrito en C.
- Phalcon accede directamente a las estructuras internas de PHP optimizando además cada ejecución.
- Usa un framework PHP basado en C.
- Aprovecha el fantástico rendimiento de Phalcon y la reduce sobrecarga de recursos.
- Utiliza solo los módulos y bibliotecas que necesitas.

Phalcon es un framework full-stack de código abierto para PHP 5 escrito como extensión en C, optimizada para alto rendimiento. No necesitas aprender o usar C, toda su funcionalidad está expuesta como clases PHP listas para usar. Phalcon está también débilmente acoplado permitiéndote usar sus clases como componentes de acuerdo a como tu aplicación lo requiera.

1.7.5 GESTOR DE BASES DE DATOS

Es un sistema de gestión de bases de datos relacional, fue creada por la empresa sueca MySQL AB, la cual tiene el copyright del código fuente del servidor SQL, así como también de la marca. MySQL es un software de código abierto, licenciado bajo la GPL de la GNU, aunque MySQL AB distribuye una versión comercial, en lo único que se diferencia de la versión libre, es en el soporte técnico que se ofrece, y la posibilidad de integrar este gestor en un software propietario ya que, de otra manera, se vulneraría la licencia GPL. El lenguaje de programación que utiliza MySQL es

Structured Query Lenguaje (SQL) que fue desarrollado por IBM en 1981 y desde entonces es utilizado de forma generalizada en las bases de datos relacionales”.

1.7.5.1 Características

Inicialmente, MySQL carecía de algunos elementos esenciales en las bases de datos relacionales, tales como integridad referencial y transacciones. A pesar de esto, atrajo a los desarrolladores de páginas web con contenido dinámico, debido a su simplicidad, de tal manera que los elementos faltantes fueron complementados por la vía de las aplicaciones que la utilizan. Poco a poco estos elementos faltantes, están siendo incorporados tanto por desarrolladores internos, como por desarrolladores de software libre.

En las últimas versiones se pueden destacar las siguientes características principales:

- El principal objetivo de MySQL es velocidad y robustez.
- Soporta gran cantidad de tipos de datos para las columnas.
- Gran portabilidad entre sistemas, puede trabajar en distintos sistemas operativos.
- Cada base de datos cuenta con 3 archivos: Uno de estructura, uno de datos y uno de índice y soporta hasta 32 índices por tabla.
- Aprovecha la potencia de sistemas multiproceso, gracias a su implementación multihilo.
- Flexible sistema de contraseñas (passwords) y gestión de usuarios, con un muy buen nivel de seguridad en los datos.
- El servidor soporta mensajes de error en distintas lenguas.

1.7.5.2 Ventajas

- Velocidad al realizar las operaciones, lo que le hace uno de los gestores con mejor rendimiento.
- Bajo costo en requerimientos para la elaboración de bases de datos, ya que debido a su bajo consumo puede ser ejecutado en una máquina con escasos recursos sin ningún problema.
- Facilidad de configuración e instalación.
- Soporta gran variedad de Sistemas Operativos.

- Baja probabilidad de corromper datos, incluso si los errores no se producen en el propio gestor, sino en el sistema en el que está.
- Conectividad y seguridad.

1.7.5.3 Desventajas

- Un gran porcentaje de las utilidades de MySQL no están documentadas.
- No es intuitivo, como otros programas (ACCESS).

1.8 LIMITES Y ALCANCES

1.8.1 LIMITES

El presente proyecto se limita a optimizar en control de registros por la administración, consultas, información de acuerdo a la conformidad y necesidad de la administración.

- El proyecto no tendrá en cuenta en la parte contable o financiera.
- El proyecto no realizara un control de la ejecución presupuestaria.

1.8.2 ALCANCES

El sistema de información abarca a realizar un manejo eficiente y eficaz de la información logrando que el sistema sea una aplicación de gran utilidad para el gimnasio WINNER. La aplicación a desarrollar abarcará los siguientes módulos:

- **Modulo Administrativo:** Tendrá acceso a todo el sistema sin restricciones.
 - Control y administración de usuario.
 - Búsqueda, que coadyuve al control y administración de la información.
- **Modulo Configuración:** Este módulo está preparado para presentar los formularios que permiten los ajustes del sistema a ciertas necesidades del gimnasio “WINNER”. Accede a la personalización de datos y archivos indispensables del gimnasio que administra el sistema.
- **Módulo Cliente:** tendrá acceso limitado al sistema.
Podrá ver todas las disciplinas disponibles que ofrece el gimnasio “WINNER”, realizar reservas y consultas en línea.

Verificación y optimización de la información de los clientes.

Dar aviso oportunamente a la terminación del mes a los clientes que se encuentre inscrito en el gimnasio.

Dar una informe rápida y eficaz de los registros.

Emitir informes de todos los clientes registrados en el gimnasio.

El sistema de información utilizará métodos y herramientas de desarrollo.

1.9 APORTES

El desarrollo de este proyecto será el aporte al Gimnasio Winner por el hecho que habrá ciertas mejoras que permitirá administrar de manera eficiente la información de los registros para la optimizar sus procesos rutinarios disminuyendo el tiempo de búsqueda, generando información que ayude a facilitar el control administrativo del gimnasio, con ello también mejorarán sus ingresos económicos, también un aporte a la sociedad debido a que se beneficiará con un mejor servicio.

Se logrará un mejor control de la información de registros de acuerdo a los requerimientos del gimnasio, mejorando así de manera eficiente y confiable el manejo y administración de información.

Este aspecto mencionado anteriormente ayudará al gimnasio a mejorar de modificación de información, además de optimizar las tareas operativas de la gerencia, también será una herramienta confiable en la toma de decisiones.

2 MARCO TEÓRICO

2.1 INTRODUCCIÓN

En el presente capítulo se elabora toda las definiciones, términos y conceptos, así como también las herramientas a utilizar dentro del desarrollo del proyecto para conocer y aprender en cada una de ellas.

En este caso todos los términos y las herramientas de desarrollo de software son definidos con una descripción detallada de cada uno de los elementos de la teoría que serán directamente utilizados en el desarrollo del proyecto de grado.

2.2 DEFINICIÓN DE TÉRMINOS WEB

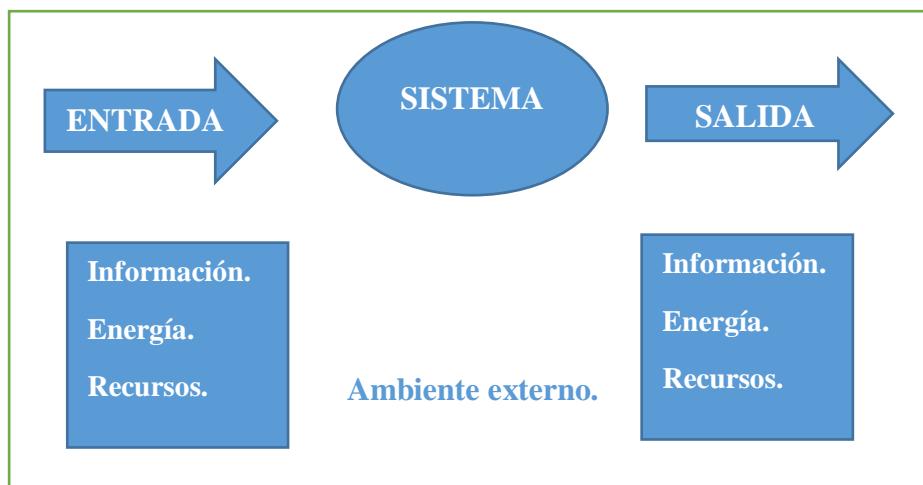
2.2.1 SISTEMA

Sistema es un conjunto de elementos relacionados entre sí y que interactúan de modo que forman una unidad lógica. En informática la palabra sistema se utiliza precisamente para indicar varios componentes, tanto de hardware como de software que en su conjunto constituye un sistema de procesamiento de datos (Pressman, 2002)

Los sistemas también pueden servir como guía de trabajo más práctico y simplificado. Dependerá de la manera en que necesites emplear un sistema y sus diferentes fases, para la elaboración de un trabajo que conlleve diferentes metas, pero con un mismo propósito. Es posible que la meta sea apoyar una función de negocio o desarrollar un producto que pueda venderse para generar beneficios.

Un sistema es un conjunto ordenado de elementos que están interrelacionados y que interactúan entre sí para lograr un objetivo. Para poder analizar un sistema se hace necesario conocer sus componentes y las relaciones existentes entre esos componentes. En conclusión, sistema es conjunto de elementos que ordenadamente relacionadas entre sí, contribuyen a un determinado objetivo.

Figura 2.1.: Grafica General de Sistema.



Fuente: Elaboración Propia.

2.2.2 SISTEMA DE INFORMACIÓN

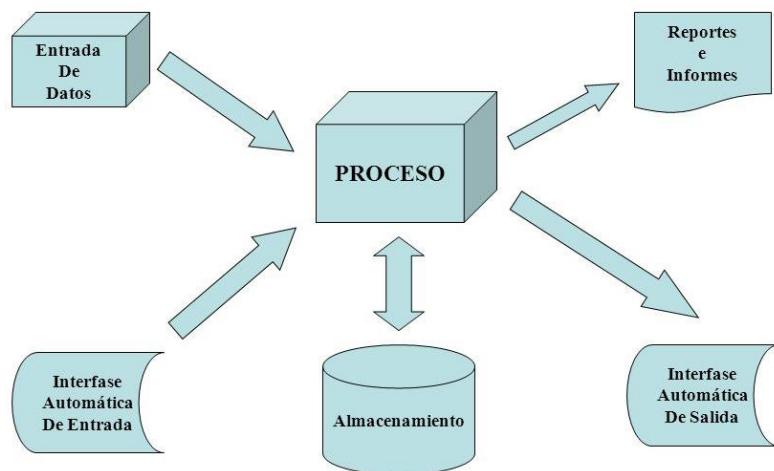
“Es un conjunto de componentes interrelacionados que recolectan, procesan, almacenan y distribuyen información para apoyar los procesos de toma de decisiones y de control de la organización” (Laudon y Laudon, 2012, pag. 15).

Además de apoyar en la toma de decisiones, la coordinación y el control, los sistemas de información son indispensables para ayudar a los gerentes a mantener ordenada su compañía, a analizar todo lo que por ella. Existen elementos de naturaleza diversa y están incluyen:

- El equipo computacional: es decir el hardware necesario para que el sistema de información pueda operar. Lo constituyen las computadoras y el equipo periférico que se conectan a ellas.
- El recurso humano: que interactúa con el sistema de información, el cual está formado por las personas que utilizan el sistema, alimentándolo con datos o utilizando los resultados que genere.
- Los datos o información fuente: que son introducidos en el sistema, son todas las entradas que este necesita para generar como resultados la información que se deseé.

- Los programas: que son ejecutados por las computadoras y producen diferentes tipos de resultados. Los programas son parte del software del sistema de información que hará que los datos de entrada introducidos sean procesados correctamente y generen los resultados que se esperan.

Figura 2.2.: Actividades de un Sistema de Información.



Fuente: Cohen & Asín, 2000.

En la Figura 2.2 se puede observar las diferentes actividades que realiza un sistema de información, que se basa en la definición del mismo.

2.2.2.1 Características de Sistema de Información

“Las características a tener en cuenta en el diseño de todo sistema de información afectaran a su relación valor-coste, es decir una mejora en cualquiera de sus características proporcionara un aumento en el valor total del sistema” (Pechuan, 1996). En lista un conjunto de características propuestas por James Emery:

- Fiabilidad: probabilidad de que el sistema opere correctamente durante un periodo de disponibilidad de uso.
- El tiempo de respuesta del sistema: diferencia entre una petición de servicio y su realización.

- Seguridad: protección contra pérdida o uso no autorizado de los recursos del sistema (accidentales o no).
- Amigabilidad para con el usuario: grado con que el sistema reduce las necesidades de aprendizaje para su manejo

2.2.2.2 Componentes de un Sistema de Información.

Los componentes que forman un sistema de comunicación son:

- **La entrada:** por donde se alimentan los datos,
- **El proceso:** uso de las herramientas de las áreas contempladas para relacionar, resumir o concluir.
- **La salida:** refleja la producción de la información.
- **La retroalimentación:** los resultados obtenidos son ingresados y procesados nuevamente.

2.2.3 SISTEMA WEB

Los sistemas Web o también conocido como “aplicaciones web” son aquellos que están creados e instalados no sobre sistemas operativos (Windows, Linux). Sino que se alojan en un servidor de internet o sobre una intranet (red local). Su aspecto es muy similar a páginas Web que vemos normalmente, pero en realidad los sistemas Web tienen funcionalidades muy potentes que brindan respuesta a casos particulares.

Los sistemas Web se pueden utilizar en cualquier navegador Web (Chrome, Firefox, Internet Explorer, etc.) sin importar el sistema operativo. Para utilizar las aplicaciones Web no es necesario instalarlas en cada computadora ya que los usuarios se conectan a un servidor donde se aloja el sistema.

Las aplicaciones Web trabajan con bases de datos que permiten procesar y mostrar información de forma dinámica para el usuario. Los sistemas desarrollados en plataformas Web, tienen marcadas diferencias con otros tipos de sistemas, lo que lo hacen muy beneficioso tanto para las empresas que lo utilizan, como para los usuarios que operan en el

sistema. Este tipo de diferencias se ven reflejadas en los costos, en la rapidez de obtención de la información, en la optimización de tareas por parte de los usuarios y en alcanzar una gestión estable. (Baez, 2012)

2.2.4 SISTEMA DE CONTROL

La rápida secuencia de resultados en teoría de control automático durante los últimos años dificulta la realización de un análisis imparcial de un área de conocimiento que se halla en continuo desarrollo, pese a ello, mirando retrospectivamente el progreso de la teoría de control realimentado, pueden distinguirse en algunas tendencias principales y apuntar diversos avances claves. “El carácter de ingeniería de esta disciplina provocó que sus logros estén fuertemente vinculados a los problemas prácticos, que han de ser solucionados en diferentes fases de la historia” (Giner, 2004).

En la actualidad las empresas u otro tipo de organizaciones tienen la necesidad de poder controlar su actividad y más concretamente los resultados obtenidos, para ello es necesario que exista un sistema de control de gestión que permita poder medir, valorar e informar sobre el cumplimiento y evolución de los objetivos fijados y que además permita prever la evolución futura de los mismos, esto ayudara en el cumplimiento de estos.

Un sistema de control es un conjunto de dispositivos encargados de administrar, ordenar, dirigir o regular el comportamiento de otro sistema, con el fin de reducir las probabilidades de fallo y obtener los resultados deseados. Existen dos clases comunes de sistemas de control, sistemas de lazo abierto y sistemas de lazo cerrado. En los sistemas de control de lazo abierto la salida no interviene en la acción de control; mientras que en los de lazo cerrado si se va a requerir conocer la salida para ejercer el control del sistema. Un sistema de lazo cerrado es llamado también sistema de control con realimentación.

2.2.5 ADMINISTRACIÓN

“Método por el cual un grupo de cooperación dirige sus acciones hacia metas comunes. Este método implica mediante la cuales un grupo principal de personas coordinan las actividades de otro” (Massie, 2011).

“La administración es un proceso de un integral para planear, organizar, dirigir, controlar y coordinar una actividad o relación de trabajo, la que se fundamenta en la utilización para alcanzar un fin determinado” (Chiavenato, 2004).

Se puede conceptualizar a la administración como el esfuerzo humano coordinado para obtener la optimización de todos los recursos a través del proceso administrativo a fin de lograr los objetivos constitucionales.

2.2.6 SISTEMA WEB DE ADMINISTRACIÓN

“Los sistemas web de administración son un conjunto de elementos para dirigir una organización o procesos, que está conformado por las tareas de planeación, organización, comunicación y control” (Ramírez, 2005).

Todo aquello cuya característica principal es que cumple algún tipo de organización, sin embargo, un sistema web de administración más que algo que tiene cierto grado de organización, es una estructura para la administración y mejora la continuidad de los procedimientos y procesos.

De acuerdo a esto un sistema de administración ayuda a lograr los objetivos de la organización mediante una serie de estrategias, que incluyen la optimización de procesos, el enfoque centrado en la gestión y el pensamiento disciplinado.

A nivel empresarial los sistemas de administración son necesarios ya que las empresas hoy en día se enfrentan a muchos retos bastante significativos como lo son la rentabilidad, competitividad, globalización, capacidad de adaptación, crecimiento y tecnología, entre otros.

2.2.7 SISTEMA WEB

Sistema web se basa en una arquitectura cliente-servidor. El cliente es el encargado de iniciar la comunicación por medio de un navegador o explorador que interpreta y visualiza la información suministrada por el servidor. El servidor está a la escucha de nuevos clientes realicen peticiones para suministrar las páginas solicitadas o procesar la información recibida por el cliente.

La Web ha venido evolucionando drásticamente estos últimos años, comenzando con una Web de Hipertexto que hizo posible enlazar y conectar los documentos compuestos por texto, imágenes, videos, etc.

(Santos y Aguilar, 2017) determinan que Inicialmente, la Web era interpretada por los humanos y no por las máquinas, pero con el surgimiento de la Web 2.0 se logró compartir los documentos y permitir la colaboración, aunque igualmente la interpretación seguía siendo realizada por los humanos. Lo que persiguen las páginas web son enlazar datos que luego permiten identificar, describir, conectar y relacionar los elementos de esos datos en la Web.

Los sistemas Web o también conocido como “aplicaciones web” son aquellos que están creados e instalados no sobre sistemas operativos (Windows, Linux). Sino que se alojan en un servidor de internet o sobre una intranet (red local). Su aspecto es muy similar a páginas Web que vemos normalmente, pero en realidad los sistemas Web tienen funcionalidades muy potentes que brindan respuesta a casos particulares.

Los sistemas Web se pueden utilizar en cualquier navegador Web (Chrome, Firefox, Internet Explorer, etc.) sin importar el sistema operativo. Para utilizar las aplicaciones Web no es necesario instalarlas en cada computadora ya que los usuarios se conectan a un servidor donde se aloja el sistema.

2.3 INGENIERÍA DE REQUERIMIENTO

La ingeniería de requerimientos proporciona el mecanismo apropiado para entender lo que desea el cliente, analizar las necesidades, evaluar la factibilidad, negociar una solución razonable, especificar la solución sin ambigüedades, validar la especificación y administrar los requerimientos a medida que se transforman en un sistema funcional.

(Pressman, 2010)

Incluye siete tareas diferentes: concepción, indagación, elaboración, negociación, especificación, validación y administración. Es importante notar que algunas de estas tareas ocurren en paralelo y que todas se adaptan a las necesidades del proyecto.

- **Concepción:** “se establece el entendimiento básico del problema, las personas que quieren una solución, la naturaleza de la solución que se desea, así como la eficacia de la comunicación y colaboración preliminares entre los otros participantes y el equipo de software” (Pressman, 2010).
- En esta parte, los ingenieros de software realizan preguntas “libres de contexto” (generales), para establecer un entendimiento básico del problema, determinar las personas que quieren una solución, la naturaleza de la solución, y la efectividad de las colaboraciones y comunicaciones preliminares que se generan entre el consumidor y el desarrollador.
- **Indagación:** se refiere a definir formalmente los requerimientos de la solución. Preguntar al cliente, a los usuarios y a otras personas cuáles son los objetivos para el sistema o producto, qué es lo que va a lograrse, cómo se ajusta el sistema o producto a las necesidades del negocio y, finalmente, cómo va a usarse el sistema o producto en las operaciones cotidianas (Pressman, 2010). Sin embargo, esta tarea es difícil de realizar, puesto que existe cierto número de problemas que pueden presentarse, Christel y Kang identificaron los siguientes:
 - Problemas de alcance: la frontera de los sistemas está mal definida o los clientes o usuarios finales especifican detalles técnicos innecesarios que confunden.
 - Problemas de entendimiento: los clientes o usuarios no están completamente seguros de lo que se necesita.
 - Problemas de volatilidad: los requerimientos cambian con el tiempo.
- **Elaboración:** se centra en desarrollar un modelo refinado de los requerimientos que identifique distintos aspectos de la función del software, su comportamiento e información. La elaboración está motivada por la creación y mejora de escenarios de usuario que describan cómo interactuará el usuario final (y otros actores) con el sistema. Cada escenario de usuario se enuncia con sintaxis apropiada para extraer clases de análisis, que son entidades del dominio del negocio visibles para el usuario final. Se definen los atributos de cada clase de análisis y se identifican los servicios que requiere cada una de ellas. Se

identifican las relaciones y colaboración entre clases, y se producen varios diagramas adicionales. (Pressman, 2010).

- **Negociación:** no es raro que los clientes y usuarios pidan más de lo que se logra dado lo limitado de los recursos del negocio. También es relativamente común que distintos clientes o usuarios propongan requerimientos conflictivos con el argumento de que su versión es esencial para nuestras necesidades especiales. Estos conflictos deben reconciliarse por medio de un proceso de negociación. Se pide a clientes, usuarios y otros participantes que ordenen sus requerimientos según su prioridad y que después analicen los conflictos. Con el empleo de un enfoque iterativo que da prioridad a los requerimientos, se evalúa su costo y riesgo, y se enfrentan los conflictos internos; algunos requerimientos se eliminan, se combinan o se modifican de modo que cada parte logre cierto grado de satisfacción. (Pressman, 2010).
- **Especificación:** una especificación puede ser un documento escrito, un conjunto de modelos gráficos, un modelo matemático formal, un conjunto de escenarios de uso, un prototipo o cualquier combinación de éstos. Significa diferentes cosas para diferentes personas. Algunos sugieren que para una especificación debe desarrollarse y utilizarse una “plantilla estándar”, con el argumento de que esto conduce a requerimientos presentados en forma consistente y por ello más comprensible. (Pressman, 2010) Sin embargo, en ocasiones es necesario ser flexible cuando se desarrolla una especificación. Para productos o sistemas pequeños que residan en ambientes bien entendidos, quizá todo lo que se requiera sea escenarios de uso.
- **Validación:** la calidad de los productos del trabajo que se generan como consecuencia de la ingeniería de los requerimientos se evalúa durante el paso de validación. La validación de los requerimientos analiza la especificación a fin de garantizar que todos ellos han sido enunciados sin ambigüedades; que se detectaron y corrigieron las inconsistencias, las omisiones y los errores, y que los productos del trabajo se presentan conforme a los estándares establecidos para el proceso, el proyecto y el producto. (Pressman, 2010).

- **Administración:** es el conjunto de actividades que ayudan al equipo del proyecto a identificar, controlar y dar seguimiento a los requerimientos y a sus cambios en cualquier momento del desarrollo del proyecto. Esta tarea sólo se practica para proyectos grandes que tienen cientos de requerimientos identificables. Para proyectos pequeños, esta actividad tiene considerablemente menos formalidad. (Pressman, 2010).

2.3.1 REQUISITOS FUNCIONALES Y NO FUNCIONALES

Los requerimientos funcionales son declaraciones de los servicios que prestará el sistema, en la forma en que reaccionará a entradas particulares. Cuando hablamos de las entradas, no necesariamente hablamos sólo de las entradas de los usuarios. Pueden ser interacciones con otros sistemas, respuestas automáticas, procesos predefinidos.

En algunos casos, los requisitos funcionales de los sistemas también establecen explícitamente lo que el sistema no debe hacer. Es importante recordar esto: un RF puede ser también una declaración negativa. Siempre y cuando el resultado de su comportamiento sea una respuesta funcional al usuario o a otro sistema, es correcto. Y más aún, no sólo es correcto, sino que es necesario definirlo. Y eso nos lleva al siguiente punto.

2.4 METODOLOGÍA DE DESARROLLO

Una metodología de desarrollo de software se refiere a un sistema de trabajo que es usado para estructurar, planear y controlar el proceso de desarrollo en sistemas de información. A lo largo del tiempo, una gran cantidad de métodos han sido desarrollados diferenciándose por su fortaleza y debilidad. Esta metodología consiste en "Una filosofía de desarrollo de programas de computación con el enfoque del proceso de desarrollo de software" (Guarachi, 2014). Estos son a menudo vinculados a algún tipo de organización, que además desarrolla, apoya el uso y promueve la metodología.

Actualmente las metodologías de ingeniería de software pueden considerarse como una base necesaria para la ejecución de cualquier proyecto de desarrollo de software

que se considere serio, y que necesite sustentarse en algo más que la experiencia y capacidades de sus programadores y equipo.

Cada metodología de desarrollo de software tiene más o menos su propio enfoque para el desarrollo de software. Estos son los enfoques más generales, que se desarrollan en varias metodologías específicas:

- Modelo en cascada: Framework lineal.
- Prototipado: Framework iterativo.
- Incremental: Combinación de framework lineal e iterativo.
- Espiral: Combinación de framework lineal e iterativo.
- RAD: Rapid Application Development, framework iterativo.

2.5 METODOLOGÍA UWE

Una de las metodologías web, más requeridas para desarrollo de software es sin dudas uwe, debido a la compatibilidad de asimilación en sus diferentes fases de modelación. En su página oficial encontramos que “UWE es un enfoque de ingeniería de software para el dominio web que tiene como objetivo cubrir todo el ciclo de vida del desarrollo de aplicaciones web”. El aspecto clave que distingue a UWE es la confianza en los estándares.

Esta metodología está basada en UML de ahí su sigla (UML Web Engineering, en español Ingeniería web basado en UML) nos permite especificar de mejor manera una aplicación web, para el proceso de creación de aplicaciones detalla esta, con una gran cantidad de definiciones, en el proceso de diseño, lista que debe utilizarse, procede de manera iterativa e incremental, coincidiendo con UML, incluyendo flujos de trabajo y puntos de control.

El principal objetivo del enfoque UWE es proporcionar, un lenguaje de modelado específico del dominio basado en UML; una metodología dirigida por modelos; herramientas de soporte para el diseño sistemático; y herramientas de soporte para la generación semi-automática de Aplicaciones Web. (Blanco, 2015)

Las características principales de UWE son:

- **XMI como modelo de intercambio de formatos:** Es un metalenguaje simple pero estricto, desarrollado por W3C. Desarrolla un papel fundamental en el intercambio de una gran variedad de datos. XML es un formato que permite la interpretación de datos a través de varias aplicaciones. Es una simplificación y adaptación del SGML y permite definir la gramática de lenguajes específicos. En realidad, XML es una manera de definir lenguajes para diferentes necesidades.
- **MOF para meta modelados:** El Meta-Object Facility, es un estándar de OMG (Object Management Group) para la definición de meta modelos, de tal forma que se facilite su importación e interacción en otros formatos, reflejada en un marco común de gestión de metadatos, independientemente de la plataforma empleada (Platform Independent Models -PMI), permitiendo la interoperabilidad entre sistemas dirigidos por modelos, reflejado en el intercambio de metadatos e integración entre aplicaciones. En este estándar, un metamodelo es definido como un modelo que permite dirigir el modelamiento del mismo modelo y también puede utilizarse para modelar metadatos.

MOF es el meta-meta modelo del Lenguaje Unificado de Modelado (UML: Unified Modeling Language) y a su vez UML se referencia para definir las bases de MOF.

- **MOF:** Utiliza paquetes para el agrupamiento de elementos relacionados de modelos, facilitando el re-uso y la importación, así como la combinación de características de metamodelos nuevas o re-usadas para crear lenguajes de modelado extendido.
- **Los principios de la aproximación MDA (dirigida por el modelo):** Según el OMG, MDA proporciona una solución para los cambios de negocio y de tecnología, permitiendo construir aplicaciones independientes de la plataforma e implementarlas en plataformas como CORBA, J2EE o Servicios Web.
- **MDA:** Promueve el uso eficiente de modelos de sistemas en el proceso de desarrollo de software, representa para los desarrolladores, una nueva manera de organizar y administrar arquitecturas empresariales, basada en la utilización

de herramientas de automatización de etapas en el ciclo de desarrollo y servicios, también permite definir los modelos y facilitar transformaciones paulatinas entre diferentes modelos como ser el modelo de análisis, el de diseño y el de comportamientos, entre otros. Es decir que, a partir de uno de los, podemos generar otro de menor abstracción.

- **El modelo de transformación de lenguaje QVT:** Transformación de modelos es una técnica clave utilizado en la arquitectura dirigida por modelos. Como su nombre indica QVT, el estándar OMG cubre transformaciones, vistas y consultas juntos. Consultas de modelo y las vistas de modelo puede ser visto como un tipo especial de transformación de modelos, siempre y cuando se utiliza una adecuada definición amplia de transformación de modelos: un modelo de transformación es un programa que funciona en los modelos.
- **El estándar QVT:** Define tres idiomas de transformación de modelos. Todos ellos operan sobre los modelos que se ajustan a Meta-Object Facility (MOF) 2.0 meta modelos; los estados de transformación que se utilizan meta modelos. Una transformación en cualquiera de los tres idiomas QVT en sí puede ser considerado como un modelo, conforme a uno del meta modelos especificados en la norma. El estándar QVT integra el OCL 2.0 estándar y se extiende también con características imperativas.
- **QVT-operacional:** Es un lenguaje imperativo diseñada para escribir transformaciones.
- **QVT-Relations:** Es un lenguaje declarativo diseñado para permitir que ambas transformaciones de modelos unidireccionales y bidireccionales a ser escritos. Una transformación encarna una relación de coherencia en conjuntos de modelos. La consistencia se puede comprobar mediante la ejecución de la transformación en el modo de checkonly; la transformación continuación, devuelve True si el conjunto de modelos es consistente de acuerdo a la transformación y False en caso contrario. La misma transformación puede ser utilizado en modo de aplicación para intentar modificar uno de los modelos de modo que el conjunto de modelos será consistente. El lenguaje QVT-Relations tiene tanto una textual y gráfica de una sintaxis concreta.

2.5.1 CARACTERÍSTICAS UWE

Las principales características en los que se fundamenta UWE son los siguientes:

- Una de las características de UWE es el uso de una notación estándar, para todos los modelos Lenguaje de modelado unificado UML.
- Definición de métodos: UWE presenta una definición de los pasos para la construcción de los diferentes modelos.
- Especificación de Restricciones: en la metodología UWE, se recomienda el uso de restricciones en su desarrollo.

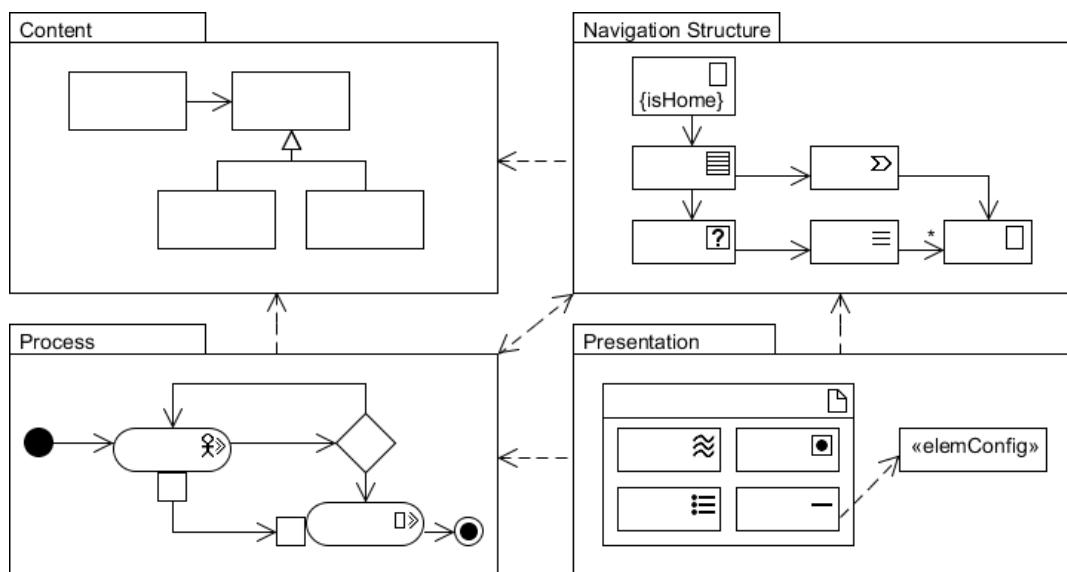
2.5.2 MODELOS DE UWE

UWE es una metodología dirigida y enfocada al modelado de aplicaciones Web, ya que está basada estrictamente en UML, esta metodología nos garantiza que sus modelos sean fáciles de entender para los que manejan UML, también nos proporcionan un esquema de trabajo que garantizara un proceso de desarrollo que nos ayudara de diferentes maneras.

Los modelos UWE son construidos en las diferentes etapas del proceso de desarrollo; la ingeniería de requerimientos, análisis, diseño e implementación y son utilizados para representar las diferentes visitas de la aplicación Web que corresponden a los diferentes enfoques.

- Modelo de requerimientos
- Modelo Conceptual
- Modelo Navegación
- Modelo de Presentación
- Modelo de Proceso

Figura 2.3.: Vista General De Modelos UWE.

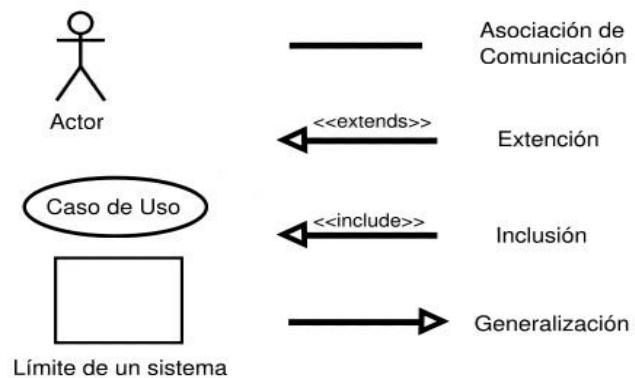


Fuente: Nolivos Y Coronel, (t-ESPE, 2013).

2.5.2.1 Modelo de Requerimientos

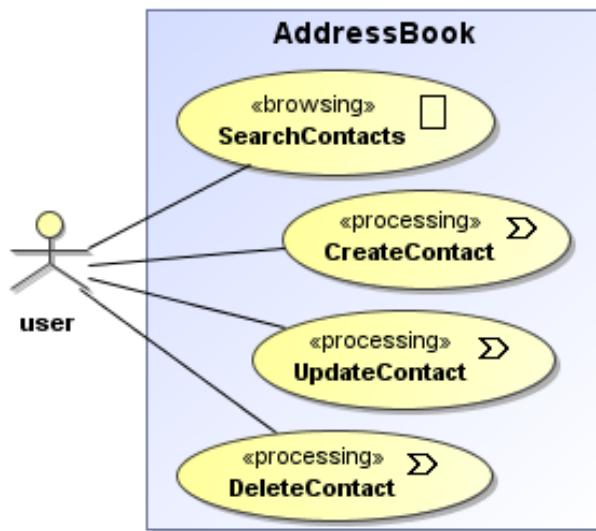
La fase de Análisis de Requerimientos realiza captura, analiza y especifica los requisitos del proyecto de los mismos mediante diagramas de casos de uso acompañado de documentación detallada, en simple palabras y básicamente, durante esta fase, se adquieren, reúnen y especifican las características funcionales y no funcionales que deberá cumplir la aplicación web, las necesidades de navegación, las necesidades de adaptación y las de interfaz de usuario, así como algunos requisitos adicionales.

Figura 2.4.: Estereotipos de Casos de Uso.



Fuente: (Ludwig-Maximilians-Universität München [UWE, 2016]).

Figura 2.5.: Modelo de Caso de Uso.



Fuente: Ludwig – Maximilians - Universität München [UWE,2016].

Los detalles pueden ser representados por objetos de nodos y pins de acciones, el objeto de nodo es para modelar clases de contenido y los pines son atributos.

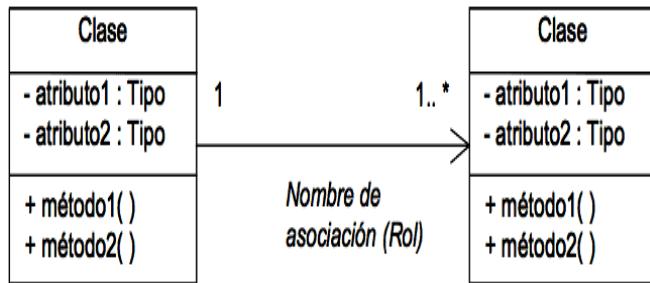
Durante la ingeniería de requisitos es usual determinar qué datos son representados donde y cuando, como estos estereotipos se utilizan para indicar elementos de presentación durante la etapa de ingeniería de requisitos, aspectos que caracterizan, pueden ser especificadas mediante valores etiquetados para estos mismos elementos.

2.5.2.2 Modelo Conceptual

Caracterizado por un modelo de dominio, que utiliza los requisitos que se detallan en los casos de uso, en esta etapa se representa el dominio del problema con un diagrama de clases de UML, que permiten determinar, métodos y atributos.

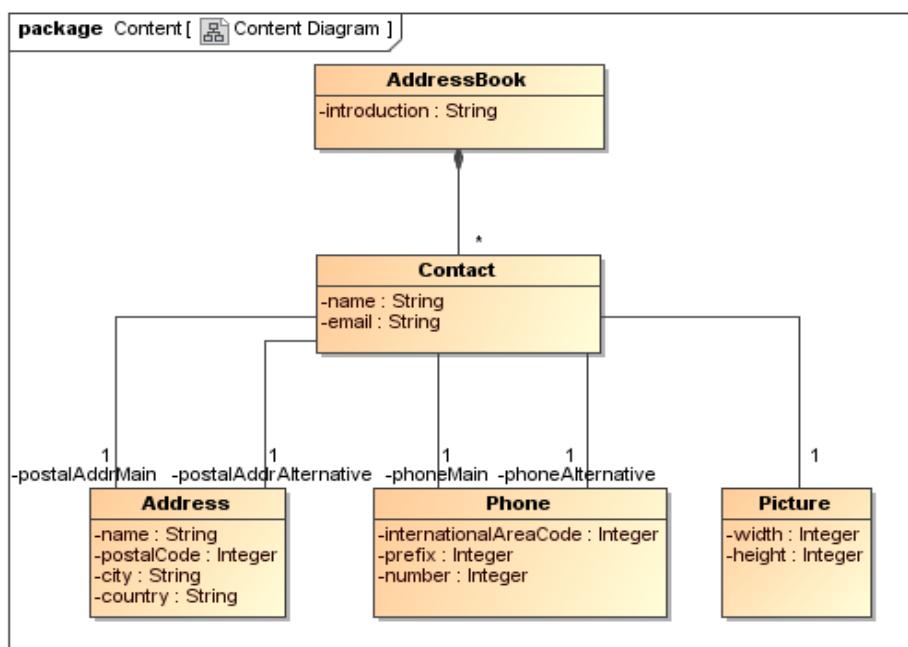
El propósito de este diagrama es construir un modelo del dominio que intenta no considerar el paseo de navegación, la presentación y los aspectos de interacción. Aspectos que se analizaran en los pasos respectivos de navegación y presentación de la planificación.

Figura 2.6.: Estereotipo de Diagrama de Contenido.



Fuente: Ludwig – Maximilians - Universität München [UWE,2016].

Figura 2.7.: Diagrama de Contenido.



Fuente: Ludwig-Maximilians-Universität München (UWE - 2016).

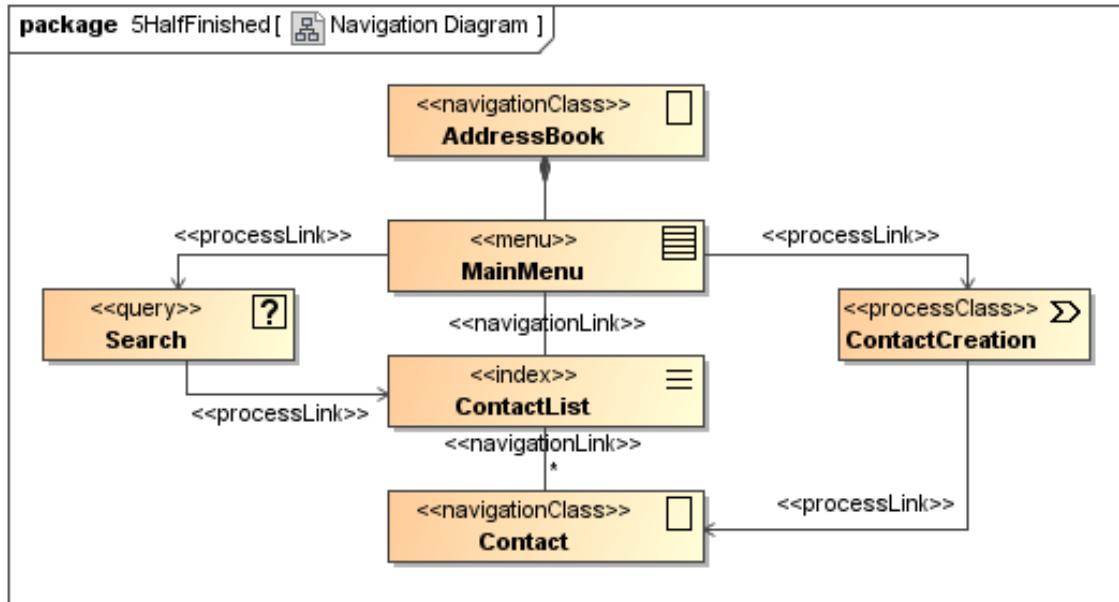
Este es un diagrama de clases UML normal, por lo tanto, es necesario pensar en las clases que se necesitan para su ejecución. Se requiere de información en cadena, relacionada en aspectos abstractos de tal forma que coadyuven en la integración de una clase total y relacionada.

2.5.2.3 Modelo de Navegación

Basado en el diagrama de la fase conceptual, donde se especifica los objetivos que serán visitados dentro de la aplicación web y la relación entre los mismos. Su objetivo principal es representar el diseño y estructura de las rutas de navegación al usuario para evitar la desorientación en el proceso de navegación.

Este modelo se destaca en el marco de UWE como el más importante, ya que representa elementos estáticos, a la vez que se pueden incorporar lineamiento semántico de referencia para las funcionalidades dinámicas de una aplicación Web.

Figura 2.8.: Diagrama de Navegación UML.



Fuente: LMU – Ludwig-Maximilians-Universität München (UWE-2016).

En un sistema para la web es útil saber cómo están enlazadas las páginas, esto significa que necesitamos un diagrama contenido en nodos (nodes) y enlaces (links).

Los nodos, son unidades de navegación y están conectados por medio de enlaces, pueden ser presentados en diferentes páginas o en una misma página.

La fase de navegación a su vez podemos dividirlo en dos áreas.

- **Modelo del espacio de navegación:** basada en lo estructurado en la fase de conceptualización, es decir en los diagramas de clases.
- **Modelos de la estructura de navegación:** Muestra la firma de navegación ante el espacio de navegación. Están constituidas por menús, índices, visitas guiadas, y formularios.
 - **Los índices** es la colección de objetos permitiendo una navegación directa.

- **Las visitas guiadas** compuesta por grupo de referencias, permitiendo una navegación secuencial.
- **Un menú** es un elemento parte de la navegación con un número específico de conexiones a otros objetos.
- **Un formulario** facilita al usuario ingresar información para completar las condiciones de selección de objetos pertenecientes a las colecciones de índices y visitas guiadas.

Figura 2.9.: Nombre y Símbolo de Estereotipos – Modelo de Navegación.

nombres de estereotipos y sus iconos	
□ clase de navegación	≡ menú
☰ índice	❓ pregunta
➡ visita guiada	▷ clase de proceso
➡ nodo externo	

Fuente: Nolivos y coronel (2013).

2.5.2.4 Modelo de Presentación

La fase de diseño de presentación tiene como objetivo la representación de las vistas del interfaz del usuario final, la representación gráfica de esta fase se encuentra basada en los diagramas realizados en las fases anteriores. Permite la especificación lógica de la aplicación Web, una presentación física puede ser construida sobre este método lógico. Representa la interfaz del usuario por medio de vistas estándares de interacción. Las clases del modelo de presentación representan páginas Web o parte de ellas, organizando la composición de los elementos de la interfaz de usuario y las jerarquías del modelo de presentación.

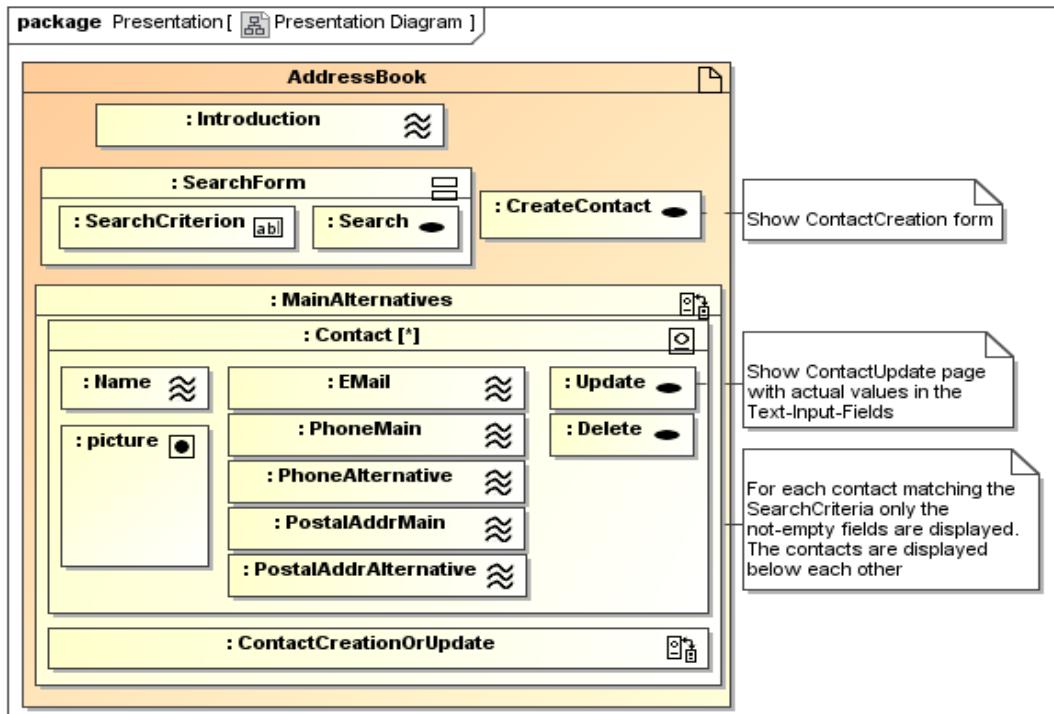
Figura 2.10.: Nombre y Símbolo de Estereotipos – Modelo de Presentación.

nombres de estereotipos y sus iconos	
grupo de presentación	página de presentación
texto	entrada de texto
ancla	fileUpload
botón	imagen
formulario	componente de cliente
alternativas de presentación	selección

Fuente: Nolivos y coronel (2013).

El diagrama de esta fase representa los objetivos de navegación y elementos de acceso, por ejemplo, en que marco o ventana se encuentran el contenido y que será remplazado cuando se accione un enlace. En la siguiente imagen podremos observar un ejemplo de un diagrama de presentación mediante UWE.

Figura 2.11.: Diseño de Presentación UWE.



Fuente: Ludwig-Maximilians-Universität München (UWE, 2014).

2.5.2.5 Modelo de Proceso

Los procesos proveen a una aplicación Web aspectos dinámicos. Un proceso está compuesto por uno o más subprocessos y/o acciones que el usuario ejecuta para alcanzar una meta, esto representará un cambio en el estado de la aplicación.

Existen diferentes notaciones UML propuestas para modelar procesos, entre las más conocidas están los diagramas de estados y diagramas de actividades. UWE utiliza diagramas de actividades las cuales proveen un mapa de las características funcionales del sistema por un lado y por otro representan el mapa de interacciones del usuario con el sistema. El modelamiento de procesos en UWE utiliza un subconjunto de elementos UML aplicado a diagramas de actividades y extendiendo su notación para modelar elementos como excepciones, eventos o nodos de actividades estructurados.

El modelamiento de procesos en UWE está dividido en las siguientes partes:

- Integración de procesos, que permite integrar la ejecución de procesos desde el modelo de navegación.
- Procesamiento de datos, que permite representar el acceso a datos por los procesos.
- Flujos de procesos, representados en flujos de procesos dinámicos, que ejecutan operaciones desde el modelo de contenido.

2.5.3 FACES O ETAPAS DE UWE

UWE es una metodología dirigida y enfocada al modelado de aplicaciones Web, ya que está basada estrictamente en UML, esta metodología nos garantiza que sus modelos sean fáciles de entender para los que manejan UML, también nos proporcionan un esquema de trabajo que garantizara un proceso de desarrollo que nos ayudara de diferentes maneras.

2.5.3.1 Análisis de Requisitos

Captura, analiza y especifica los requisitos del proyecto, en simple palabras y básicamente, durante esta fase, se adquieren, reúnen y especifican las características funcionales y no funcionales que deberá cumplir la aplicación web, las necesidades de

navegación, las necesidades de adaptación y las de interfaz de usuario, así como algunos requisitos adicionales.

2.5.3.2 Diseño del Sistema

Se basa en la especificación de requisitos producido por el análisis de los requerimientos, el diseño define cómo estos requisitos se cumplirán, la estructura que debe darse a la aplicación web.

2.5.3.3 Codificación del Software

Durante esta etapa se realizan las tareas que comúnmente se conocen como programación, que consiste, esencialmente, en llevar a código fuente, en el lenguaje de programación elegido, todo lo diseñado en la fase anterior.

2.5.3.4 Pruebas

Pruebas, las pruebas se utilizan para asegurar el correcto funcionamiento de secciones de código.

2.5.3.5 Instalación o Fase de Implementación

Es el proceso por el cual los programas desarrollados son transferidos apropiadamente al computador destino, inicializados y eventualmente, configurados. Todo ello con el propósito de ser ya utilizados por el usuario final.

2.5.3.6 Mantenimiento

Es el proceso de control, mejora y optimización del software ya desarrollado e instalada.

2.5.4 DIAGRAMAS DE UWE

2.5.4.1 Diagrama de Clase

Representan un conjunto de elementos del modelo que son replantados tomando en cuenta sus atributos, sus métodos las relaciones que puedan existir entre ellas.

Estos diagramas son utilizados durante el proceso de análisis y diseño de sistemas de software, donde se crea el diseño conceptual que manejará la información del sistema y los componentes que se encargaran del funcionamiento y la relación entre cada uno de ellos (Zenteno, 2016).

2.5.4.2 Diagrama de Objetos

En UML, los diagramas de objetos muestran un instante en el sistema y las relaciones entre distintas instancias. Algunas líneas generales en comparación con el diagrama de clases son las siguientes.

- El diagrama de objetos utiliza notaciones similares a los usados en el diagrama de clases.
- Los diagramas de objetos se utilizan para modelar los elementos que están presentes en un diagrama de clases.
- El diagrama de objetos muestra los clasificadores reales del sistema y las relaciones entre ellos en un punto específico del tiempo.
- Los diagramas de objetos se pueden instanciar como diagrama de clases, despliegue, componentes e, incluso, casos de uso.
- En ninguno de los dos diagramas se muestran los mensajes entre los elementos que colaboran, ya que se trata de diagramas estructurales.

Elementos de Diagrama de Objeto

- **Objetos:** Cada objeto se representa con un rectángulo con su nombre y el de su clase en la parte superior subrayados y separados por dos puntos. En caso de ser un objeto anónimo no se escribe su nombre, dejando solo el de la clase.
- **Atributos:** De igual forma que el diagrama de clases, se muestra en un compartimento en la parte inferior del nombre del objeto. A diferencia de las clases, los atributos pueden tener valores asignados a ellos:
- **Vínculos:** Son asociaciones entre dos objetos y se representan con los mismos elementos que en el diagrama de clases. Por ejemplo, una asociación.

2.5.4.3 Diagramas de Caso de Uso

Un caso de uso es una descripción de las acciones de un sistema desde el punto de vista del usuario. Es una herramienta valiosa dado que es una técnica de aciertos y errores para obtener los requerimientos del sistema, justamente desde el punto de vista del usuario. (Cevallos, 2015)

Los diagramas de caso de uso modelan la funcionalidad del sistema usando actores y casos de uso. Los casos de uso son servicios o funciones provistas por el sistema para sus usuarios.

- **Sistema:** El rectángulo representa los límites del sistema que contiene los casos de uso. Los actores se ubican fuera de los límites del Sistema.
- **Caso de uso:** Se representan con óvalos. La etiqueta en el óvalo indica la función del sistema.
- **Actor:** Un diagrama de caso de uso contiene los símbolos del actor y del caso de uso, junto con líneas conectadoras. Los actores son similares a las entidades externas; existen fuera del sistema. El término actor se refiere a un rol específico de un usuario del sistema.
- **Extend:** Cuando un caso de uso especializa a otro extendiendo su funcionalidad.
- **Include:** Cuando un caso de uso utiliza a otro.

Se representa como una línea que une a los dos casos de uso relacionados, con una flecha en forma de triángulo y con una etiqueta <<extend>> o <<include>> según sea el tipo de relación.

2.5.4.4 Diagramas de Secuencia

Un diagrama de secuencias muestra la interacción de un conjunto de objetos de una aplicación a través del tiempo, en el cual se indicarán los módulos o clases que formaran parte del programa y las llamadas que se hacen cada uno de ellos para realizar una tarea determinada, por esta razón permite observar la perspectiva cronológica de las interacciones. Es importante recordar que el diagrama de secuencias se realiza a partir de la descripción de un caso de uso (Ceballos, 2015).

Entre las ventajas que tiene la elaboración de un diagrama de secuencias están las siguientes:

- **Rol de la Clase:** El rol de la clase describe la manera en que un objeto se va a comportar en el contexto. No se listan los atributos del objeto.

- **Activación:** Los cuadros de activación representan el tiempo que un objeto necesita para completar una tarea.
- **Mensajes:** Son flechas que representan comunicaciones entre objetos. Las medias flechas representan mensajes asincrónicos. Los mensajes asincrónicos son enviados desde un objeto que no va a esperar una respuesta del receptor para continuar con sus tareas.
- **Líneas de Vida:** Las líneas de vida son verticales y en línea de puntos, ellas indican la presencia del objeto durante el tiempo.
- **Destrucción de Objetos:** Los objetos pueden ser eliminados tempranamente usando una flecha etiquetada “<<destruir>>” que apunta a una X.
- **Loops:** Una repetición o loop en un diagrama de secuencias, es representado como un rectángulo. La condición para abandonar el loop se coloca en la parte inferior entre corchetes [].

2.5.4.5 Diagramas de Colaboración

“Un diagrama de colaboración es una forma alternativa al diagrama de secuencia de mostrar un escenario. Este tipo de diagrama muestra las interacciones entre objetos organizados entorno a los objetos y los enlaces entre ellos” (Fowler & Kendall, 1999).

El diagrama de colaboración es un tipo de diagrama de interacción cuyo objetivo es describir el comportamiento dinámico del sistema de información mostrando cómo interactúan los objetos.

- **Objetos o Roles:** nodos del grafo.
- **Enlaces o comunicaciones:** arcos del grafo.
- **Mensajes:** llevan número de secuencia y flecha dirigida.
- **Anidamiento:** se utiliza la numeración decimal.
- **Iteración:** colocar un * antes del número de secuencia y una cláusula de condición, si es necesario.
- **Bifurcación:** los caminos alternativos tendrán el mismo número de secuencia, seguido del número de sub secuencia, y se deben distinguir por una condición.

2.5.4.6 Diagrama de Estado

Los diagramas de estado son una técnica conocida para describir el comportamiento de un sistema. Describen todos los estados posibles en los que puede entrar un objeto, como resultado de los eventos que llegan a él. En la mayor parte de las técnicas orientadas a objetos, los diagramas de estados se dibujan para una sola clase, mostrando el comportamiento de un solo objeto durante todo su ciclo de vida. (Fowler y Kendall, 1999).

El diagrama de estado se usa para dar forma al comportamiento de un objeto, de una clase. Se representa la secuencia de estados que un objeto de la clase tiene durante su vida, según las acciones que van sucediendo.

- **Pseudoestado de opción:** es un símbolo con forma de diamante indicando una condición dinámica con resultados potenciales ramificados.
- **Punto de salida:** a partir de donde se sale de un estado compuesto, o de una máquina de estados. Se representa con un círculo con una «X».
- **Evento:** digamos que es lo que activa una transición.
- **Estado final:** Se representa con un círculo negro con una flecha de transición.
- **Protección:** una condición booleana que permite o detiene una transición.
- **Estado:** un rectángulo redondeado que muestra el estado en que se encuentra un objeto.
- **Sub estado:** un estado contenido dentro de la región de un estado compuesto.
- **Disparador:** es un mensaje que mueve un objeto de estado en estado.
- **Transición:** una flecha que corre de un estado a otro que indica un estado cambiante.
- **Estado compuesto:** es un estado que contiene sub-estados.
- **Comportamiento de transición:** un tipo de comportamiento resultante que ocurre durante la transición de un estado. Se escribe arriba de la flecha de transición.

2.5.4.7 Diagrama de Actividad

Las partes interesadas tienen muchos asuntos que manejar, por lo que es importante una comunicación clara y concisa. Los diagramas de actividades ayudan a que las

personas en las áreas de negocios y desarrollo de una organización se integren para comprender el mismo proceso y comportamiento. Usarás un conjunto de símbolos especializados incluidos aquellos para pasos de inicio, finalización, fusión y recepción en el flujo para crear un diagrama de actividades, lo cual cubriremos con más detalle dentro de esta guía de diagramas de actividades.

Antes de empezar a crear un diagrama de actividades, debes comprender primero su composición. Algunos de los componentes más comunes de un diagrama de actividades incluyen:

- **Acción:** Un paso en la actividad en el que los usuarios o el software realizan una tarea dada. En Lucidchart, las acciones se representan a través de rectángulos con aristas redondeadas.
- **Nodo de decisión:** Una rama condicional en el flujo que se representa con un diamante. Incluye una sola entrada y dos o más salidas.
- **Flujos de control:** Otro nombre para los conectores que muestran el flujo entre pasos en el diagrama.
- **Nodo inicial:** Simboliza el inicio de la actividad. El nodo inicial se representa con un círculo negro.
- **Nodo terminal:** Representa el paso final en la actividad. El nodo terminal se representa por medio de un círculo negro de contorno blanco.

2.5.4.8 Diagrama de Componentes

“El diagrama de componentes proporciona una visión física de la construcción del sistema de información. Muestra la organización de los componentes software, sus interfaces y las dependencias entre ellos” (Cillero, 2013).

Un componente es un módulo de software que puede ser código fuente, código binario, un ejecutable, o una librería con una interfaz definida. Una interfaz establece las operaciones externas de un componente, las cuales determinan una parte del comportamiento del mismo. Además, se representan las dependencias entre componentes o entre un componente y la interfaz de otro, es decir uno de ellos usa los servicios o facilidades del otro.

- **Componente:** Un componente se representa como un rectángulo, con dos pequeños rectángulos superpuestos perpendicularmente en el lado izquierdo. Para distinguir distintos tipos de componentes se les puede asignar un estereotipo, cuyo nombre estará dentro del símbolo: << ... >>
- **Interfaz:** Se representa como un pequeño círculo situado junto al componente que lo implementa y unido a él por una línea continua. La interfaz puede tener un nombre que se escribe junto al círculo. Un componente puede proporcionar más de una interfaz.
- **Paquete:** Un paquete se representa con un ícono de carpeta (ver Diagrama de Paquetes).
- **Relación de dependencia:** Una relación de dependencia se representa mediante una línea discontinua con una flecha que apunta al componente o interfaz que provee del servicio o facilidad al otro. La relación puede tener un estereotipo que se coloca junto a la línea, entre el símbolo: <<...>>.

2.6 INGENIERIA DE SOFTWARE

La ingeniería de software es la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento de software; es decir, la aplicación de la ingeniería al software (Pressman, 2010). La ingeniería de software es una tecnología con varias capas, cualquier enfoque de ingeniería (incluso la de software) debe basarse en un compromiso organizacional con la calidad. La administración total de la calidad, Six Sigma y otras filosofías similares alimentan la cultura de mejora continua, y es esta cultura la que lleva en última instancia al desarrollo de enfoques cada vez más eficaces de la ingeniería de software. El fundamento en el que se apoya la ingeniería de software es el compromiso con la calidad (Pressman, 2010).

Figura 2.12.: Capa de la ingeniería de software.



Fuente: (Pressman, 2010).

El fundamento para la ingeniería de software es la capa proceso. El proceso de ingeniería de software es el aglutinante que une las capas de la tecnología y permite el desarrollo racional y oportuno del software de cómputo. El proceso define una estructura que debe establecerse para la obtención eficaz de tecnología de ingeniería de software. El proceso de software forma la base para el control de la administración de proyectos de software, y establece el contexto en el que se aplican métodos técnicos, se generan productos del trabajo (modelos, documentos, datos, reportes, formatos, etc.), se establecen puntos de referencia, se asegura la calidad y se administra el cambio de manera apropiada (Pressman, 2010).

Los métodos de la ingeniería de software proporcionan la experiencia técnica para elaborar software. Incluyen un conjunto amplio de tareas, como comunicación, análisis de los requerimientos, modelación del diseño, construcción del programa, pruebas y apoyo. Los métodos de la ingeniería de software se basan en un conjunto de principios fundamentales que gobiernan cada área de la tecnología e incluyen actividades de modelación y otras técnicas descriptivas (Pressman, 2010).

Las herramientas de la ingeniería de software proporcionan un apoyo automatizado o semiautomatizado para el proceso y los métodos. Cuando se integran las herramientas de modo que la información creada por una pueda ser utilizada por otra, queda establecido un sistema llamado ingeniería de software asistido por computadora que apoya el desarrollo de software (Pressman, 2010).

2.6.1 PROCESO DE SOFTWARE

Un proceso es un conjunto de actividades, acciones y tareas que se ejecutan cuando va a crearse algún producto del trabajo. Una actividad busca lograr un objetivo amplio (por ejemplo, comunicación con los participantes) y se desarrolla sin importar el dominio de la aplicación, tamaño del proyecto, complejidad del esfuerzo o grado de rigor con el que se usará la ingeniería de software. Una acción (diseño de la arquitectura) es un conjunto de tareas que producen un producto importante del trabajo (por ejemplo, un modelo del diseño de la arquitectura). Una tarea se centra en un objetivo pequeño, pero bien definido (por ejemplo, realizar una prueba unitaria) que produce un resultado tangible (Pressman, 2010).

En el contexto de la ingeniería de software, un proceso no es una prescripción rígida de cómo elaborar software de cómputo. Por el contrario, es un enfoque adaptable que permite que las personas que hacen el trabajo (el equipo de software) busquen y elijan el conjunto apropiado de acciones y tareas para el trabajo. Se busca siempre entregar el software en forma oportuna y con calidad suficiente para satisfacer a quienes patrocinaron su creación y a aquellos que lo usarán (Pressman, 2010).

La estructura del proceso establece el fundamento para el proceso completo de la ingeniería de software por medio de la identificación de un número pequeño de actividades estructurales que sean aplicables a todos los proyectos de software, sin importar su tamaño o complejidad. Además, la estructura del proceso incluye un conjunto de actividades sombrilla que son aplicables a través de todo el proceso del software. Una estructura de proceso general para la ingeniería de software consta de cinco actividades: (Pressman, 2010).

- **Comunicación:** Antes de que comience cualquier trabajo técnico, tiene importancia crítica comunicarse y colaborar con el cliente (y con otros participantes). Se busca entender los objetivos de los participantes respecto del proyecto, y reunir los requerimientos que ayuden a definir las características y funciones del software (Pressman, 2010).
- **Planeación:** Cualquier viaje complicado se simplifica si existe un mapa. Un proyecto de software es un viaje difícil, y la actividad de planeación crea un

“mapa” que guía al equipo mientras viaja. El mapa llamado plan del proyecto de software define el trabajo de ingeniería de software al describir las tareas técnicas por realizar, los riesgos probables, los recursos que se requieren, los productos del trabajo que se obtendrán y una programación de las actividades (Pressman, 2010).

- **Modelado:** Ya sea usted diseñador de paisaje, constructor de puentes, ingeniero aeronáutico, carpintero o arquitecto, a diario trabaja con modelos. Crea un “bosquejo” del objeto por hacer a fin de entender el panorama general cómo se verá arquitectónicamente, cómo ajustan entre sí las partes constituyentes y muchas características más. Si se requiere, re-fina el bosquejo con más y más detalles en un esfuerzo por comprender mejor el problema y cómo resolverlo. Un ingeniero de software hace lo mismo al crear modelos a fin de entender mejor los requerimientos del software y el diseño que los satisfará (Pressman, 2010).
- **Construcción:** Esta actividad combina la generación de código (ya sea manual o automatizada) y las pruebas que se requieren para descubrir errores en éste (Pressman, 2010).
- **Despliegue:** El software (como entidad completa o como un incremento parcialmente terminado) se entrega al consumidor que lo evalúa y que le da retroalimentación, misma que se basa en dicha evaluación (Pressman, 2010).

Estas cinco actividades estructurales genéricas se usan durante el desarrollo de programas pequeños y sencillos, en la creación de aplicaciones web grandes y en la ingeniería de sistemas enormes y complejos basados en computadoras. Los detalles del proceso de software serán distintos en cada caso, pero las actividades estructurales son las mismas (Pressman, 2010).

Para muchos proyectos de software, las actividades estructurales se aplican en forma iterativa a medida que avanza el proyecto. Es decir, la comunicación, la planeación, el modelado, la construcción y el despliegue se ejecutan a través de cierto número de repeticiones del proyecto. Cada iteración produce un incremento del software que da a los participantes un subconjunto de características y funcionalidad generales del

software. Conforme se produce cada incremento, el software se hace más y más completo (Pressman, 2010).

2.7 HERRAMIENTAS

2.7.1 LENGUAJE DE PROGRAMACIÓN

2.7.1.1 Php

PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.

Lo que distingue a PHP de algo del lado del cliente como Javascript es que el código es ejecutado en el servidor, generando HTML y enviándolo al cliente. El cliente recibirá el resultado de ejecutar el script, aunque no se sabrá el código subyacente que era. El servidor web puede ser configurado incluso para que procese todos los ficheros HTML con PHP, por lo que no hay manera de que los usuarios puedan saber qué se tiene debajo de la manga.

Figura 2.13.: Descripción de uso PHP.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Ejemplo</title>
  </head>
  <body>
    <?php
      echo "¡Hola, soy un script de PHP!";
    ?>
  </body>
</html>
```

Fuentes: Manual de PHP, (PHP, 2020).

Lo mejor de utilizar PHP es su extrema simplicidad para el principiante, pero a su vez ofrece muchas características avanzadas para los programadores profesionales. No sienta miedo de leer la larga lista de características de PHP. En unas pocas horas podrá empezar a escribir sus primeros scripts.

2.7.1.2 Html

HTML es un lenguaje de marcado que se utiliza para el desarrollo de páginas de Internet. Se trata de la sigla que corresponden a HyperText Markup Language, es decir, Lenguaje de Marcas de Hipertexto, contiene instrucciones especiales que hacen que el texto pueda enlazar con otro contenido de internet, estos se suelen llamar hipervínculo y son muy importantes en la composición de la red. (Prescot, 2015).

Figura 2.14.: Estructura básica de HTML.

```
<!DOCTYPE HTML >
<HTML>
  <HEAD>
    <TITLE>Mi primer documento HTML</TITLE>
  </HEAD>
  <BODY>
    <P>¡Hola mundo!
  </BODY>
</HTML>
```

Fuentes: Manual de HTML, (HTML, 2020).

El consorcio World Wide Web público un documento sobre HTML.5 (que es la quinta versión de este lenguaje) en el 2011, en él se detallan algunas de las características de esta versión, “HTML.5 permite crear aplicaciones web que son capaces de interactuar con los datos locales y servidores de manera sencilla, además de brindar soporte multimedia”. Al principio surgió como una respuesta de incompatibilidad de versiones anteriores, una diferencia era la no dependencia de APIs ni complemento de terceros, que daban problemas de compatibilidad con todos los navegadores al ofrecer una interfaz.

2.7.2 FRAMEWORK

2.7.2.1 BootStrap

Bootstrap, es un framework originalmente creado por Twitter, que permite crear interfaces web con CSS y JavaScript, cuya particularidad es la de adaptar la interfaz del sitio web al tamaño del dispositivo en que se visualice. Es decir, el sitio web se adapta automáticamente al tamaño de una PC, una Tablet u otro dispositivo. Esta técnica de diseño y desarrollo se conoce como Responsive Design o Diseño Adaptativo.

Este Framework te abstrae de tener que preocuparte por las medias y los porcentajes en tus CSS para hacer una web, facilitando la programación del site. Además, se basa en la simplicidad de sus interfaces, lo cual es una tendencia del mercado, en las que tiende a diseño plano, botones grandes, etc... para facilitar la usabilidad en los dispositivos más pequeño, atendiendo a la User Experience.

Actualmente es uno de los Framework más usados, si no el que más, pero ¿merece la pena usarlo?

Ventajas

- Utiliza componentes y servicios creados por la comunidad web, tales como: HTML5 shim, Normalize.css, OOCSS (CSS orientado a objetos), jQuery UI, LESS y GitHub.
- Es un conjunto de buenas prácticas.
- El famoso Grid system, que por defecto incluye 12 columnas fijas o fluidas, dependiendo de si tu diseño. Está orientado siguiente el concepto Mobile First.
- **Soporte:** Hay una enorme comunidad que soporta este desarrollo y cuenta con implementaciones externas como WordPress, Drupal, SASS o jQuery UI.
- **Comodidad y rapidez:** Herramienta sencilla y ágil para construir sitios web e interfaces.
- **Componentes:** Mucha variedad de plantillas y temas.

Desventajas

- **Aprendizaje:** Es necesario adaptarse a su forma de trabajo, si bien su curva de aprendizaje es liviana, deberás comprender y familiarizarte con su estructura y nomenclatura.
- **Adaptación:** Debes adaptar tu diseño a un grid de 12 columnas, que se modifican según el dispositivo. Aquí empiezan los problemas, Bootstrap por defecto te trae anchos, márgenes y altos de línea, y realizar cambios específicos es por decir, un poco tedioso.
- **Mantenimiento:** Es complicado, cambiar de versión si has realizado modificaciones profundas sobre el core.

- **Ampliar componentes:** Si necesitas añadir componentes que no existen, debes hacerlos tú mismo en CSS y cuidar de que mantenga coherencia con tu diseño y cuidando el responsive.
- **Pesado:** No es ligero y además para algunas funcionalidades, será necesario tener que usar JavaScript y jQuery.

Personalmente creo que BootStrap es una gran opción en cuanto a lo que frameworks para el front end se refiere. Estas son algunas de las razones por las que yo utilizaría Bootstrap:

- Posee ciertos elementos realmente bien configurados y su forma de entender y manejar los diferentes tamaños de pantalla se hace realmente cómoda y útil.
- Es rápido y aunque no en un primer momento, simplemente con realizar dos o tres proyectos se convierte en una herramienta realmente intuitiva.
- Cuenta con un gran número de elementos ya personalizados con una apariencia al día de las páginas web más actuales.
- Cuenta con una gran comunidad que se ofrecerá a resolver cualquiera de nuestras dudas acerca de Bootstrap.
- Como usuarios de BootStrap podemos contar con infinidad de proyectos ya realizados y de los que sin duda podremos aprender

2.7.2.2 Jquery

JQuery es una librería de JavaScript (JavaScript es un lenguaje de programación muy usado en desarrollo web). Esta librería de código abierto, simplifica la tarea de programar en JavaScript y permite agregar interactividad a un sitio web sin tener conocimientos del lenguaje. Características de Jquery.

A continuación, se presentan las principales características de Jquery:

- Es un software libre, por lo que puede ser empleado por cualquier usuario de manera gratuita.
- Su librería permite actualizaciones constantes y rápidas.
- Posee un código abierto y compatible con diferentes navegadores.
- Es fácil de usar, por lo que permite ahorrar tiempo y esfuerzo.

- Su desempeño se integra muy bien con AJAX, una técnica de desarrollo web.
- Permite realizar animaciones, efectos y personalizaciones.
- Es compatible con diferentes buscadores como Google Chrome, Microsoft Edge,

Ventajas

- Incluye, listos para usar, elementos como botones, barras de herramientas, formularios, listas y tipografías especialmente optimizadas y estilizadas para aplicaciones móviles. Además, cuenta con un sistema de bloques ideal para construir plantillas.
- Nos permite construir aplicaciones aptas para todo tipo de plataformas casi sin esfuerzo, evitándonos preocupaciones por problemas de compatibilidad.
- Es muy fácil de aprender y utilizar. Si se tienen adecuados conocimientos de HTML y CSS, se pueden construir aplicaciones robustas en muy poco tiempo. El sitio web de jQuery cuenta con abundante documentación para aprender.
- Tiene soporte para Ajax, una tecnología para crear aplicaciones dinámicas que se ejecuten del lado cliente.
- No requiere del uso de programas especiales, a diferencia de otras tecnologías, como Adobe Flash, que exigen una aplicación para desarrollar sobre ellas.

Desventajas

- Las funciones que ofrece son muchas, pero resultan difíciles de personalizar. Su aspecto visual es estandarizado y no se integra con el de la plataforma. En algunos casos, no queda otra opción que usar JavaScript simple para adaptar la aplicación a nuestras necesidades.
- Como es necesario invocar a un archivo para utilizar sus funciones, ralentiza ligeramente la carga de la página.
- Su manejo de CSS suele resultar innecesariamente complejo. A veces cuesta saber qué clases utilizar.
- No existen muchas plantillas prediseñadas sobre las cuales empezar a construir nuestra aplicación.

2.7.2.3 Phalcon

Hay muchos frameworks para PHP hoy en día, pero ninguno como Phalcon. Casi todos los programadores preferimos usar un framework. Esto debido a que nos proporcionan una gran funcionalidad que está probada y lista para usar, al mismo tiempo no repitiéndonos y reusando código. Sin embargo, los frameworks requieren incluir muchos archivos e interpretar miles de líneas de código en cada petición. Adicionalmente frameworks orientados a objetos agregan una cantidad considerable de overhead. Todas estas operaciones hacen que las aplicaciones sean más lentas por consiguiente impactando la experiencia de usuario de manera negativa.

Características

- Los componentes están libremente acoplados. Con Phalcon, nada está impuesto: tienes la libertad de usar todo el framework, o solo las partes que necesites.
- Optimizaciones de bajo nivel ayudan a reducir la sobrecarga requerida para correr aplicaciones MVC.
- Las operaciones con base de datos se efectúan con la máxima eficiencia al usar un ORM para PHP escrito en C.
- Phalcon accede directamente a las estructuras internas de PHP optimizando además cada ejecución.
- Usa un framework PHP basado en C.
- Aprovecha el fantástico rendimiento de Phalcon y la reduce sobrecarga de recursos.
- Utiliza solo los módulos y bibliotecas que necesitas.

Phalcon es un framework full-stack de código abierto para PHP 5 escrito como extensión en C, optimizada para alto rendimiento. No necesitas aprender o usar C, toda su funcionalidad está expuesta como clases PHP listas para usar. Phalcon está también débilmente acoplado permitiéndote usar sus clases como componentes de acuerdo a como tu aplicación lo requiera.

2.7.3 GESTOR DE BASES DE DATOS MYSQL

Es un sistema de gestión de bases de datos relacional, fue creada por la empresa sueca MySQL AB, la cual tiene el copyright del código fuente del servidor SQL, así como también de la marca. MySQL es un software de código abierto, licenciado bajo la GPL de la GNU, aunque MySQL AB distribuye una versión comercial, en lo único que se diferencia de la versión libre, es en el soporte técnico que se ofrece, y la posibilidad de integrar este gestor en un software propietario ya que, de otra manera, se vulneraría la licencia GPL. El lenguaje de programación que utiliza MySQL es Structured Query Language (SQL) que fue desarrollado por IBM en 1981 y desde entonces es utilizado de forma generalizada en las bases de datos relacionales. (Nipotti, 2011)

Características principales

Inicialmente, MySQL carecía de algunos elementos esenciales en las bases de datos relacionales, tales como integridad referencial y transacciones. A pesar de esto, atrajo a los desarrolladores de páginas web con contenido dinámico, debido a su simplicidad, de tal manera que los elementos faltantes fueron complementados por la vía de las aplicaciones que la utilizan. Poco a poco estos elementos faltantes, están siendo incorporados tanto por desarrolladores internos, como por desarrolladores de software libre.

En las últimas versiones se pueden destacar las siguientes características principales:

- El principal objetivo de MySQL es velocidad y robustez.
- Soporta gran cantidad de tipos de datos para las columnas.
- Gran portabilidad entre sistemas, puede trabajar en distintos sistemas operativos.
- Cada base de datos cuenta con 3 archivos: Uno de estructura, uno de datos y uno de índice y soporta hasta 32 índices por tabla.
- Aprovecha la potencia de sistemas multiproceso, gracias a su implementación multihilo.

- Flexible sistema de contraseñas (passwords) y gestión de usuarios, con un muy buen nivel de seguridad en los datos.
- El servidor soporta mensajes de error en distintas lenguas.

Ventajas

- Velocidad al realizar las operaciones, lo que le hace uno de los gestores con mejor rendimiento.
- Bajo costo en requerimientos para la elaboración de bases de datos, ya que debido a su bajo consumo puede ser ejecutado en una máquina con escasos recursos sin ningún problema.
- Facilidad de configuración e instalación.
- Soporta gran variedad de Sistemas Operativos.
- Baja probabilidad de corromper datos, incluso si los errores no se producen en el propio gestor, sino en el sistema en el que está.
- Conectividad y seguridad.

Desventajas

- Un gran porcentaje de las utilidades de MySQL no están documentadas.
- No es intuitivo, como otros programas (ACCESS).

2.7.4 SERVIDOR APACHE

Apache Server es un software de servidor web gratuito y de código abierto para plataformas Unix, Microsoft, Macintosh y otros, con el cual se ejecutan el 46% de los sitios web de todo el mundo. Es mantenido y desarrollado por la Apache Software Foundation.

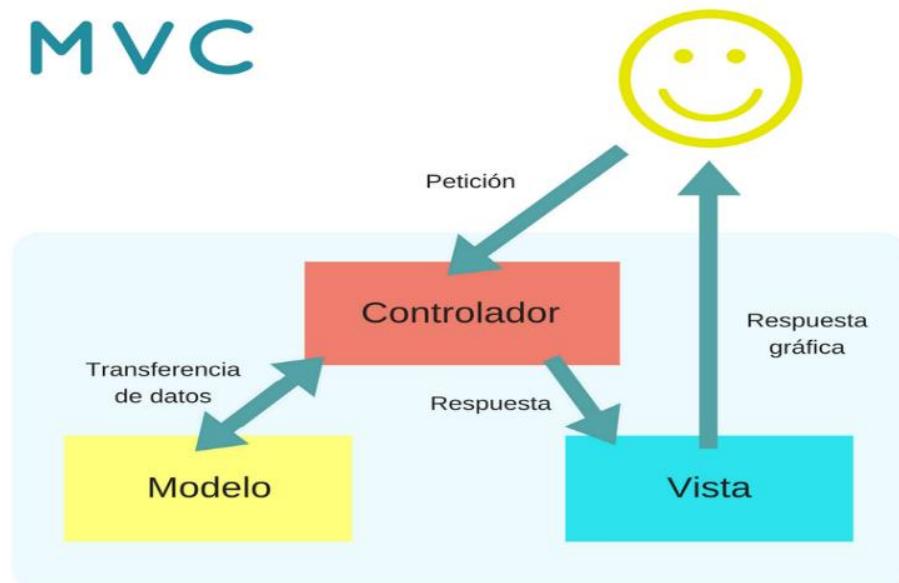
Es el servidor más utilizado, aunque ha vivido tiempos mejores. Parte de su éxito se debe a que es multiplataforma y a su estructura modular, que permite emplear diversos lenguajes en el lado del servidor (PHP, Python y Perl principalmente), así como incorporar características como la comprensión de datos, las conexiones seguras y la utilización de URLs amigables. (Pino, 2010)

Apache tiene amplia aceptación en la red, desde 1996, Apache es el servidor HTTP más usado ya que su papel fue fundamental en el desarrollo de la World Wide Web y alcanzó su máxima cuota de mercado en 2005, siendo el servidor empleado en el 70% de los sitios web en el mundo. Sin embargo, ha sufrido un descenso en su cuota de mercado en los últimos años, en 2009 se convirtió en el primer servidor web que alojó más de 100 millones de sitios web.

2.7.5 ARQUITECTURA MVC

MVC es una propuesta de arquitectura del software utilizada para separar el código por sus distintas responsabilidades, manteniendo distintas capas que se encargan de hacer una tarea muy concreta, lo que ofrece beneficios diversos, se usa inicialmente en sistemas donde se requiere el uso de interfaces de usuario, aunque en la práctica el mismo patrón de arquitectura se puede utilizar para distintos tipos de aplicaciones. Surge de la necesidad de crear software más robusto con un ciclo de vida más adecuado, donde se potencie la facilidad de mantenimiento, reutilización del código y la separación de conceptos. (Alvarez, 2019)

Figura 2.15.: Composición MVC.



Fuente: Recuperado de “Que es MVC”, (Alvarez, 2019).

2.7.5.1 Modelo

Es la capa donde se trabaja con los datos y su estructura, por tanto, contendrá mecanismos para acceder a la información y también para actualizar su estado. Los datos los tendremos habitualmente en una base de datos, por lo que en los modelos tendremos todas las funciones que accederán a las tablas, no obstante, cabe mencionar que cuando se trabaja con MVC lo habitual también es utilizar otras librerías como PDO o algún ORM como Doctrine, que nos permita trabajar con abstracción de bases de datos y persistencia en objetos. Por ello, en vez de usar directamente sentencias SQL, que suelen depender del motor de base de datos con el que se esté trabajando, se utiliza un dialecto de acceso a datos basado en clases y objetos.

2.7.5.2 Vista

Las vistas, como su nombre nos hacen entender, contienen el código de nuestra aplicación que va a producir la visualización de las interfaces de usuario, o sea, el código que nos permitirá renderizar los estados de nuestra aplicación en HTML. En las vistas nada más tenemos los códigos HTML y PHP que nos permite mostrar la salida, generalmente se trabaja con los datos, sin embargo, no se realiza un acceso directo a éstos. Las vistas requerirán los datos a los modelos y ellas se generarán la salida, tal como nuestra aplicación requiera.

2.7.5.3 Controlador

Contiene el código necesario para responder a las acciones que se solicitan en la aplicación, como visualizar un elemento, realizar una compra, una búsqueda de información, etc., es una capa que sirve de enlace entre las vistas y los modelos, respondiendo a los mecanismos que puedan requerirse para implementar las necesidades de nuestra aplicación. Sin embargo, su responsabilidad no es manipular directamente datos, ni mostrar ningún tipo de salida, sino servir de enlace entre los modelos y las vistas para implementar las diversas necesidades del desarrollo.

Calidad de software

2.8 CALIDAD DE SOFTWARE

“La gestión de calidad proporciona una comprobación independiente sobre el proceso de desarrollo de software. El proceso de gestión de calidad verifica los entregables del

proyecto para garantizar que sean conscientes con los estándares y las metas de la organización” (Somerville, 2011).

2.8.1 MÉTRICAS DE MCCALL

Este modelo es considerado por primera vez en 1977 por McCall. Destinado a ser utilizado durante el proceso de desarrollo de sistemas, demostró muy temprano como puente entre los usuarios y los desarrolladores, concilia los puntos de la vista de los usuarios con las prioridades de los desarrolladores. Con una perspectiva de visión basada en los criterios de la evaluación de la calidad.

Los factores desarrollados según el modelo de McCall, se centra en tres aspectos importantes de unos productos de software:

- Sus características operativas/Operación del Producto.
- Su capacidad para soportar los cambios/Revisión del Producto.
- Su adaptabilidad a nuevos entornos/Transición del producto.

Figura 2.16.: Modelo de calidad de Software McCALL.



Fuente: Roger Pressman, 2005.

2.8.1.1 Operación del Producto

- **Corrección:** Mide el grado en que un programa satisface sus especificaciones y consigue los objetivos del usuario.

- **Fiabilidad:** Mide el grado en que se puede esperar que un programa lleve a cabo sus funciones esperada con la precisión requerida.
- **Eficiencia:** Mide la cantidad de recursos de computadora y de código requerido por un programa para que lleve a cabo las funciones especificadas.
- **Integridad:** Es el grado en que puede controlarse el acceso al software o a los datos por personal no autorizado.
- **Facilidad de Uso:** Es el esfuerzo requerido para aprender un programa e interpretar la información de entrada y de salida.

2.8.1.2 Revisión del Producto

- **Facilidad de Mantenimiento:** Es el esfuerzo requerido para localizar y arreglar programas.
- **Facilidad de Prueba:** Es el esfuerzo requerido para probar un programa.
- **Flexibilidad:** Es el esfuerzo requerido para modificar un sistema operativo.

2.8.1.3 Transición del Producto

- **Portabilidad:** Es el esfuerzo requerido para transferir un software de un hardware o un entorno de sistemas a otro.
- **Reusabilidad:** Es el grado en que un programa (o partes de un programa) se puede reutilizar en otro.
- **Facilidad de Interoperación:** Es el esfuerzo requerido para asociar un programa a otro.

2.9 ESTIMACIÓN DE COSTOS DEL PROYECTO

2.9.1 MODELO DE COSTO COCOMO II

COCOMO II está compuesto por tres modelos denominados: Composición de Aplicación, Diseño Temprano y Post-Arquitectura.

Éstos surgen en respuesta a la diversidad del mercado actual y futuro de desarrollo de software. Esta diversidad podría representarse con el siguiente esquema.

Tabla 2.1: Distribución del Mercado de Software Actual y Futuro.

Aplicaciones desarrolladas por usuarios finales		
Generadores de Aplicaciones	Aplicaciones con Componentes	Sistemas Integrados
Infraestructura		

Fuente: [Boehm 1995/1].

- **Aplicaciones desarrolladas por Usuarios Finales:** En este sector se encuentran las aplicaciones de procesamiento de información generadas directamente por usuarios finales, mediante la utilización de generadores de aplicaciones tales como planillas de cálculo, sistemas de consultas, etc. Estas aplicaciones surgen debido al uso masivo de estas herramientas, conjuntamente con la presión actual para obtener soluciones rápidas y flexibles.
- **Generadores de Aplicaciones:** En este sector operan firmas como Lotus, Microsoft, Novell, Borland con el objetivo de crear módulos pre-empaquetados que serán usados por usuarios finales y programadores.
- **Aplicaciones con Componentes:** Sector en el que se encuentran aquellas aplicaciones que son específicas para ser resueltas por soluciones pre-empaquetadas, pero son lo suficientemente simples para ser construidas a partir de componentes interoperables. Componentes típicas son constructores de interfaces gráficas, administradores de bases de datos, buscadores inteligentes de datos, componentes de dominio-específico (medicina, finanzas, procesos industriales, etc.). Estas aplicaciones son generadas por un equipo reducido de personas, en pocas semanas o meses.
- **Sistemas Integrados:** Sistemas de gran escala, con un alto grado de integración entre sus componentes, sin antecedentes en el mercado que se puedan tomar como base. Porciones de estos sistemas pueden ser desarrolladas a través de la composición de aplicaciones. Entre las empresas que desarrollan software representativo de este sector, se encuentran grandes

firmas que desarrollan software de telecomunicaciones, sistemas de información corporativos, sistemas de control de fabricación, etc.

- **Infraestructura:** Área que comprende el desarrollo de sistemas operativos, protocolos de redes, sistemas administradores de bases de datos, etc. Incrementalmente este sector direccionará sus soluciones, hacia problemas genéricos de procesamiento distribuido y procesamiento de transacciones, a solución.

2.9.2 CARACTERÍSTICAS PRINCIPALES

Los objetivos principales que se tuvieron en cuenta para construir el modelo COCOMO II fueron:

- Desarrollar un modelo de estimación de costo y cronograma de proyectos de software que se adaptara tanto a las prácticas de desarrollo de la década del 90 como a las futuras.
- Construir una base de datos de proyectos de software que permitiera la calibración continua del modelo, y así incrementar la precisión en la estimación.
- Implementar una herramienta de software que soportara el modelo.
- Proveer un marco analítico cuantitativo y un conjunto de herramientas y técnicas que evaluaran el impacto de las mejoras tecnológicas de software sobre los costos y tiempos en las diferentes etapas del ciclo de vida de desarrollo.

COCOMO II está compuesto por tres modelos denominados: Composición de Aplicación, Diseño Temprano y Post-Arquitectura, Los tres modelos se adaptan tanto a las necesidades de los diferentes sectores descriptos, como al tipo y cantidad de información disponible en cada etapa del ciclo de vida de desarrollo, lo que se conoce por granularidad de la información.

Modelos de estimación

El esfuerzo necesario para concretar un proyecto de desarrollo de software, cualquiera sea el modelo empleado, se expresa en meses/persona (PM) y representa los meses de trabajo de una persona full time, requeridos para desarrollar el proyecto.

Modelo Composición de Aplicación

La fórmula propuesta en este modelo es la siguiente:

$$\mathbf{PM = NOP / PROD}$$

Dónde:

NOP (Nuevos Puntos Objeto): Tamaño del nuevo software a desarrollar expresado en Puntos Objeto y se calcula de la siguiente manera:

$$\mathbf{NOP = OP \times (100 - \%reuso) / 100}$$

OP (Punto Objeto): Tamaño del software a desarrollar expresado en Puntos Objeto.

%reuso: Porcentaje de reuso que se espera lograr en el proyecto.

PROD: Es la productividad promedio determinada a partir del análisis de datos de proyectos.

Tabla 2.2.: Productividad para el modelo Composición de Aplicación.

Experiencia y capacidad de los desarrolladores	Muy Bajo	Bajo	Normal	Alto	Muy Alto
Madurez y capacidad del ICASE	Muy Bajo	Bajo	Normal	Alto	Muy Alto
PROD	4	7	1	25	50

Fuente: [Boehm 1995/2].

Modelo diseño temprano

Este modelo se usa en las etapas tempranas de un proyecto de software, cuando se conoce muy poco del tamaño del producto a ser desarrollado, de la naturaleza de la plataforma, del personal a ser incorporado al proyecto o detalles específicos del proceso a utilizar. Este modelo podría emplearse tanto en productos desarrollados en sectores de Generadores de Aplicación, Sistemas Integrados o Infraestructura. El

modelo de Diseño Temprano ajusta el esfuerzo nominal usando siete factores de costo. La fórmula para el cálculo del esfuerzo es la siguiente:

$$PM_{estimado} = PM_{nominal} \times \prod_{i=1}^7 EM_i$$

$$PM_{nominal} = A * (KSLOC)^B$$

$$B = 1.01 + 0.01 * \sum_{j=1}^5 W_j$$

Donde:

PMEstimado es el esfuerzo Nominal ajustado por 7 factores, que reflejan otros aspectos propios del proyecto que afectan al esfuerzo necesario para la ejecución del mismo.

- KSLOC es el tamaño del software a desarrollar expresado en miles de líneas de código fuente.
- A es una constante que captura los efectos lineales sobre el esfuerzo de acuerdo a la variación del tamaño, ($A=2.94$).
- B es el factor exponencial de escala, toma en cuenta las características relacionadas con las economías y des economías de escala producidas cuando un proyecto de software incrementa su tamaño.
- EM_i corresponde a los factores de costo que tienen un efecto multiplicativo sobre el esfuerzo, llamados Multiplicadores de Esfuerzo (Effort Multipliers). Cada factor se puede clasificar en seis niveles diferentes que expresan el impacto del multiplicador sobre el esfuerzo de desarrollo. Esta escala varía desde un nivel Extra Bajo hasta un nivel Extra Alto. Cada nivel tiene un peso asociado. El peso promedio o nominal es 1.0. Si el factor provoca un efecto nocivo en el esfuerzo de un proyecto, el valor del multiplicador correspondiente será mayor que 1.0, caso contrario el multiplicador será inferior a 1.0. La Figura 4 muestra una pantalla del software COCOMO II.1999.0, donde se aprecian los valores de los factores de acuerdo a cada nivel, según la calibración efectuada para el año 1999.

Clasificados en categorías, los 7 Multiplicadores de Esfuerzo son:

Del Producto

RCPX: Confiabilidad y Complejidad del producto

RUSE: Reusabilidad Requerida

De la Plataforma

PDIF: Dificultad de la Plataforma

Del Personal

PERS: Aptitud del Personal

PREX: Experiencia del Personal

Del Proyecto

FCIL: Facilidades

SCED: Cronograma de Desarrollo Requerido

Modelo de Post-Arquitectura

Es el modelo de estimación más detallado y se aplica cuando la arquitectura del proyecto está completamente definida. Este modelo se aplica durante el desarrollo y mantenimiento de productos de software incluidos en las áreas de Sistemas Integrados, Infraestructura y Generadores de Aplicaciones. (Adriana Gómez, María del C. López, Silvina Migani, Alejandra Otazú).

El esfuerzo nominal se ajusta usando 17 factores multiplicadores de esfuerzo. El mayor número de multiplicadores permite analizar con más exactitud el conocimiento disponible en las últimas etapas de desarrollo, ajustando el modelo de tal forma que refleje fielmente el producto de software bajo desarrollo. La fórmula para el cálculo del esfuerzo es la siguiente:

$$PM_{estimado} = PM_{nominal} \times \prod_{i=1}^{17} EM_i$$

El modelo Post-Arquitectura es el que se adapta más al proyecto en desarrollo por ello se especifica los parámetros necesarios para su cálculo.

Un Punto de Función, es definida como una función comercial del usuario final para calcular el valor se utiliza la siguiente ecuación:

$$PFA = PF * (0.65 + 0.01 * \sum Fi)$$

Donde:

PFA, es el Punto de Función Ajustado. PF, Puntos de Función sin Ajustar. 0.65 y 0.01 son constantes.

$\sum Fi$, es el Factor de Complejidad

El PF se calcula con la siguiente tabla dependiendo de la cantidad que cada uno de los parámetros tengan:

Tabla 2.3.: Ponderación para Puntos de Función.

PARAM. MEDICIÓN	FACTOR DE PONDERACIÓN		
Entradas de Usuarios	3	4	6
Salidas de Usuarios	4	5	7
Peticiones de Usuarios	3	4	6
Archivos	7	10	15
Interfaces Externas	5	7	10

Fuente: (Garcia, 2010).

Se multiplica la cantidad de parámetros que tiene por el factor de ponderación que le corresponda.

La sumatoria del Factor de Complejidad se calcula según los resultados a interrogantes de la siguiente tabla, los valores se deben expresar en la escala de cero a cinco.

Tabla 2.4.: Factores de Complejidad.

Nro.	Factor de Complejidad	Valor
1	Requiere copias de seguridad	0 – 5
2	Necesita comunicación de datos	0 – 5
3	Existe funciones de procedimientos distribuidos	0 – 5
4	Rendimiento critico	0 – 5
5	Se ejecuta en un entorno operativo existente	0 – 5
6	Se requiere entrada de datos en línea	0 – 5
7	Transacciones de entradas en múltiples pantallas	0 – 5
8	Archivos maestros actualizados en línea	0 – 5
9	Complejidad de valores del dominio de la información	0 – 5
10	Complejidad del procedimiento interno	0 – 5
11	Código diseñado para reutilización	0 – 5
12	Conversión/instalación de diseño	0 – 5
13	Instalaciones múltiples	0 – 5
14	Aplicación diseñada para el cambio	0 – 5

Fuente: (Maquera, 2017).

Las líneas de código se calculan con la siguiente fórmula:

$$\text{LDC} = \text{PFA} * \text{Factor LDC/PF}$$

Donde:

LDC: Líneas de Código.

PFA: Puntos de Función Ajustados.

Factor LDC/PF: Dependiendo del Lenguaje de programación se obtiene el valor.

Los miles de Líneas de Código se calculan de la con la ecuación que sigue:

$$\text{KLDC} = \text{LDC}/1000$$

Donde:

KLDC: Miles de líneas de código.

LDC: Líneas de código

$$\text{PMNominal} = \text{A} * (\text{KLDC})^{\text{B}}$$

Dónde:

PMnominal: Esfuerzo nominal en Personas/Mes.

A: Constante derivada de la calibración igual a 2.94.

B: Es el factor exponencial.

El cálculo de B

$$B = 0.91 + 0.01 * \sum_{j=1}^5 w_j$$

Dónde:

$$\sum_{j=1}^5 w_j$$
 es la sumatoria de factores de escala

Los factores de Escala en la siguiente tabla se explican en la siguiente tabla:

Tabla 2.5: Factores de Escala.

Factor de Escala	Valor
PREC (Procedencia)	Experiencia en aplicaciones del mismo tipo
FLEX (Flexibilidad de desarrollo)	Grado de sujeción del desarrollo a tiempo y requisitos
RESL (Resolución de arquitectura)	Identificación de riesgos en la aplicación
TEAM (Cohesión de equipo)	Nivel de integración del equipo de desarrollo.
PMAT (Madurez del Proceso)	Experiencia en el modelo de desarrollo

Fuente: (Maquera, 2017).

En la siguiente tabla se describe los valores que pueden tomar los Factores de Escala.

Tabla 2.6.: Valores para los Factores de Escala.

Factor de Escala	Muy bajo	Bajo	Nominal	Alto	Muy alto	Extra alto
PREC	6.2	4.96	3.72	2.48	1.24	0
FLEX	5.07	4.05	3.04	2.03	1.01	0
RESL	7.07	5.65	4.24	2.83	1.41	0
TEAM	5.48	4.38	3.29	2.19	1.1	0
PMAT	7.8	6.29	4.68	3.12	1.56	0

Fuente: (Maquera, 2017).

El esfuerzo estimado se calcula con la siguiente fórmula:

$$PM_{estimado} = PM_{nominal} \times \prod_{i=1}^{17} EM_i$$

$\prod_{i=1}^{17} EM_i$ es el multiplicador de esfuerzo que se calcula según la siguiente tabla.

Tabla 2.7.: Multiplicadores de Esfuerzo.

Conductores de Costo	Nro.	Sigla	Descripción
PRODUCTO	1	RELY	Seguridad requerida
	2	DATA	Tamaño de la Base de Datos
	3	CPLX	Complejidad
	4	RUSE	Reutilización requerida
	5	DOCU	Documentación adaptada al ciclo de vida
PLATAFORMA	6	TIME	Tiempo de ejecución requerido
	7	STOR	Almacenamiento principal requerido
	8	PVOL	Volatilidad de la plataforma
PERSONAL	9	ACAP	Capacidad de análisis
	10	PCAP	Capacidad del programador
	11	PCON	Continuidad del personal
	12	AEXP	Experiencia del analista
	13	PEXP	Experiencia en la plataforma del sistema operativo
	14	LTEX	Experiencia en el lenguaje y herramientas
PROYECTO	15	TOOL	Uso de herramientas de software
	16	SITE	Desarrollo multitarea
	17	SCED	Esquemas de desarrollo programado

Fuente: (Maquera, 2017).

A continuación, se muestra la tabla de multiplicadores de esfuerzo con los posibles valores que podría seleccionarse.

Tabla 2.8: Valores para Multiplicadores de Esfuerzo.

Conductores de Costo	Nro.	Multiplicadores de esfuerzo	Muy bajo	Bajo	Nominal	alto	Muy alto	Extra alto
PRODUCTO	1	RELY	0.82	0.92	1	1.1	1.26	-
	2	DATA	-	0.9	1	1.1	1.28	-
	3	CPLX	0.73	0.87	1	1.2	1.34	-
	4	RUSE	-	0.95	1	1.1	1.15	-
	5	DOCU	0.81	0.91	1	1.1	1.23	-
PLATAFORMA	6	TIME	-	-	1	1.1	1.29	-
	7	STOR	-	-	1	1.1	1.17	-
	8	PVOL	-	0.87	1	1.2	1.3	-
PERSONAL	9	ACAP	1.42	1.19	1	0.9	0.71	-
	10	PCAP	1.34	1.15	1	0.9	0.76	-
	11	PCON	1.29	1.12	1	0.9	0.81	-
	12	AEXP	1.22	1.1	1	0.9	0.81	-
	13	PEXP	1.19	1.09	1	0.9	0.85	-
	14	LTEX	1.2	1.09	1	0.9	0.84	-
PROYECTO	15	TOOL	1.17		1	0.9	0.78	-
	16	SITE	1.22		1	0.9	0.86	-
	17	SCED	1.43		1	1	1	-

Fuente: (Maquera, 2017).

2.10 SEGURIDAD DE SOFTWARE

El tema de seguridad de la información es ampliamente discutido y evidenciado en el medio corporativo, ya que la información se considera uno de los bienes más valiosos para las organizaciones, independientemente de su segmento o porte. La importancia de la información amplió la necesidad del desarrollo de estructura estandarizada para implantación y operación de los conceptos de seguridad de la información. En el marco

de esta demanda, organizaciones mundiales (ISO/IEC) iniciaron el desarrollo de normas, originando la familia ISO 27000, que estandariza actividades relacionadas con la implantación y operación de Sistemas de Gestión de Seguridad de la Información (SGSI).

2.10.1 LA TRIADA DE LA SEGURIDAD DE LA INFORMACIÓN

Basado en los documentos de la familia ISO 27000, podemos resaltar parámetros de seguridad que son fundamentales en el desarrollo de implementación o incluso, considerar estos parámetros en el desarrollo del proyecto, entre estos parámetros encontramos los tres más principales, pero cabe mencionar que existen aspectos que se complementan a estos.

Figura 2.17.: Triada de la Información.



Fuente: Recuperado de “La triada CIA”, (Cambio Digital, 2020).

- a) **Confidencialidad:** Desde las grandes empresas hasta las personas, tienen información y contenidos que no desean compartir con cualquiera. Es necesario que los sistemas de seguridad rechacen a quienes no deberían ver estos contenidos. Es la clave de la confidencialidad. Por ejemplo, hay una ley federal que indica que la información de los estudiantes en las universidades debe estar restringida. Así que solo el personal autorizado debería poder acceder a los registros de calificaciones.

- b) Integridad:** Es la garantía de que la información a la que se accede no se ha alterado y que lo que allí se lee es exactamente lo que se pretende. Lo que la integridad nos sugiere es que la información a la que accedemos es verdaderamente confiable, que podemos aceptarla como un hecho. Pero sabemos que la información puede perder su integridad con la intervención maliciosa. Por ejemplo, cuando un pirata informático ingresa al sistema de una universidad y cambia las calificaciones a conveniencia. Sin embargo, la integridad también se puede perder de manera involuntaria. Un ejemplo de esto es cuando alguien autorizado por error borra o cambia algún archivo de la información, o cuando por una falla de energía, la computadora corrompe un archivo, y así sucesivamente.
- c) Disponibilidad:** Esta tercera parte de la triada de la CIA se refiere a la posibilidad de que alguien autorizado pueda acceder sin problema a la información y si es necesario, modificarla. Esto, durante un plazo de tiempo adecuado, que se define de acuerdo al tipo de trabajo. Un operador de acciones debe recibir la información de inmediato, mientras que un vendedor de inmuebles puede esperar al día siguiente para ver cómo cerraron las ventas del día. En términos de disponibilidad, el gigante Amazon es un excelente ejemplo, ya que debe estar disponible los 365 días del año sin fallas. Algunos sitios web de empresas, sin embargo, pueden permitirse entrar en mantenimiento por unos días o algo por el estilo.

2.11 PRUEBAS DEL SISTEMA

Un sistema de pruebas implica la operación o aplicación del mismo a trances de condiciones controladas y la consiguiente evaluación de la información. Las condiciones controladas deben incluir tanto situaciones normales como anormales. El objetivo del sistema de pruebas es encontrar un error para determinar situaciones en donde algo pasa cuando no debe de pasar y viceversa. En una palabra, un sistema de pruebas está orientado a detectar. Para la planeación de las pruebas que se van a aplicar al sistema evaluador se integraron los distintos tipos de pruebas que se explicaran a continuación:

- **Prueba de Caja Blanca.** Al contrario de las pruebas de caja negra, estas se basan en el conocimiento de la lógica interna del código del sistema. Las pruebas contemplan los distintos caminos que se pueden generar gracias a las estructuras condicionales, a los distintos estados del mismo etc.
- **Prueba de Caja Negra.** En el sistema de pruebas de caja negra no considera la codificación dentro de sus parámetros a evaluar, es decir, que no están basadas en el conocimiento del diseño interno del programa. Estas pruebas se enfocan en los requerimientos establecidos y en la funcionalidad del sistema.

3 DISEÑO METODOLOGICO

3.1 DIAGNÓSTICO DEL SISTEMA ACTUAL

El administrador del Gimnasio Winner juntamente con los instructores son el personal encargado de realizar los cambios en los horarios y la alimentación necesaria o insertar nuevas rutinas, esto se realiza de acuerdo a las exigencias de los inscritos.

En caso de que los inscritos requieran nuevas rutinas de entrenamiento o una nueva alimentación para el quemado de grasa o aumento de masa muscular se realizara una actualización de los nuevos métodos de rutina y nutrición.

3.2 ANÁLISIS DEL SISTEMA

3.2.1 INGENIERÍA DE REQUERIMIENTOS.

Recolección y descripción de la información

Entrevista.

Se realizó una entrevista al Administrador del Gimnasio Winner quien expresa la necesidad de un Sistema Web para una mejor administración, de esa manera los usuarios o inscritos podrán acceder al sistema desde cualquier sitio web donde podrá visualizar las rutinas, alimentación, horarios, así como los horarios y los instructores disponibles.

3.2.2 REQUERIMIENTOS FUNCIONALES

En base al pedido del Gimnasio Winner, en la entrevista se formula la siguiente tabla en la que muestra los requisitos funcionales que el sistema debe satisfacer tomando en cuenta el Sistema Web en la que se ejecuta.

Tabla 3.1.: Requerimientos Funcionales.

Nº	REQUERIMIENTOS FUNCIONALES	CATEGORIA
RF01	El administrador del sistema Web podrá adicionar, editar a un usuario.	Evidente
RF02	Los usuarios o clientes deben tener un usuario y contraseña para poder ingresar al Sistema.	Evidente
RF03	Los usuarios o clientes podrán ver las rutinas en el sistema y así poder realizar los ejercicios de manera eficaz.	Evidente
RF04	Los clientes o usuario podrán ver los horarios donde se mostrara en el sitio web.	Evidente
RF05	Los clientes o usuarios podrán mandar sus consultas o preguntas por mediante del sitio web.	Evidente
RF06	Todos los clientes o usuarios inscritos al gimnasio tendrán acceso a la información de rutinas horarios nutrición y videos que se publicara en el sistema web.	Evidente
RF07	Los nuevos clientes o usuarios podrán realizar su reserva mediante el Sitio Web.	Evidente

Fuente: Elaboración Propia.

3.2.3 REQUERIMIENTO NO FUNCIONALES

Tabla 3.2.: Requerimientos no funcionales.

Nº	REQUERIMIENTOS NO FUNCIONALES
RNF01	El Sistema Web debe funcionar correctamente en los navegadores Google Chrome, Mozilla Firefox y Opera .
RNF02	La interfaz gráfica debe ser amigable para los clientes o usuaria que visitan el sitio web.
RNF03	La interfaz gráfica debe ser amigable para los clientes o usuaria que visitan el sitio web.

Fuente: Elaboración Propia.

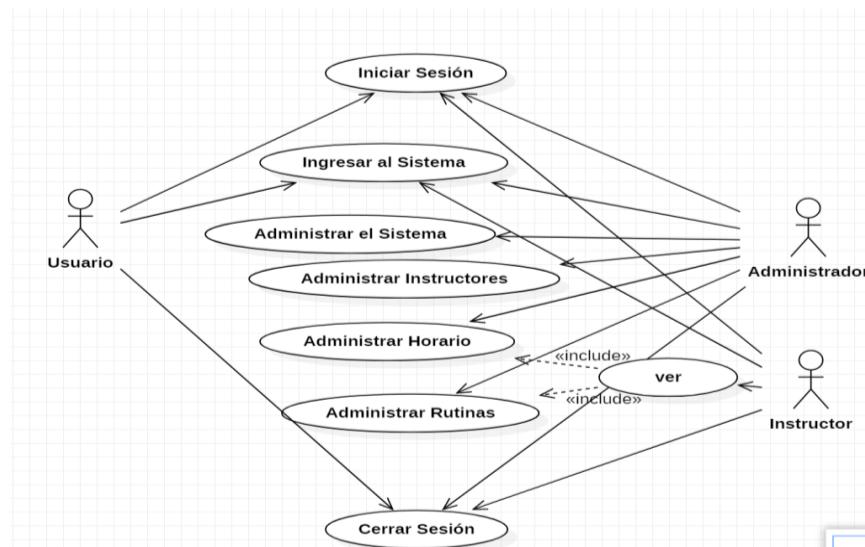
3.2.4 MODELO DEL NEGOCIO

Ejecutamos un estudio preliminar del Gimnasio Winner para lograr identificar y describir el proceso del gimnasio y de esa manera poder comprender la actividad de la organización con respecto al problema planteado, con el fin de dar solución a los requerimientos de los usuarios.

Diagrama de Caso de Uso

Se muestra el Diagrama de Casos de Uso en la siguiente imagen.

Figura 3.1.: Diagrama de Casos de Uso del Gimnasio Winner.



Fuente: Elaboración Propia.

Tabla 3.3.: Descripción de Actores de Casos de Uso del Gimnasio Winner.

ACTORES	DESCRIPCION
Usuarios/Clientes	Podrán ingresar a nuestro sistema como podrán tener el acceso a los horarios y las rutinas.
Administrador	Es el encargado de administrar el Sistema ya que puede adicionar, editar y eliminar a los instructores como a también a los usuarios inscritos. También podrá realizar cambios en el horario y las rutinas de se llevan. También podrá adicionar nuevos módulos.
Instructores	Podrán ingresar al sistema y ver los horarios y las rutinas al mismo tiempo adicionar, cambiar, eliminar las rutinas y como también la alimentación.

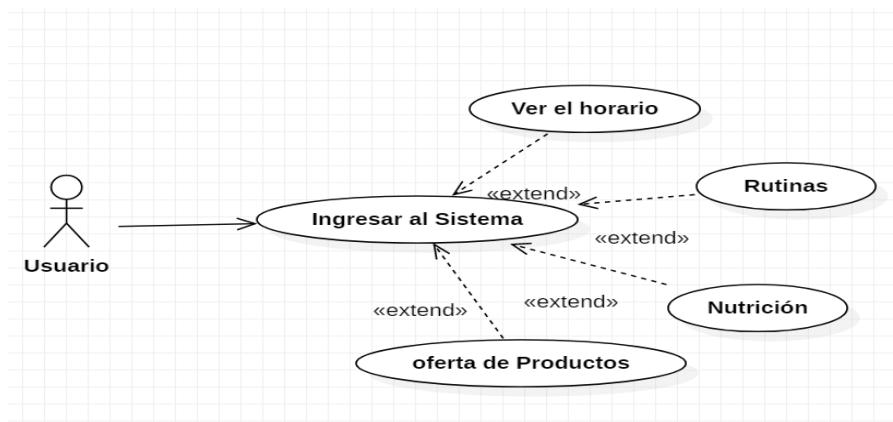
Fuente: Elaboración Propia.

Detallado de Caso de Uso

Caso de Uso: Ingresar al Sistema (Usuario).

El usuario deberá de estar registrado en el sistema antes de ingresar al sistema ya que esto ayudará al usuario o inscrito pueda ver los horarios, rutinas, nutrición y la oferta de productos.

Figura 3.2.: Caso de Uso, Ingresar al Sistema (Usuario).



Fuente: Elaboración Propia.

Tabla 3.4.: Caso de Uso Ingresar al Sistema (Usuario).

CASO DE USO	
Nombre	Ingresar al sistema por parte del Usuario
Autor	Estanislao Yahuasi Apaza
Descripción	Permite que el Usuario tenga acceso al Sistema donde podrán interactuar con los servicios que ofrece el Gimnasio Winner.
Actores	Usuario.
Precondiciones	Los Usuarios deben tener acceso a internet

Fuente: Elaboración Propia.

Tabla 3.5.: Flujo de Eventos: Caso de Uso Ingresar al Sistema (Usuario),

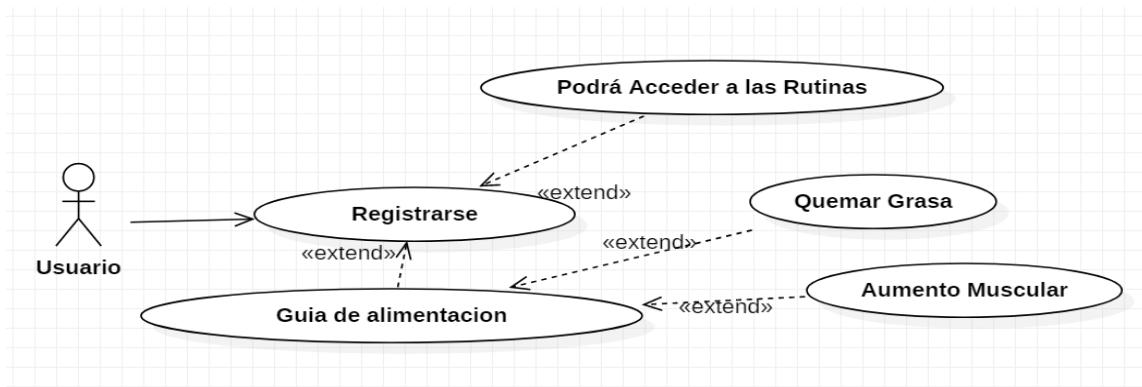
FLUJO NORMAL	FLUJO ALTERNATIVO
1. Ingresar al Sistema	<i>En caso de que la cobertura de internet no llegase se mostrara una un mensaje de error</i>
2. Interactuar con el sistema	

Fuente: Elaboración Propia.

Caso de Uso: Registrarse (Usuario).

La administración, usuario e instructor deben registrarse para poder ingresar al sistema y acceder a las rutinas, como también ver la guía de alimentación para la quema de grasa o aumento de masa muscular.

Figura 3.3.: Caso de Uso Registrarse (Usuario).



Fuente: Elaboración Propia.

Tabla 3.6.: Caso de Uso Registrarse (Usuario).

CASO DE USO	
Nombre	Registrarse por parte del Usuario.
Autor	Estanislao Yahuasi Apaza
Descripción	Permite que el Usuario pueda registrarse desde la plataforma del sistema web para evitar la pérdida de tiempo.
Actores	Usuario.
Precondiciones	Los Usuarios deben tener acceso a internet.

Fuente: Elaboración Propia.

Tabla 3.7.: Flujo de Eventos: caso de Uso Registrarse por parte del Usuario.

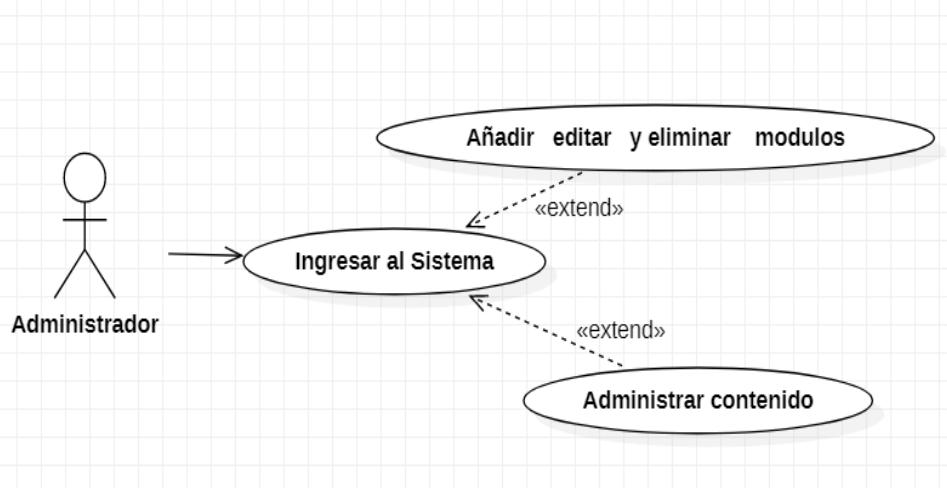
FLUJO NORMAL	FLUJO ALTERNATIVO
1. Registrarse	<i>En caso de que la cobertura de internet no llegase se mostrara una un mensaje de error</i>
2. Inscribirse	

Fuente: Elaboración Propia.

Caso de Uso: Ingresar al Sistema (Administrador).

El usuario o inscrito deberá registrarse o inscribirse para poder acceder al sistema.

Figura 3.4.: Caso de Uso Ingresar al Sistema (Administrador).



Fuente: Elaboración Propia.

Tabla 3.8.: Caso de Uso Ingresar al Sistema (Administrador).

CASO DE USO	
Nombre	Ingresar al sistema por parte del Administrador
Autor	Estanislao Yahuasi Apaza
Descripción	Permite que el Administrador pueda registrarse desde la plataforma del sistema web para evitar la pérdida de tiempo.
Actores	Administrador
Precondiciones	El Administrador debe tener acceso a internet

Fuente: Elaboración Propia.

Tabla 3.9.: Flujo de Eventos: caso de Uso Ingresar al Sistema

(Administrador).

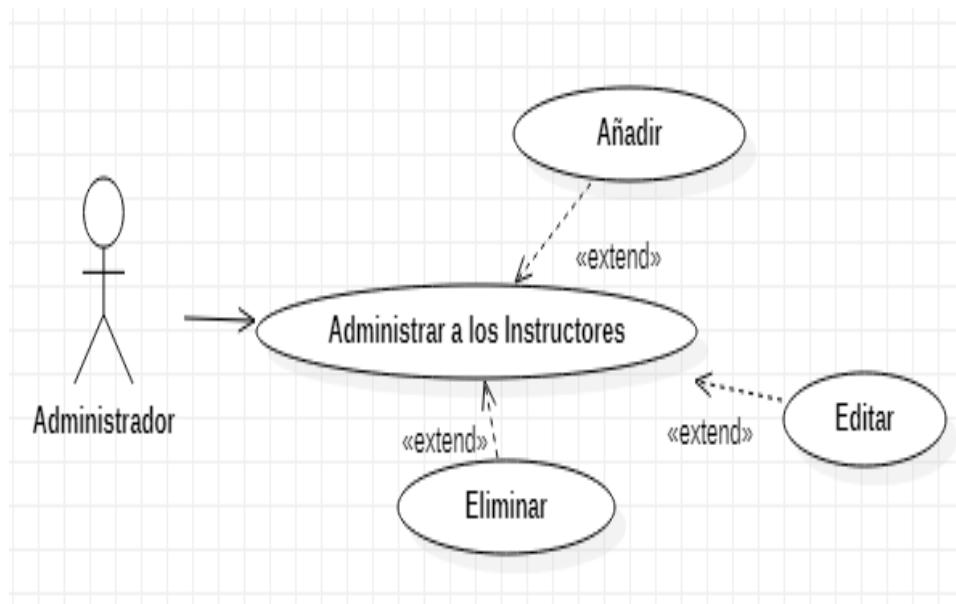
FLUJO NORMAL	FLUJO ALTERNATIVO
<ol style="list-style-type: none"> 1. <i>Ingresar al Sistema</i> 2. <i>Interactuar con el sistema</i> 	<i>En caso de que la cobertura de internet no llegase se mostrara una un mensaje de error.</i>

Fuente: Elaboración Propia.

Caso de Uso: Administrar Instructores.

El Administrador puede añadir editar o eliminar a los instructores de acuerdo al trabajo necesario de cada instructor.

Figura 3.5.: Caso de Uso Administrar Instructores.



Fuente: Elaboración Propia.

Tabla 3.10.: Caso de Uso Administrar Instructores.

CASO DE USO	
Nombre	Ingresar al sistema por parte del Administrador
Autor	Estanislao Yahuasi Apaza
Descripción	Permite que el administrador pueda Añadir, editar o eliminar a los instructores desde la plataforma del sistema web para evitar la pérdida de tiempo.
Actores	Administrador
Precondiciones	El Administrador debe tener acceso a internet

Fuente: Elaboración Propia.

Tabla 3.11.: Flujo de Eventos: Administrar Instructores.

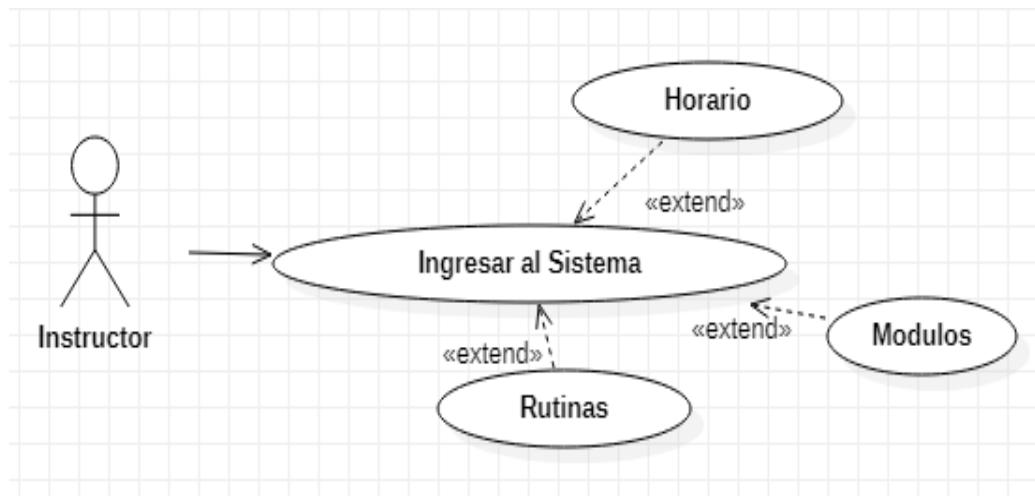
FLUJO NORMAL	FLUJO ALTERNATIVO
<ol style="list-style-type: none"> 1. Ingresar al Sistema 2. Administrar a los Instructores 	<i>En caso de que la cobertura de internet no llegase se mostrara una un mensaje de error.</i>

Fuente: Elaboración Propia.

Caso de Uso: Ingresar al Sistema (Instructor).

El instructor puede Iniciar Sesión e ingresar al sistema con sus datos personales en caso de que los datos insertados no coincidieran con la base de datos no podrá ingresar al sistema.

Figura 3.6.: Caso de Uso Ingresar al Sistema (Instructor).



Fuente: Elaboración Propia.

Tabla 3.12.: Caso de Uso Ingresar al Sistema (Instructor).

CASO DE USO	
Nombre	Ingresar al Sistema por parte del Instructor
Autor	Estanislao Yahuasi Apaza
Descripción	Permite que el instructor pueda ingresar al sistema desde la plataforma del sitio web para evitar la pérdida de tiempo.
Actores	Instructor
Precondiciones	El Instructor debe tener acceso a internet para poder ingresar al sistema.

Fuente: Elaboración Propia.

Tabla 3.13.: Flujo de Eventos: Caso de Uso Ingresar al Sistema (Instructor).

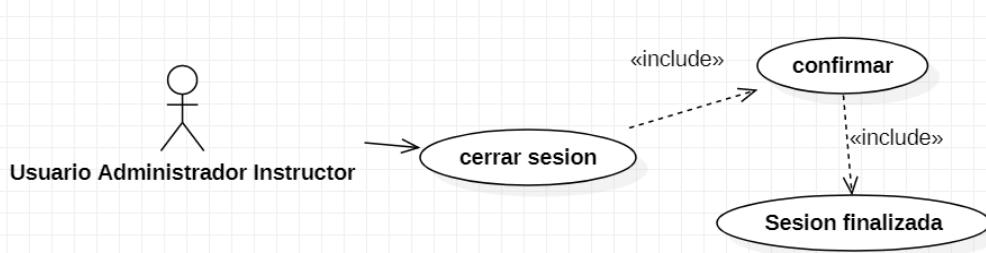
FLUJO NORMAL	FLUJO ALTERNATIVO
1. Ingresar al Sistema 2. Administrar a los Instructores	En caso de que no cuente con internet no podrá ingresar al sistema donde se mostrara una un mensaje de error.

Fuente: Elaboración Propia.

Caso de Uso: Cerrar Sesión (Usuario, administrador e instructor).

El usuario, administrador e Instructor pueden cerrar sesión y se les pedir que confirme la operación y de esa manera poder finalizar la sesión del sistema

Figura 3.7.: Caso de Uso Cerrar Sesión (Usuario, Administrador e Instructor).



Fuente: Elaboración Propia.

Tabla 3.14.: Caso de Uso Cerrar Sesión (Usuario, Administrador e Instructor).

CASO DE USO	
Nombre	Cerrar Sesión (Usuario, administrador e instructor).
Autor	Estanislao Yahuasi Apaza
Descripción	Usuario, administrador e instructor deben de confirmar para poder finalizar la sesión.
Actores	Usuario, Administrador e Instructor
Precondiciones	Usuario, administrador e instructor deben tener acceso a internet para poder ingresar al sistema.

Fuente: Elaboración Propia.

Tabla 3.15.: Flujo de Eventos: Caso de Uso Cerrar Sesión (Usuario, Administrador e Instructor).

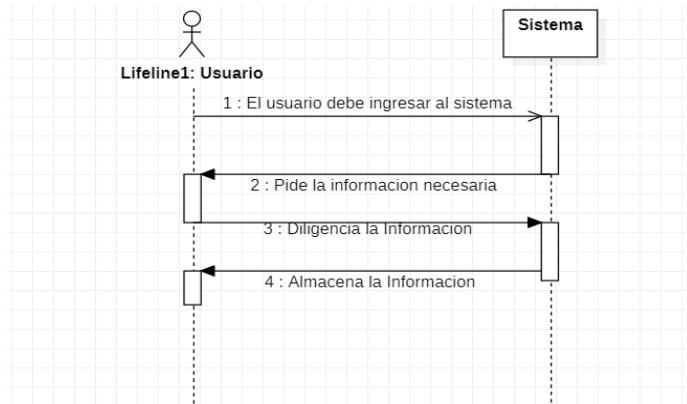
FLUJO NORMAL	FLUJO ALTERNATIVO
1. Ingresar al Sistema 2. Cerrar Sesión.	<i>En caso de que no cuente con internet no podrá ingresar al sistema donde se mostrara una un mensaje de error.</i>

Fuente: Elaboración Propia.

Diagramas de Secuencia

Ingresar al Sistema (Usuario)

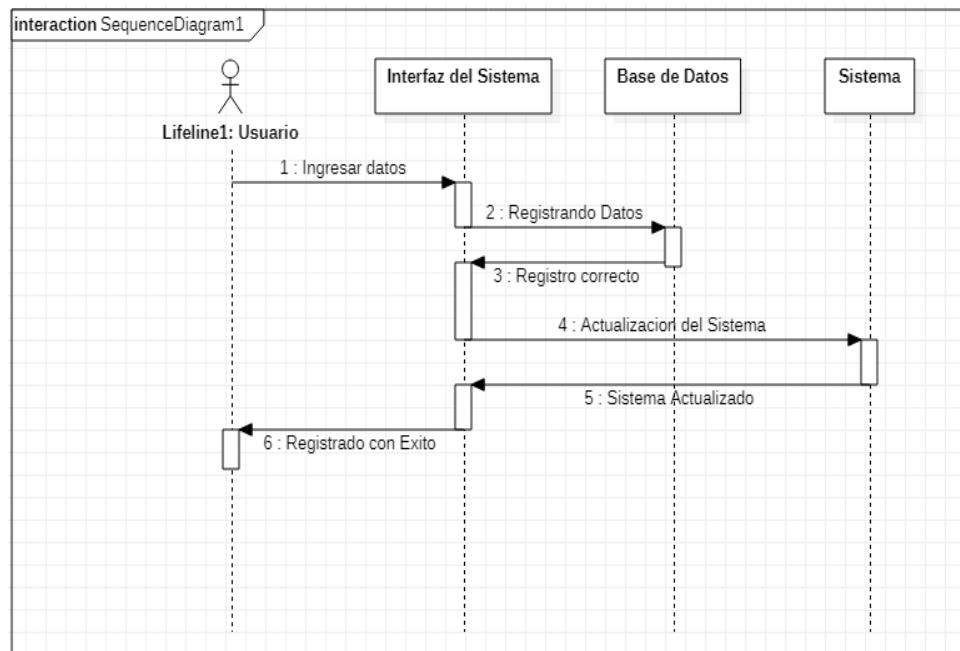
Figura 3.8.: Diagrama de Secuencia al Ingresar al Sistema (Usuario).



Fuente: Elaboración Propia.

Registrarse (Usuario)

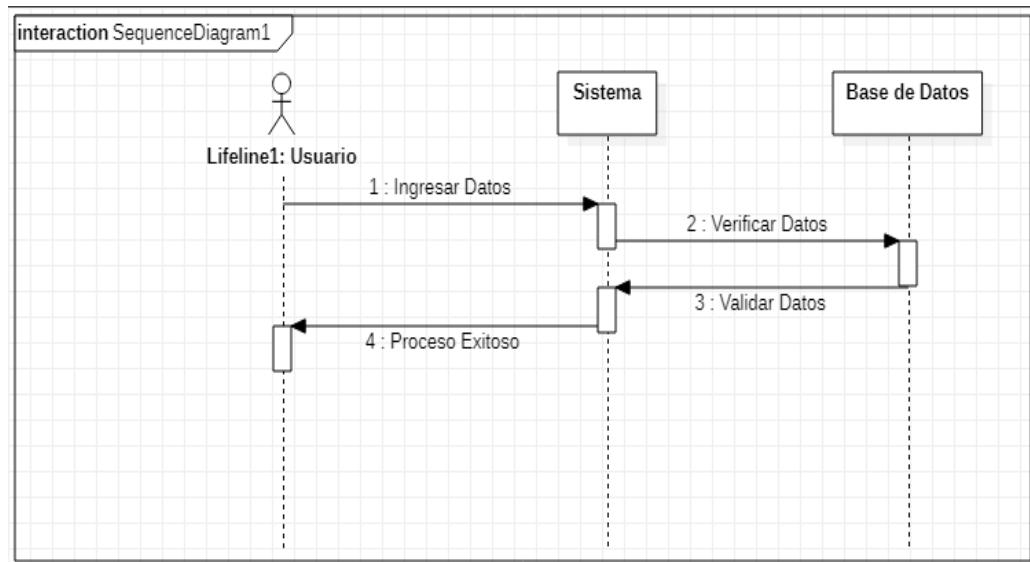
Figura 3.9.: Diagrama de Secuencia Registrarse.



Fuente: Elaboración Propia.

Ingresar al Sistema (Administrador)

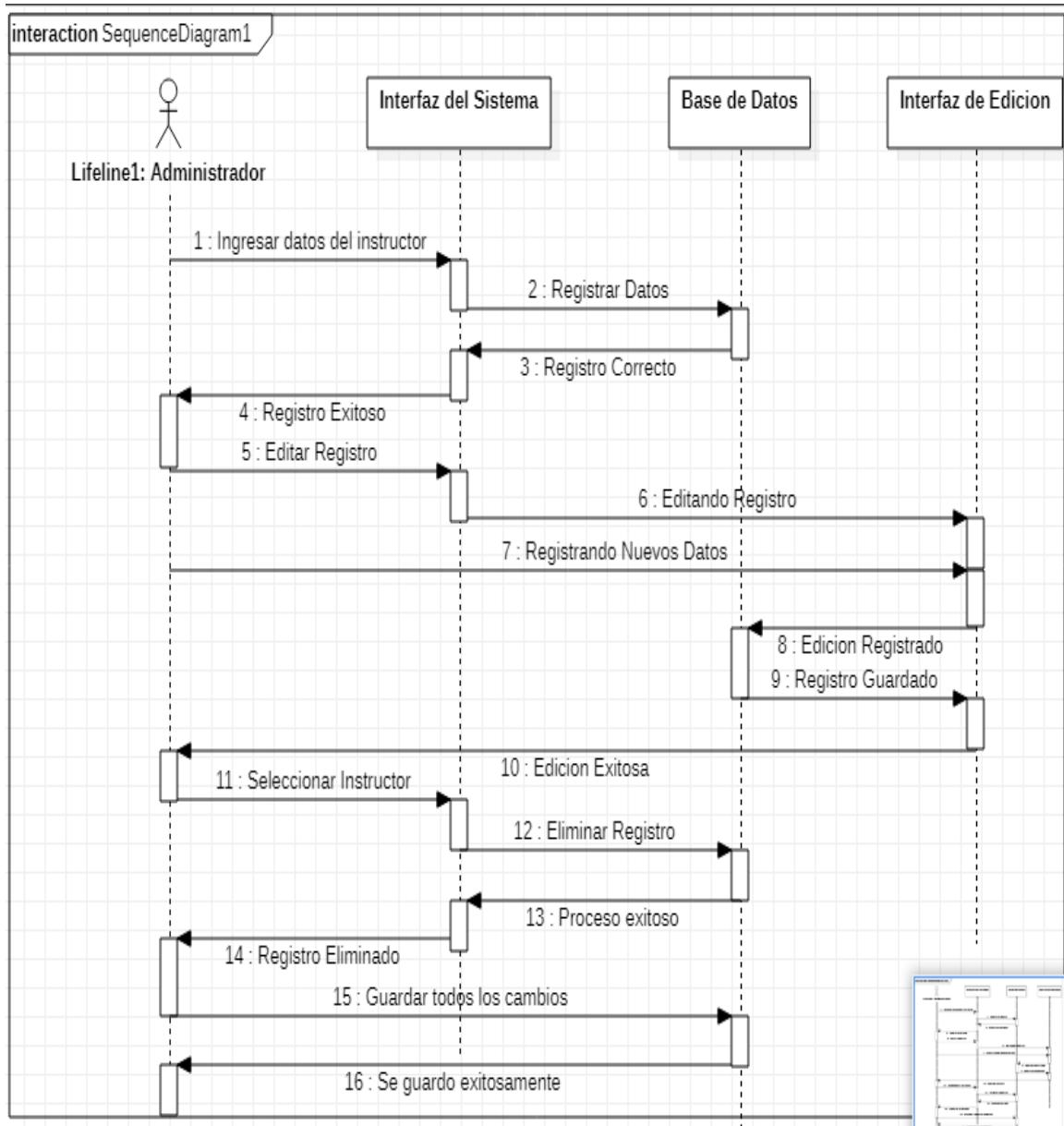
Figura 3.10.: Diagrama de Secuencia Ingresar al sistema (Administrador).



Fuente: Elaboración Propia.

Administrar Instructores

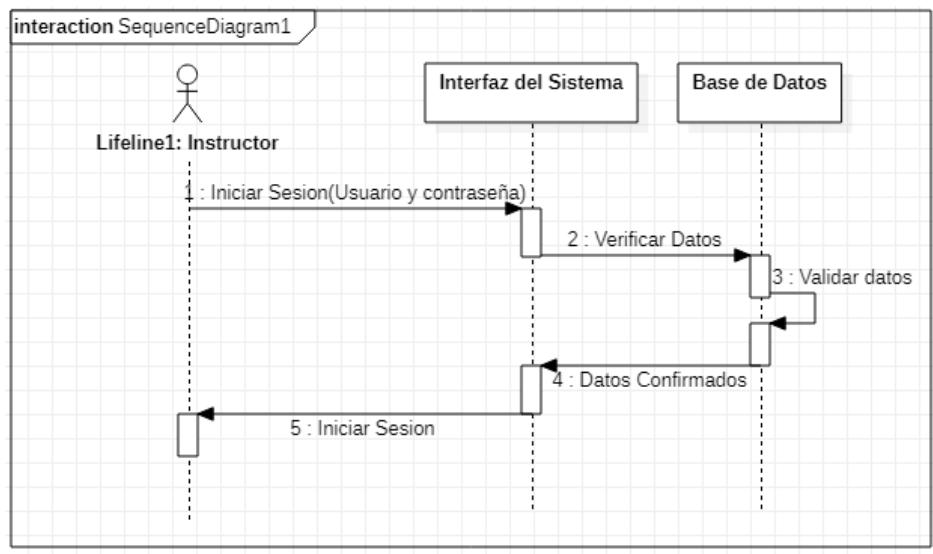
Figura 3.11.: Diagrama de Frecuencia Administrar Instructores.



Fuente: Elaboración Propia.

Ingresar al Sistema (Instructor).

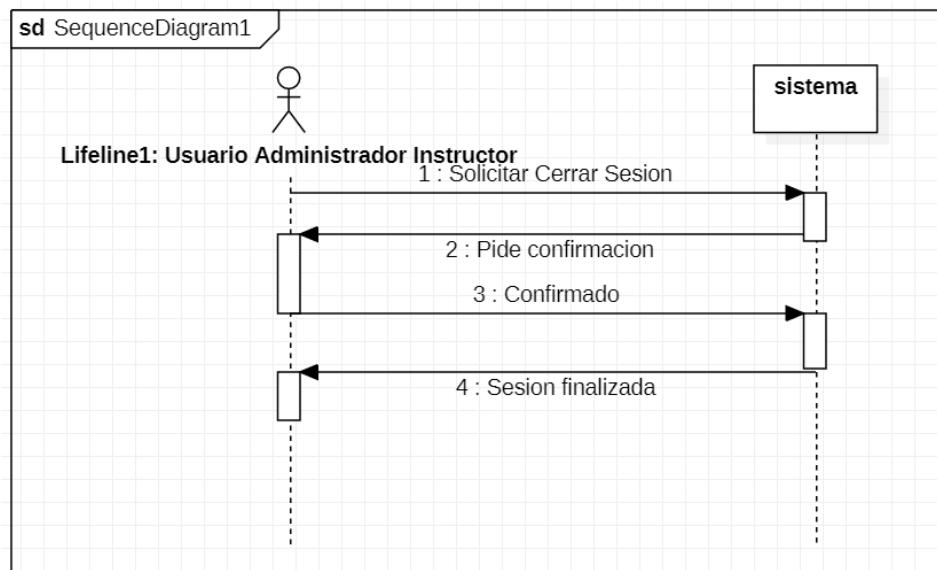
Figura 3.12.: Diagrama de Secuencia Ingresar al Sistema (Instructor).



Fuente: Elaboración Propia.

Cerrar Sesión (Usuario, administrador e instructor).

Figura 3.13.: Diagrama de Secuencia Cerrar Sesión (Usuario, Administrador e Instructor).

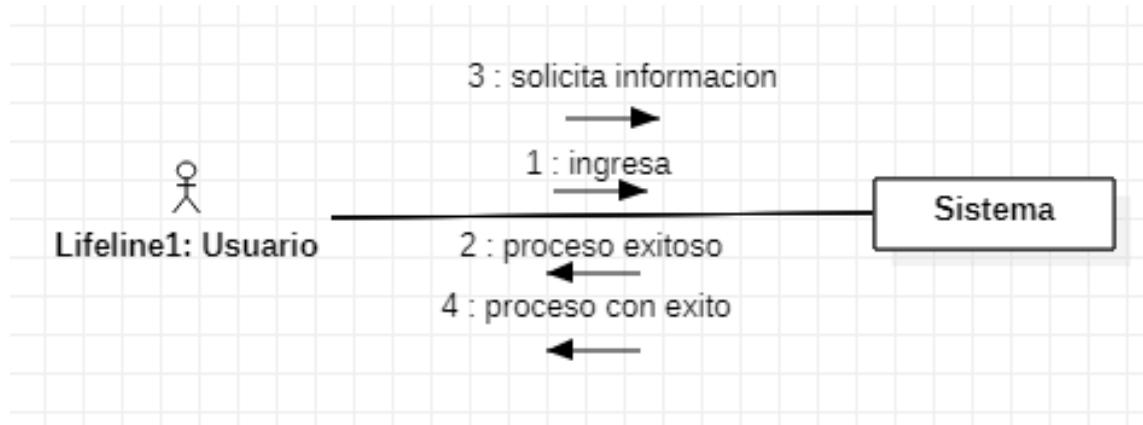


Fuente: Elaboración Propia.

Diagramas de Colaboración

Ingresar al sistema (Usuario).

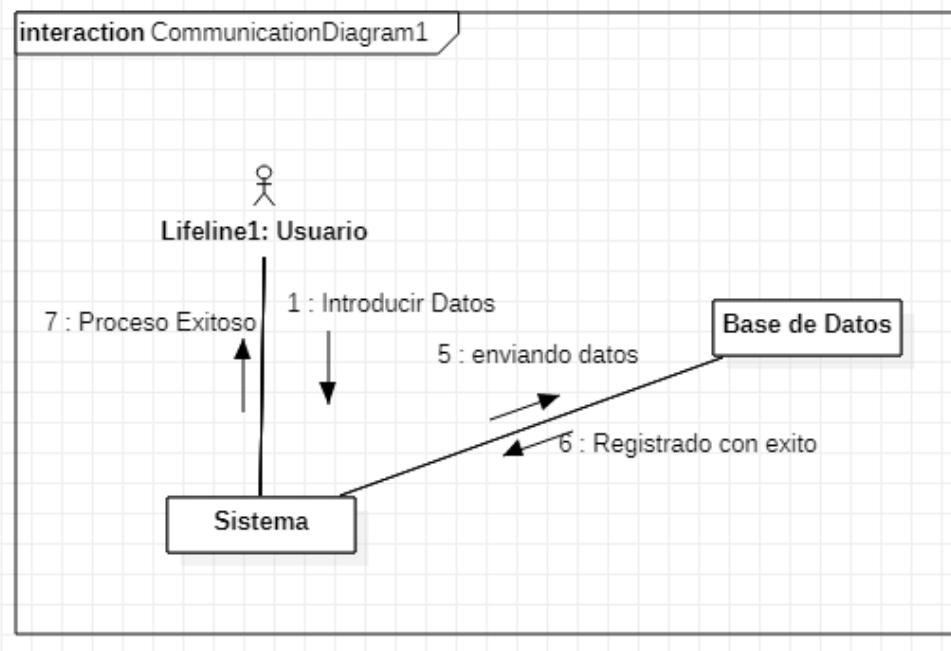
Figura 3.14.: Diagrama de Colaboración Ingresar al Sistema (Usuario).



Fuente: Elaboración Propia.

Registrarse (Usuario)

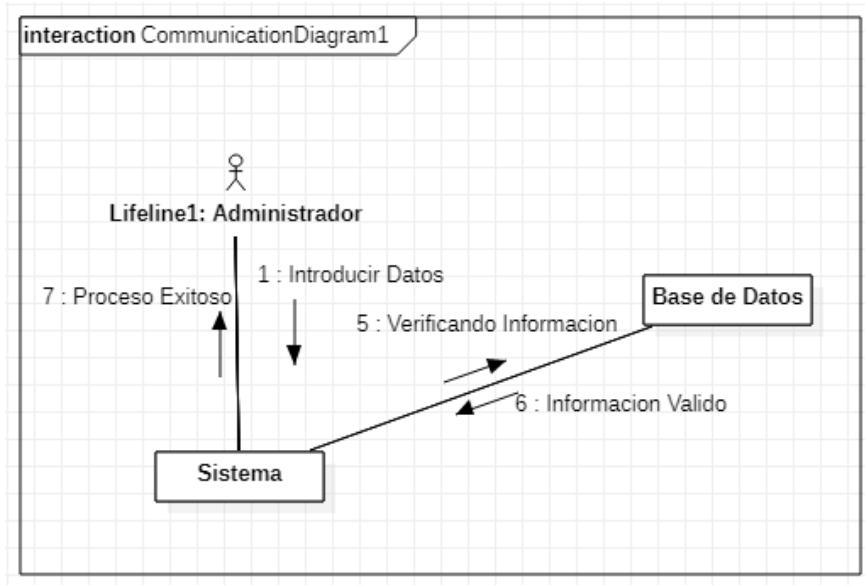
Figura 3.15.: Diagrama de Colaboración Registrarse (Usuario).



Fuente: Elaboración Propia.

Ingresar al sistema (Administrador)

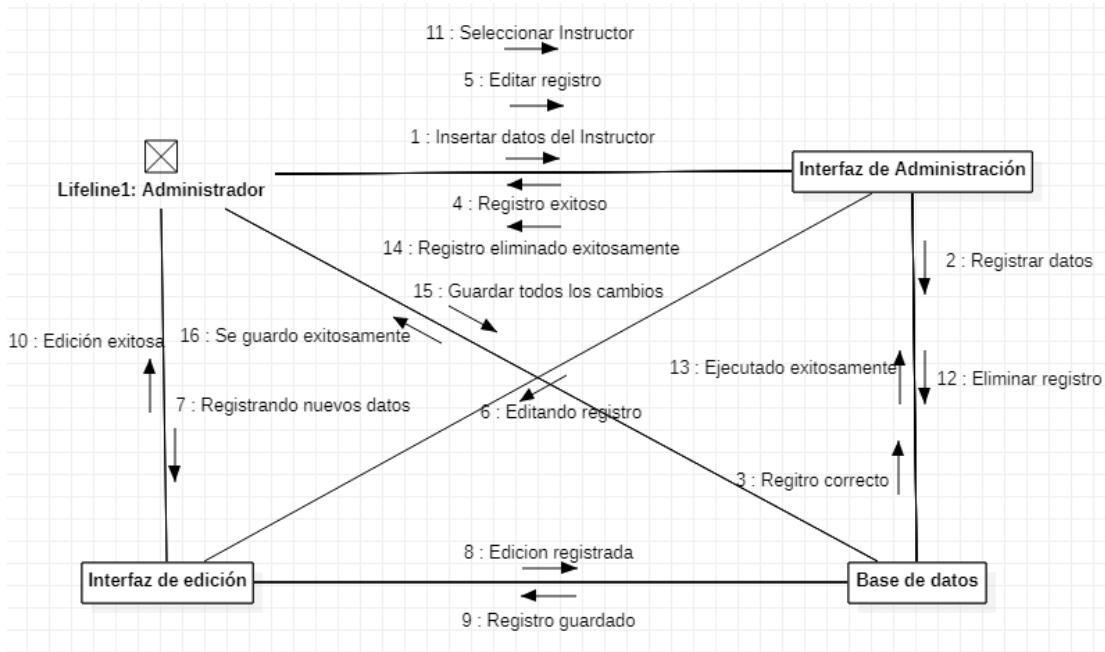
Figura 3.16.: Diagrama de Colaboración Ingresar al Sistema (Administrador).



Fuente: Elaboración Propia.

Administrar Instructores

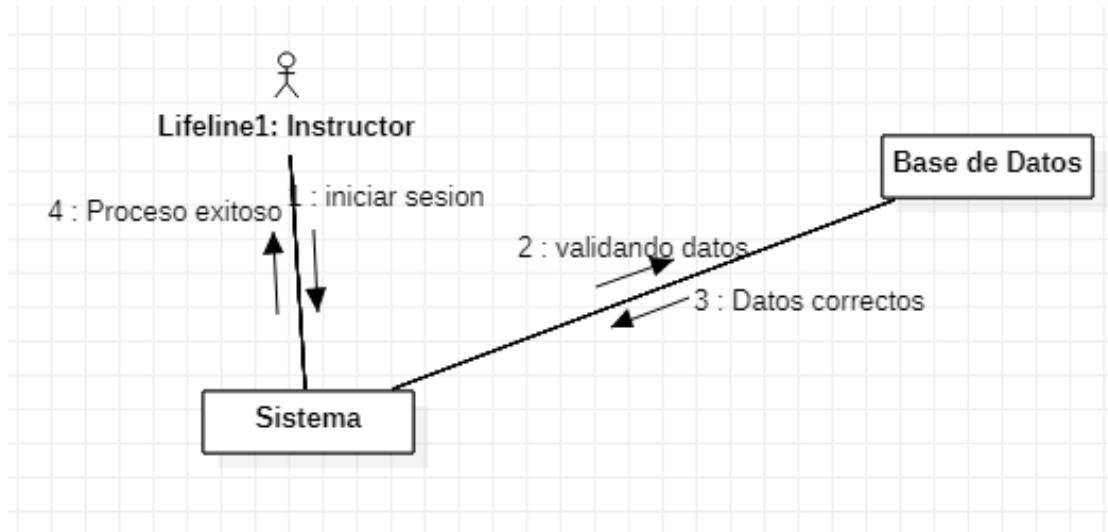
Figura 3.17.: Diagrama de Colaboración Administrar Instructores.



Fuente: Elaboración Propia.

Ingresar al Sistema (Instructor)

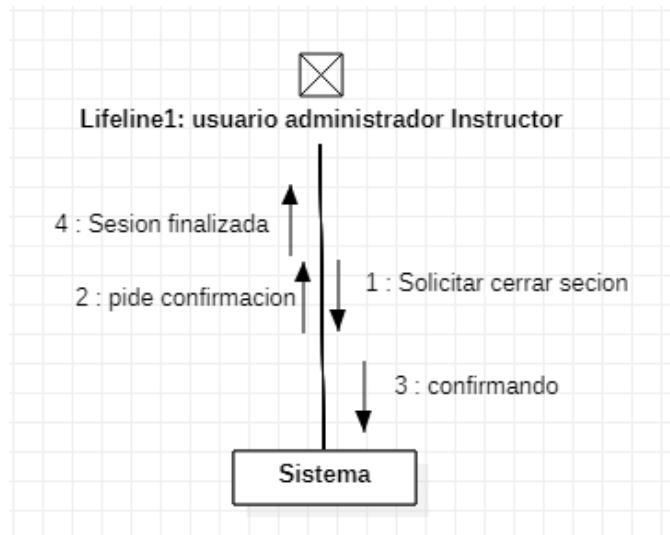
Figura 3.18.: Diagrama de Colaboración Ingresar al Sistema (Instructor).



Fuente: Elaboración Propia.

Finalizar sesión (Usuario, Administrador, Instructor).

Figura 3.19.: Diagrama de colaboración Finalizar Sesión.

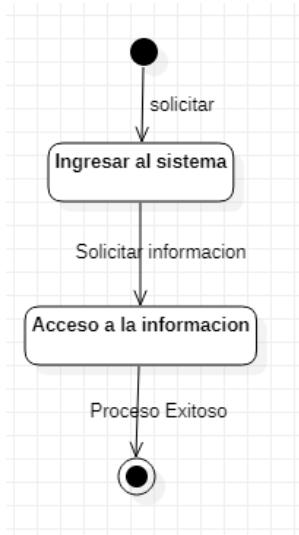


Fuente: Elaboración Propia.

Diagramas de Estado

Ingresar al Sistema (Usuario)

Figura 3.20.: Diagrama de Estado Ingresar al Sistema (Usuario).



Fuente: Elaboración Propia.

Registrarse (Usuario)

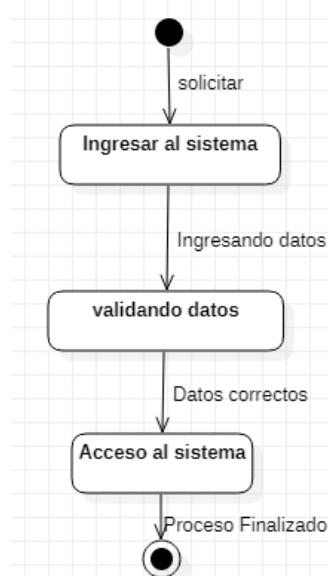
Figura 3.21.: Diagrama de Estado Registrarse (Usuario).



Fuente: Elaboración Propia.

Ingresar al sistema (Administrador)

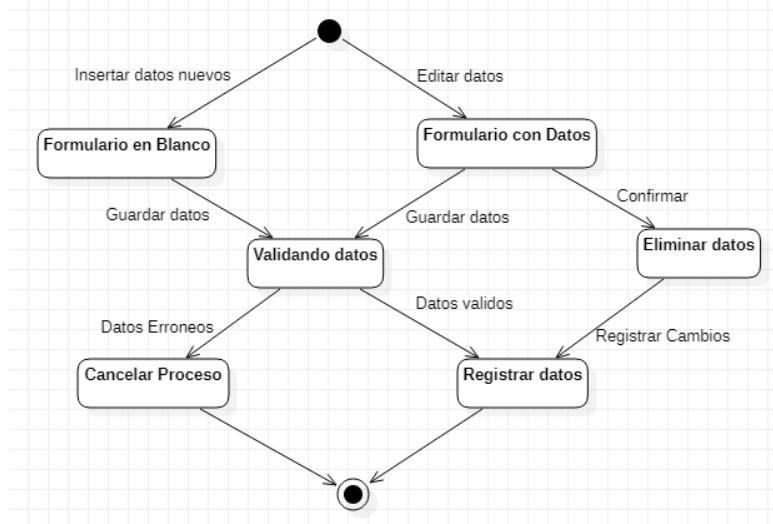
Figura 3.22.: Diagrama de Estado Ingresar al Sistemas (Administrador).



Fuente: Elaboración Propia.

Administrar Instructores

Figura 3.23.: Diagrama de Estado Administrar Instructores.



Fuente: Elaboración Propia.

Ingresar al sistema (Instructor).

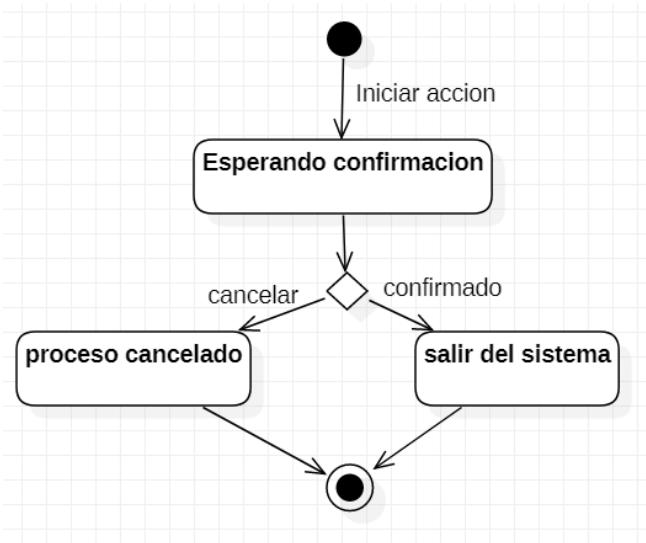
Figura 3.24.: Diagrama de Estado Ingresar al Sistema (Instructor).



Fuente: Elaboración Propia.

Cerrar Sesión (Usuario, administrador e instructor)

Figura 3.25.: Diagrama de Estado Cerrar Sesión (Usuario, Administrador e Instructor).

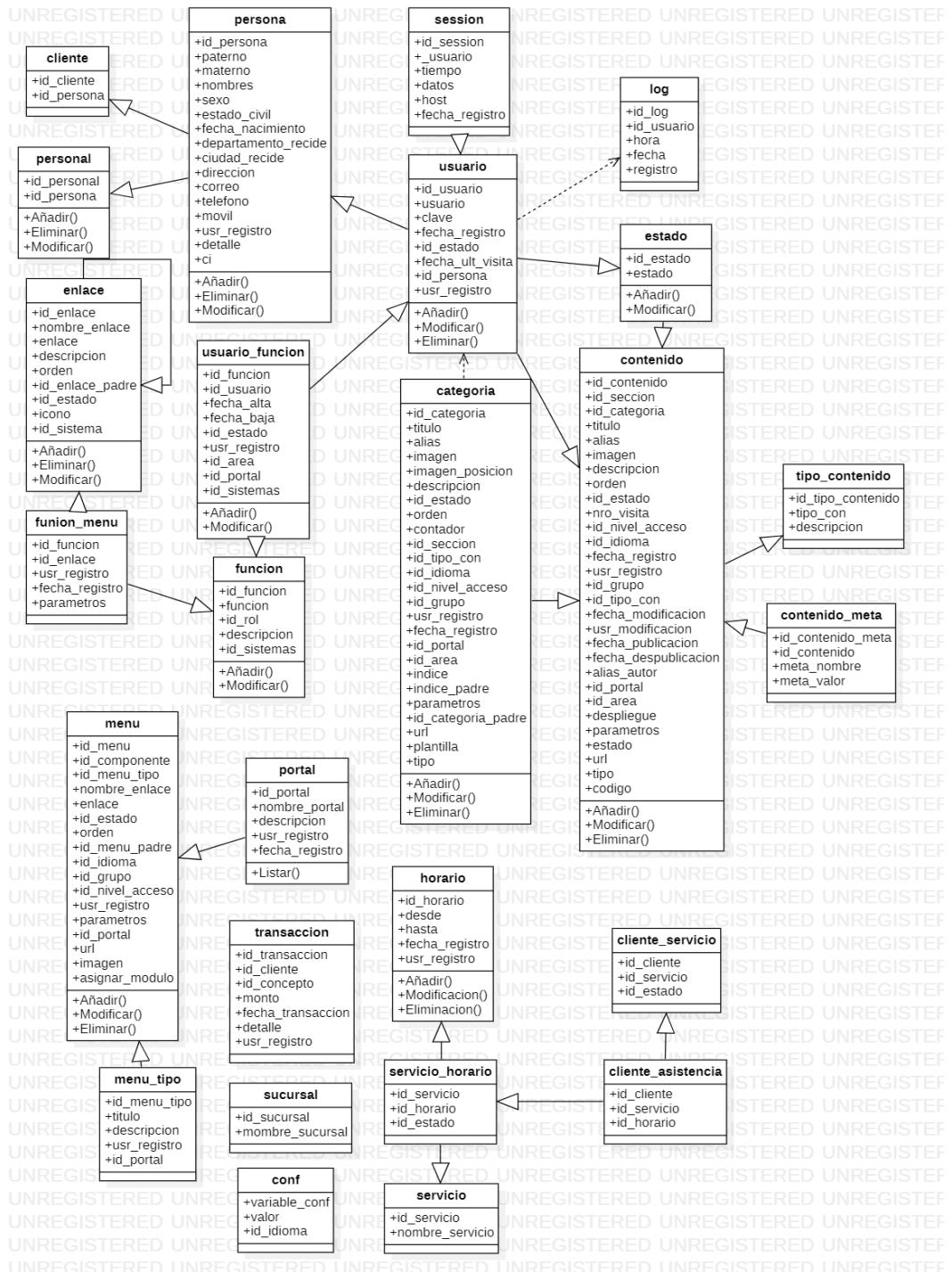


Fuente: Elaboración Propia.

3.3 DISEÑO DEL SISTEMA

3.3.1 DIAGRAMA DE CLASES

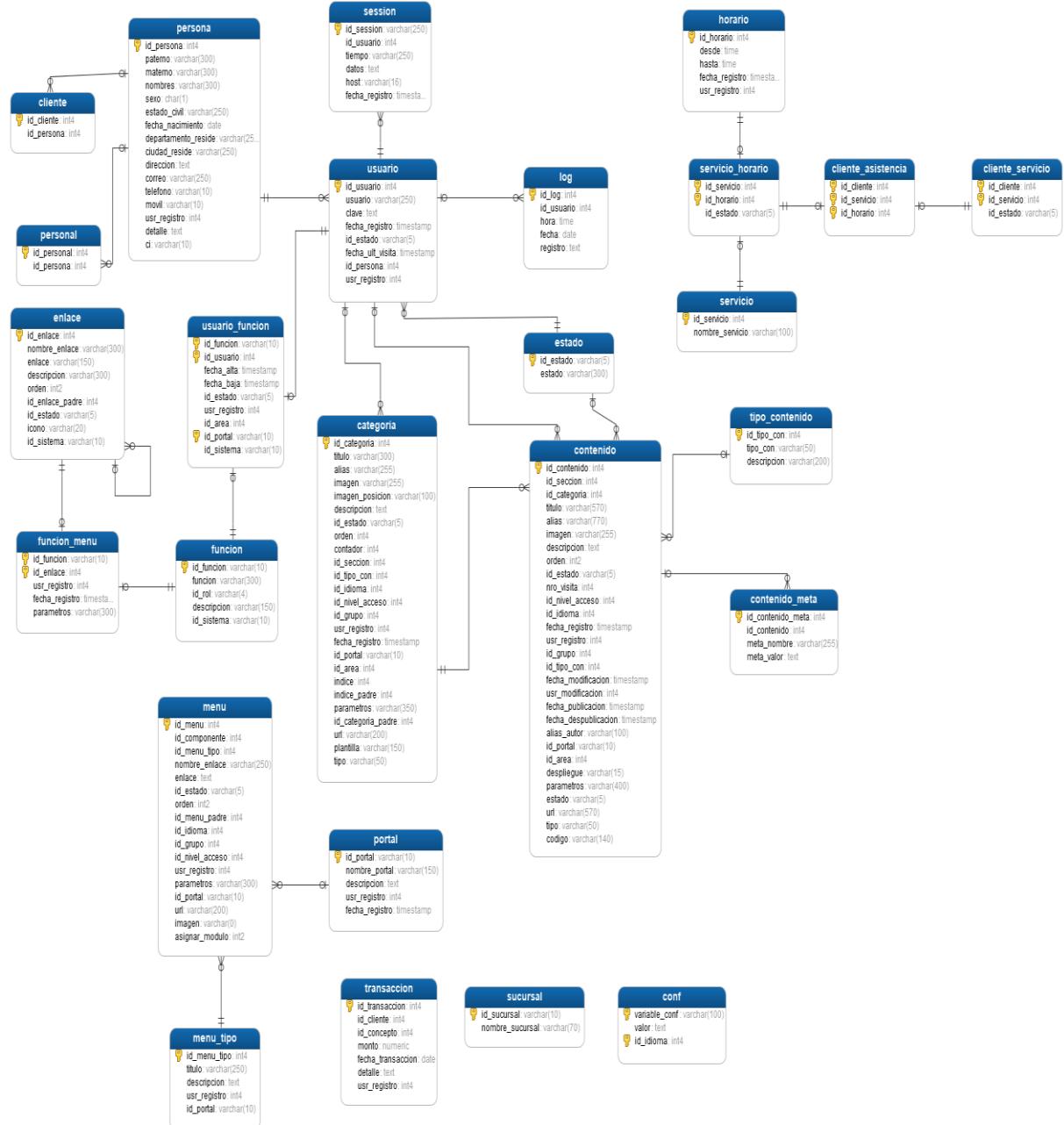
Figura 3.26.: Modelo de Presentación.



Fuente: Elaboración Propia.

3.3.2 MODELO RELACIONAL

Figura 3.27.: Modelo de Presentación.



Fuente: Elaboración Propia.

3.3.3 CODIFICACIÓN

A continuación, se muestra la codificación de los más relevante del sistema como ser el Login y el usuario que es muy importante para nuestro sistema de administración del Gimnasio Winner.

Figura 3.28.: Codificación Login.

```
public function Login() {  
  
    $vista = Principal::getVistaInstance();  
    $permitidos = 3;  
    $tiempo = 60; //60 segundos  
  
    if (isset($_SESSION['tiempo_fuera'])) {  
  
        $tiempo_ahora = time() - $_SESSION['tiempo_fuera'];  
        if ($tiempo_ahora < $tiempo) {  
            $tiempo_restante = $tiempo - $tiempo_ahora;  
            die('Debes esperar ' . $tiempo_restante . ' segundos para poder intentar el login de nuevo.<br /><br /><a href="' . BASEURI . 'index.php?c=Login&a=Login">' . $vista->getNombre() . '</a>');  
        } else {  
            unset($_SESSION['tiempo_fuera']);  
        }  
    }  
  
    if (!isset($_SESSION['intentos'])) {  
        $intento = 0;  
        $_SESSION['intentos'] = $intento;  
    } else {  
        $intento = $_SESSION['intentos'];  
    }  
  
    if ($intento >= $permitidos) {  
        unset($_SESSION['intentos']);  
        $_SESSION['tiempo_fuera'] = time();  
        die('Ha sobrepasado el numero permitido de intentos de login. No podra loguearse por ' . $tiempo . ' segundos.<br /><br /><a href="' . BASEURI . 'index.php?c=Login&a=Login">' . $vista->getNombre() . '</a>');  
    }  
  
    $datosUsuario = self::getLoginUsuario($_POST["usuario"], $_POST["clave"]);  
    if (!empty($datosUsuario)) {  
        $acceso = new SQLAcceso();  
        $menu = new Menu();  
        $sqlAdmPMenu = new SQLAdmPMenu();  
    }  
}
```

Fuente: Elaboración Propia.

Figura 3.29.: Control de Acceso Usuario.

```

.ass Usuario extends Sistema {
    private $id_usuario;
    private $id_persona;
    private $nombre_usuario;
    private $clave;
    private $nombres;
    private $paterno;
    private $materno;
    private $sexo;
    private $estado_civil;
    private $fecha_nacimiento;
    private $departamento_reside;
    private $ciudad_reside;
    private $correo;
    private $telefono;
    private $movil;
    private $ci;
    private $direccion;
    private $detalle;

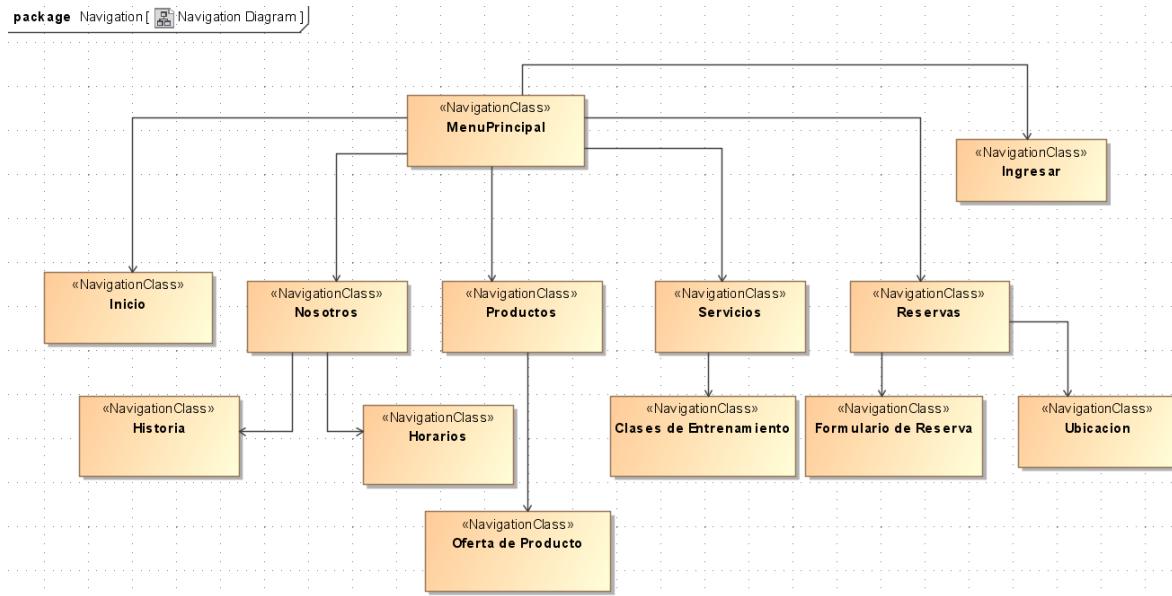
    public function setId_usuario($id_usuario) {
        $this->id_usuario = $id_usuario;
    }
    public function getId_usuario() {
        return $this->id_usuario;
    }
    public function setNombre_usuario($nombre_usuario) {
        $this->nombre_usuario = $nombre_usuario;
    }
    public function getNombre_usuario() {
        return $this->nombre_usuario;
    }
    public function setClave($clave) {
        $this->clave = $clave;
    }
    public function getClave() {
        return $this->clave;
    }
    public function setNombres($nombres) {
        $this->nombres = $nombres;
    }
    public function getNombres() {
        return $this->nombres;
    }
    public function setPaterno($paterno) {
        $this->paterno = $paterno;
    }
    public function getPaterno() {
        return $this->paterno;
    }
    public function setMaterno($materno) {
        $this->materno = $materno;
    }
    public function getMaterno() {
        return $this->materno;
    }
    public function setSexo($sexo) {
        $this->sexo = $sexo;
    }
    public function getSexo() {
        return $this->sexo;
    }
    public function setEstado_civil($estado_civil) {
        $this->estado_civil = $estado_civil;
    }
    public function getEstado_civil() {
        return $this->estado_civil;
    }
    public function setFecha_nacimiento($fecha_nacimiento) {
        $this->fecha_nacimiento = $fecha_nacimiento;
    }
    public function getFecha_nacimiento() {
        return $this->fecha_nacimiento;
    }
    public function setDepartamento_reside($departamento_reside) {
        $this->departamento_reside = $departamento_reside;
    }
    public function getDepartamento_reside() {
        return $this->departamento_reside;
    }
    public function setCiudad_reside($ciudad_reside) {
        $this->ciudad_reside = $ciudad_reside;
    }
    public function getCiudad_reside() {
        return $this->ciudad_reside;
    }
    public function setCorreo($correo) {
        $this->correo = $correo;
    }
    public function getCorreo() {
        return $this->correo;
    }
    public function setTelefono($telefono) {
        $this->telefono = $telefono;
    }
    public function getTelefono() {
        return $this->telefono;
    }
    public function setMovil($movil) {
        $this->movil = $movil;
    }
    public function getMovil() {
        return $this->movil;
    }
    public function setCi($ci) {
        $this->ci = $ci;
    }
    public function getCi() {
        return $this->ci;
    }
    public function setDireccion($direccion) {
        $this->direccion = $direccion;
    }
    public function getDireccion() {
        return $this->direccion;
    }
    public function setDetalle($detalle) {
        $this->detalle = $detalle;
    }
    public function getDetalle() {
        return $this->detalle;
    }
    public function setId_persona($id_persona) {
        $this->id_persona = $id_persona;
    }
    public function getId_persona() {
        return $this->id_persona;
    }
}

```

Fuente: Elaboración Propia.

3.3.4 MODELO DE NAVEGACIÓN

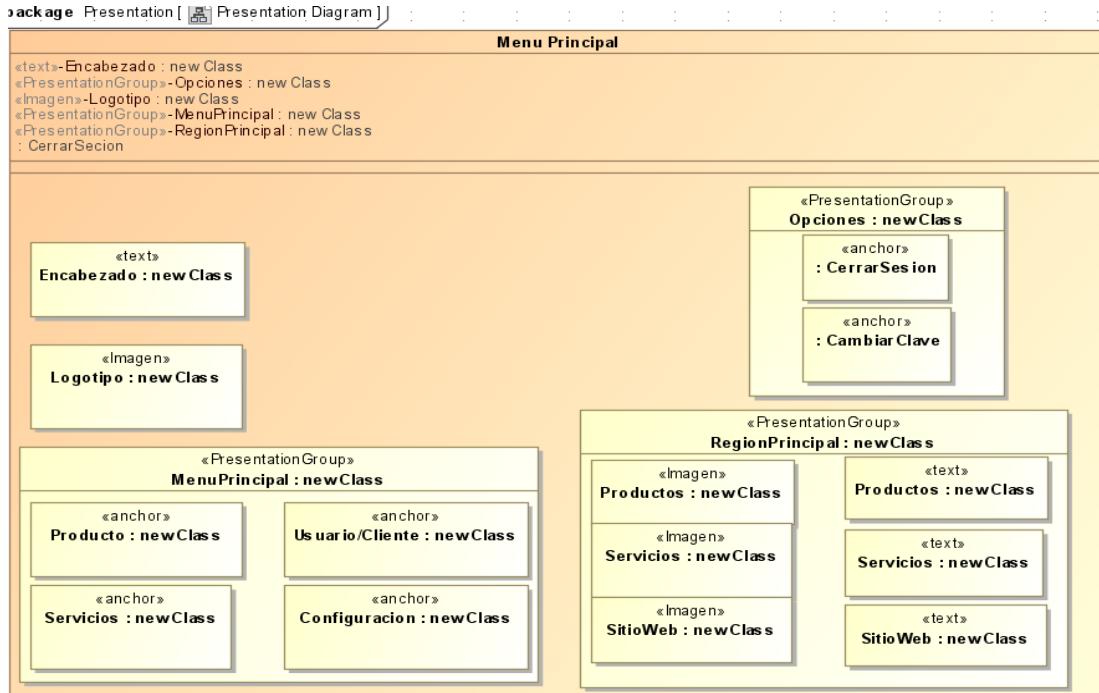
Figura 3.30.: Modelo de Navegación.



Fuente: Elaboración Propia.

3.3.5 MODELO DE PRESENTACIÓN

Figura 3.31.: Modelo de Presentación.



Fuente: Elaboración Propia.

4 MARCO APlicativo

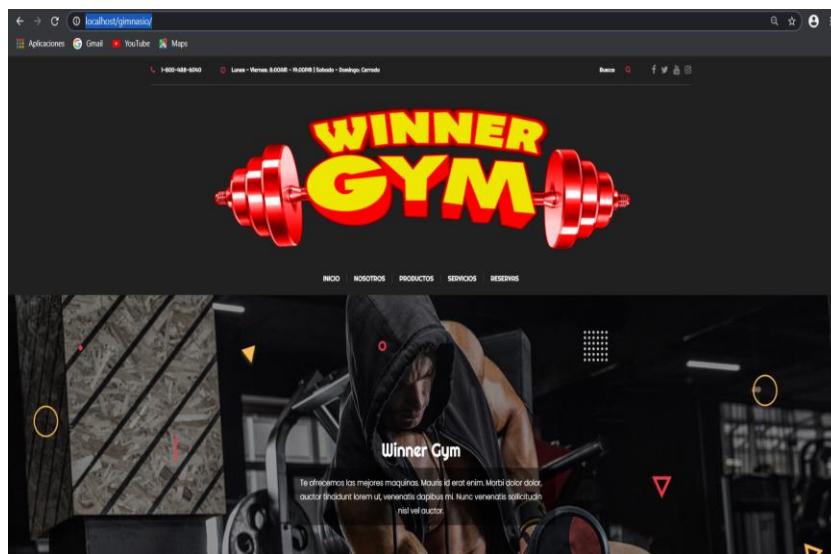
4.1 IMPLEMENTACIÓN DEL SISTEMA

4.2 SITIO WEB (FRONT END)

En esta parte podemos ver el sitio web donde contamos con 5 opciones (Nosotros, Productos, Servicios, Reservas e Ingresar).

- **Nosotros:** En esta parte podremos ver la historia como también la misión y visión que tiene el Gimnasio.
- **Productos:** En esta opción el cliente podrá ver todos los productos que podemos adquirir en el gimnasio y así también que beneficio leído al cuerpo el dicho producto.
- **Servicios:** Es donde el cliente podrá ver todos los servicios que ofrece el gimnasio y sus beneficios de cada una de ellas.
- **Reservas:** En esta parte el cliente podrá realizar su reserva preguntando todo lo que necesite acerca del gimnasio.
- **Ingresar:** En esta parte el (cliente, Instructor y administrador) una vez obtenido su usuario y contraseña) podrán ingresar al sistema.

Figura 4.1.: Sitio web.

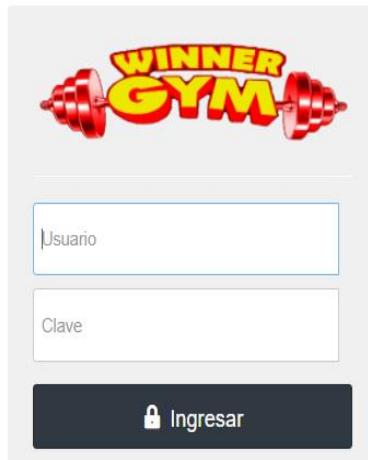


Fuente: Elaboración Propia.

4.3 INGRESO AL SISTEMA (BACKEND)

El administrador, instructor, usuario deben ingresar su Usuario y contraseña.

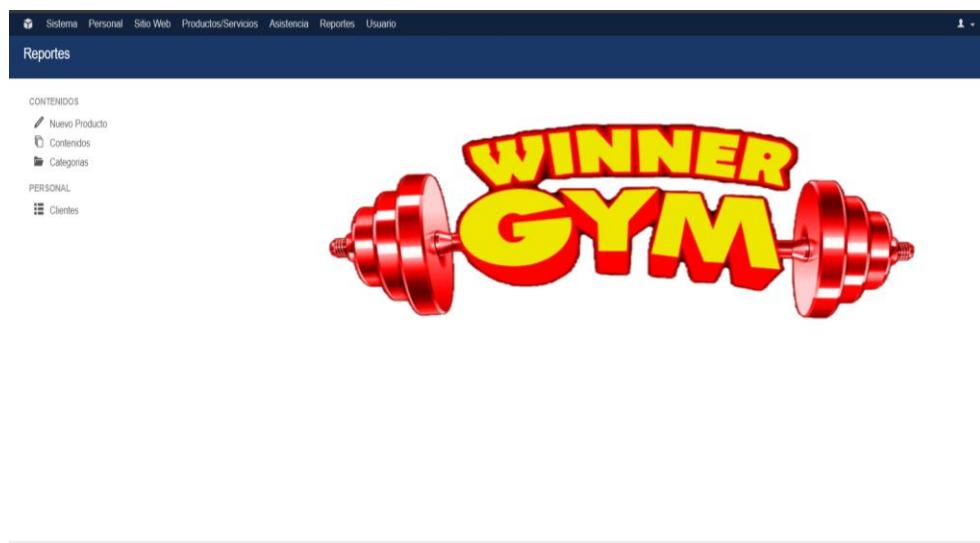
Figura 4.2.: Ingreso al Sistema.



Fuente: Elaboración Propia.

- **Sistema:** En esta parte se podrá desplegar un menú de (Inicio, Menu Panel, Asignar Menu, Roles/Cargos, Configuración del Sitio y Salir).
- **Personal:** Administra todo lo que es el personal administrativo, instructores y Usuario/clientes, también podemos realizar los dichos registros de cada uno de ellos.
- **Sitio web:** En este módulo contamos con el siguiente menú (Multimedia, Menús, Categorías, Formulario de Registro Contenidos y Archivos Multimedia).
- **Productos/Servicios:** En este Modulo Contamos con dos menús importantes para el gimnasio que son (Productos y Servicios) que son muy importantes.
- **Asistencia:** En esta parte veremos a todos los Usuarios/Clientes activos, como también podremos ver a los Instructores y Personal Administrativo.
- **Reportes:** En este menú realizaremos la descarga de un pdf mostrando el reporte de todos los Usuario/clientes registrados.
- **Usuario:** En esta parte tenemos los siguientes menús (Mis Datos Y Cambiar Clave).

Figura 4.3.: Sistema Administrativo (Back end).



Fuente: Elaboración Propia.

4.4 EVALUACIÓN DE CALIDAD

4.4.1 EVALUACIÓN DE CALIDAD DEL PORTAL WEB

Operación del Producto.

Tabla 4.1.: Factor Operación del Producto.

FACTOR	METRICA	SI	NO
FACILIDAD DE USO	1. ¿La elaboración y diseño satisface las necesidades del Gimnasio Winner?.	X	
	2. ¿Los Usuarios tienen una interacción efectiva en base a conocimientos previos con el sitio?.	X	
	3. ¿El entorno del sistema ayuda para que nuevos Gimnasios puedan emplear el sistema?.	X	

INTEGRIDAD	4. ¿El sistema posee atributos de control de acceso?. X	
	5. ¿Se dispone de mecanismos que protejan y controlen los datos del sistema?. X	
CORRECIÓN	6. ¿Poseen cualidades que proporcionan uniformidad en las técnicas y notaciones de diseño e implementación?. X	
	7. ¿Proporciona una traza desde los requisitos a la implementación con respecto a un entorno operativo concreto?. X	
	8. ¿Se cuenta con atributos que proporcionen la implementación de todas las funciones requeridas?. X	
FIABILIDAD	9. ¿Los resultados obtenidos son consistentes con los requerimientos?. X	
	10. ¿Hay precisión con los resultados obtenidos?. X	
	11. ¿Puede el usuario reestablecer fácilmente la página cuando esté en uso?. X	
	12. ¿Puede el Usuario mandar sus datos con seguridad?. X	
	13. ¿Hay exactitud en los resultados obtenidos?. X	
EFICIENCIA	14. ¿Las respuestas del sistema son rápidas?. X	
	15. ¿Puede el usuario guardar su información generada con el software, en la memoria del sistema?. X	

Fuente: Elaboración Propia.

Revisión del Producto.

Tabla 4.2.: Factor de Revisión del Producto.

FACTOR	METRICA	SI	NO
FACILIDAD DE MANTENIMIENTO	16. ¿El Sistema está dividido en componentes funcionales tiene acoplamiento y cohesión ?.	X	
	17. ¿Las aplicaciones son consistentes con conocimientos previos?.	X	
	18. ¿El Sistema está orientado hacia la simplicidad y facilidad operativa, en su manejo y mantenimiento?.	X	
	19. ¿El Sistema posibilita la implementación de una función con la menor calidad de código posible?.	X	
FACILIDAD DE PRUEBA	20. ¿Provee una plataforma modular de prueba para solucionar problemas rápidamente?.	X	
	21. ¿Resulta simple que el Sistema muestre sus fallas a través de la ejecución de pruebas?.	X	
	22. ¿Posee atributos auto descriptivos que permiten realizar pruebas sobre el Sistema?.	X	
FLEXIBILIDAD	23. ¿Existen variedad de posibilidades con las que el usuario y el Sistema puedan intercambiar información?.	X	
	24. ¿Hay la posibilidad de la expansión del Sistema en cuanto a capacidades funcionales y datos?.	X	
	25. ¿Tiene una serie de módulos que le dan potencia y flexibilidad?.	X	

Fuente: Elaboración Propia.

Transición del Producto.

Tabla 4.3.: Factor de Transición del Producto.

FACTOR	METRICA	SI	NO
REUSABILIDAD	26. ¿El portal tiene varias opciones para ingresar los datos: Archivos tipo texto, archivos provenientes de otras bases de datos?.		X
	27. ¿El usuario entiende fácilmente los mensajes que envía el Sistema?.	X	
	28. ¿La página contiene módulos para visualizar gráficamente resultados?.	X	
	29. ¿Los atributos del Sistema son dependientes del hardware?.	X	
INTEROPERABILIDAD	30. ¿El sistema dispone de mecanismos que permitan intercambiar procesos y/o datos?.	X	
	31. ¿Los atributos posibilitan el uso de protocolos de comunicación e interfaces estándares?.	X	
	32. ¿Usa estructuras de datos y de tipos estándar a lo largo de todo el programa?.	X	
PORTABILIDAD	33. ¿El sistema se puede adaptar a cualquier manejador de base de datos?.	X	
	34. ¿Al realizar mejoras del sistema, las funciones y módulos son similares?.	X	
	35. ¿Al desarrollar el sistema el hardware es un factor secundario?.	X	

Fuente: Elaboración Propia.

Una vez obtenido los resultados por cada una de las características se debe reemplazar los valores 0 a 3 dependiendo de la respuesta que se haya tenido.

Operación del Producto.

$$\text{Factor Facilidad de Uso} = \frac{\sum \text{Nº si}}{\text{Cantidad de preguntas}}$$

$$\text{Factor Facilidad de Uso} = \frac{3 + 3 + 3}{3} = 3$$

$$\text{Factor Integridad} = \frac{\sum \text{Nº si}}{\text{Cantidad de preguntas}}$$

$$\text{Factor Integridad} = \frac{3 + 3}{2} = 3$$

$$\text{Factor Corrección} = \frac{\sum \text{Nº si}}{\text{Cantidad de preguntas}}$$

$$\text{Factor Corrección} = \frac{3 + 3}{3} = 2$$

$$\text{Factor Fiabilidad} = \frac{\sum \text{Nº si}}{\text{Cantidad de preguntas}}$$

$$\text{Factor Fiabilidad} = \frac{3 + 3 + 3 + 3 + 3}{5} = 3$$

$$\text{Factor Eficiencia} = \frac{\sum \text{Nº si}}{\text{Cantidad de preguntas}}$$

$$\text{Factor Fiabilidad} = \frac{3 + 3}{2} = 3$$

Revisión del Producto.

$$\text{Factor Facilidad de Mantenimiento} = \frac{\sum \text{Nº si}}{\text{Cantidad de preguntas}}$$

$$\text{Factor Facilidad de Mantenimiento} = \frac{3 + 3 + 3 + 3}{4} = 3$$

$$\text{Factor Facilidad de Prueba} = \frac{\sum \text{Nº si}}{\text{Cantidad de preguntas}}$$

$$\text{Factor Facilidad de Prueba} = \frac{3 + 3 + 3}{3} = 3$$

$$\text{Factor Flexibilidad} = \frac{\sum \text{Nº si}}{\text{Cantidad de preguntas}}$$

$$\text{Factor Flexibilidad} = \frac{3 + 3 + 3}{3} = 3$$

Transición del Producto.

$$\text{Factor Reusabilidad} = \frac{\sum \text{Nº si}}{\text{Cantidad de preguntas}}$$

$$\text{Factor Reusabilidad} = \frac{3 + 3 + 3}{4} = 2,25$$

$$\text{Factor Interoperabilidad} = \frac{\sum \text{Nº si}}{\text{Cantidad de preguntas}}$$

$$\text{Factor Interoperabilidad} = \frac{3 + 3 + 3}{3} = 3$$

$$\text{Factor Portabilidad} = \frac{\sum \text{Nº si}}{\text{Cantidad de preguntas}}$$

$$\text{Factor Portabilidad} = \frac{3 + 3 + 3}{3} = 3$$

Medición del porcentaje de calidad.

$$\% \text{ calidad} = \frac{(3 + 3 + 2 + 3 + 3) + (3 + 3 + 3) + (2,25 + 3 + 3)}{\text{Cantidad de preguntas}}$$

$$= \frac{14 + 9 + 8,25}{11} = 3$$

$$= \frac{30,25}{11} = 2,84$$

Se realiza la conversión respectiva.

$$\% \text{ calidad} = \frac{100 * 2,84}{3} = 94,66\%$$

Con respecto al portal web se puede afirmar que su porcentaje de calidad es de un 91,66%, nivel que es aceptable.

4.5 ANÁLISIS DE COSTOS.

Uno de los trabajos más importantes al momento de desarrollar un Sistema es la estimación de costos, la cual consiste en determinar con cierto grado de certeza los recursos de hardware y software, costo, tiempo y esfuerzo necesario para el desarrollo de los mismos.

4.5.1 COCOMO II.

Cocomo II nos permite realizar estimaciones en función al tamaño del software. Se utilizará el modelo Post-Arquitectura.

Puntos de Función.

Tabla 4.4.: Puntos de Función del Sistema.

PARAM. MEDICIÓN	CANTIDAD	FACTOR DE PONDERACIÓN			SUB TOTAL
Entradas de Usuarios	3			6	36
	6	3			
Salidas de Usuarios	2	4			42
	4		5		
	2			7	
Peticiones de Usuarios	5		4		20
Archivos	0				0
Interfaces Externas	0				0
Total					98

Fuente: Elaboración Propia.

Los Puntos de Función de todo el proyecto es:

PF=PF del Sistema de administración.

PF= 98

Factor de complejidad (Fi).

Los valores son expresados de 0 a 5, dónde cero no es importante y cinco es fundamental.

Tabla 4.5.: Factor de Complejidad.

Nro.	Factor de Complejidad	Valor
1	Requiere copias de seguridad.	2
2	Necesita comunicación de datos.	5
3	Existe funciones de procedimientos distribuidos.	3
4	Rendimiento crítico.	2
5	Se ejecuta en un entorno operativo existente.	1
6	Se requiere entrada de datos en línea.	3
7	Transacciones de entradas en múltiples pantallas.	5
8	Archivos maestros actualizados en línea.	1
9	Complejidad de valores del dominio de la información.	2
10	Complejidad del procedimiento interno.	2
11	Código diseñado para reutilización.	5
12	Conversión/instalación de diseño.	1
13	Instalaciones múltiples.	1
Total.		33

Fuente: Elaboración Propia.

Puntos de Función Ajustados.

Se reemplaza los PF y la sumatoria de Factor de Complejidad en la fórmula de Puntos de Función Ajustados.

$$PFA = PF * (0.65 + 0.01 * \sum F_i)$$

$$PFA = 98 * (0.65 + 0.01 * 33)$$

$$PFA = 96$$

Líneas de Código.

Es necesario calcular las líneas de código (LDC) para ello el Factor LDC/PF es de 47 para el JavaScript porque es el lenguaje de programación que manejaremos en torno a la librería JQuery y para el desarrollo del sistema en el Framework Phalcon de PHP que está basado en MVC.

$$LDC = PFA * (\text{Factor LDC/PF})$$

$$LDC = 96 * 47$$

$$LDC = 4.512$$

Para calcular KLDC.

$$KLDC = LDC / 1000$$

$$KLDC = 4.512 / 1000$$

$$KLDC = 4.512$$

Factor de Escala.

A continuación, se debe calcular la sumatoria de los Factores de Escala ($\sum W_j$).

Tabla 4.6.: Factores de Escala.

Factor de Escala	Valor
PREC	3.72
FLEX	4.05
RESL	4.24
TEAM	2.19
PMAT	4.68
Total Factores de Escala ($\sum W_j$).	18.88

Fuente: Elaboración Propia.

Calculamos (B) Factor exponencial de escala, basado en factores de escala que influyen exponencialmente en la productividad.

$$B = 0.91 + 0.01 * (\sum W_j)$$

$$B = 0.91 + 0.01 * (18.88)$$

$$B = 1.0988$$

Esfuerzo Nominal (PM).

Se obtiene el Esfuerzo Nominal reemplazando los datos en la siguiente ecuación:

$$PM_{nominal} = A * (KLDC)B$$

$$PM_{nominal} = 2.94 * (4,512)^{1.0988}$$

$$PM_{nominal} = 15.394588$$

Multiplicadores de Esfuerzo.

Tabla 4.7.: Multiplicadores de Esfuerzo.

Conductores de Costo	Nro.	Multiplicadores de Esfuerzo	Valor
PRODUCTO	1	RELY	1
	2	DATA	1
	3	CPLX	0.87
	4	RUSE	1.1
	5	DOCU	0.91
PLATAFORMA	6	TIME	1.1
	7	STOR	1
	8	PVOL	1
PERSONAL	9	ACAP	0.9
	10	PCAP	1
	11	PCON	0.9
	12	AEXP	0.9
	13	PEXP	0.9
	14	LTEX	0.9
PROYECTO	15	TOOL	1
	16	SITE	1
	17	SCED	1.1
TOTAL PRODUCTO N°em			0.622230

Fuente: Elaboración Propia.

El Esfuerzo estimado se calcula según la siguiente fórmula:

$$PM_{estimado} = PM_{nominal} * \sum EM_{17} \quad i=1$$

$$PM_{estimado} = 15.394588 * 0.622230$$

$$PM_{estimado} = 9.57897 \text{ [Personas/Mes]}$$

Esto indica que se necesitarían nueve personas trabajando a jornada completa por un mes, para desarrollar el proyecto. Además, el salario promedio que percibe un desarrollador es de Bs 6.966 y Bs 8.352 según (Daniel Calbimonte, 2010).

Tabla 4.8.: Escala de Salarios.

Empresa	Salario Bs
Jalasoft	6.966
Assuresoft	4.182
Viva	4.176
Promedio	5.108

Fuente: (Daniel Calbimonte, 2010).

Sumando los salarios en Bs de las nueve personas por mes sería.

$$\text{Costo de Desarrollo} = 9 * 5.108$$

$$\text{Costo de Desarrollo} = 45.972$$

Se llega a la conclusión que el desarrollo tanto del Sistema Web un costo de Bs 45.972.

4.6 SEGURIDAD

4.6.1 ENCRIPCIÓN

Para mantener la seguridad en el manejo de las contraseñas de los usuarios del sistema, en este caso el administrador y los usuarios con acceso al portal de administración se debe encriptar las contraseñas.

Figura 4.4.: Seguridad de Contraseña.

id_usuario	usuario	clave	fecha_registro	id_estado	fecha_ult_visita	id_persona	usr_registro
1749	pedro	40bd001563085fc35165329ea1ff5c5ecbdbbeef	2020-11-22 03:45:12	A	(Null)	1805	1741
1741	estanly	40bd001563085fc35165329ea1ff5c5ecbdbbeef	2020-07-08 20:48:01	A	2020-11-22 15:53:40	1783	30

Fuente: Elaboracion Propia.

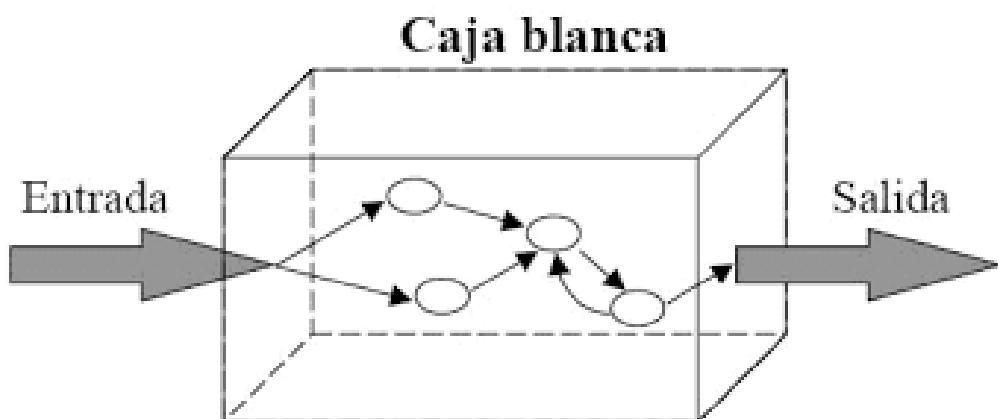
4.7 PRUEBAS DE SOFTWARE

Ahora queda adentrarnos en las evaluaciones dinámicas comúnmente denominadas las pruebas de software, para dicha tarea donde emplea las técnicas de pruebas caja negra y caja blanca.

4.7.1 PRUEBAS DE CAJA BLANCA

Para esta prueba es necesario tener un conocimiento minucioso acerca de la lógica o código a evaluar, pues tomaremos como prioridad el comportamiento interno y la estructura del programa, aplicando la cobertura de caminos describiremos casos de pruebas suficientes para que se ejecuten todos los caminos de un programa, entendiendo camino como una sentencia de sentencias encadenadas desde la entrada hasta su salida.

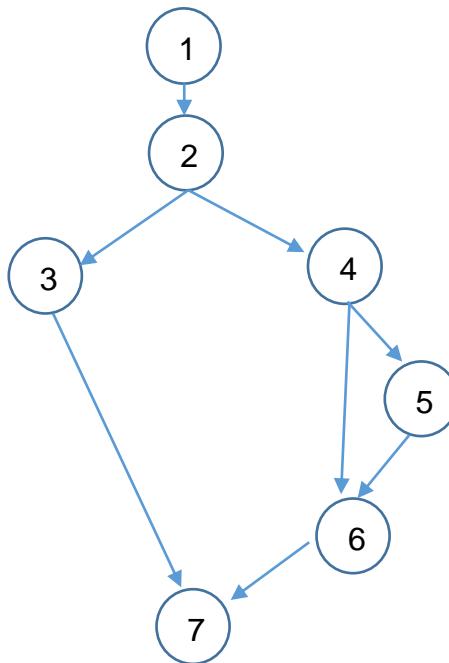
Figura 4.5.: Representación Prueba de Caja Blanca.



Fuente: Elaboración propia

Prueba de Caja Negra - Inicio de Sesión

Figura 4.6.: Prueba de Caja Blanca – Inicio de Sesión.



Fuente: Elaboración Propia.

Donde:

- Inicio de Sesión del sistema (1)
- Verificar el Ingreso de datos (2)
- Error en la verificación de datos (3)
- Datos correctos (4)
- Carga de permisos de acceso e ingreso al sistema (5)
- Cargar de mensajes (6)
- Fin del ciclo (7)

Analizando el grafo se obtiene lo siguiente:

$$V(G) = A - N + 2$$

Reemplazando se obtiene:

$$V(G) = 8 - 7 + 2 = 3$$

$$\therefore V(G) = 3$$

Ahora determinamos el conjunto de caminos independientes. Los caminos deben ser probados dadas ciertas variables.

Camino 1: 1, 2, 3, 7

Camino 2: 1, 2, 4, 6, 7

Camino 3: 1, 2, 4, 5, 6, 7

Tabla 4.9.: Evaluación de Flujo.

Nro.	Ruta	Parámetro de Entrada
1	1,2,3,7	Se verifica la información ingresada, en esta verificación se solicita que ambos campos sean introducidos. En caso que existiera error se lanza el mensaje respectivo.
2	1,2,4,6,7	Se verifica que ambos campos sean ingresados, y posteriormente se verifica las credenciales. En caso que exista un error se carga los mensajes correspondientes del error cometido.
3	1,2,4,5,6,7	Se verifica que ambos capos sean ingresados, posteriormente se verifican las credenciales, de tener éxito se identifica el tipo de usuario para posteriormente cargar los permisos asignados y permitir el ingreso al sistema.

Fuente: Elaboración Propia.

4.7.2 PRUEBA DE CAJA NEGRA

Las pruebas de caja se aplicaron en las siguientes interfaces:

Prueba de Caja Negra Iniciar Sesión

Figura 4.7.: Prueba de Caja Negra – inicio sesión.

The image shows a login screen for 'WINNER GYM'. At the top is the gym's logo, which consists of the word 'WINNER' in yellow with a red outline, and 'GYM' in red with a yellow outline, flanked by two red dumbbells. Below the logo are two input fields: 'Usuario' (User) and 'Clave' (Password). At the bottom is a dark blue button with a white lock icon and the text 'Ingresar' (Enter).

Fuente: Elaboración Propia.

Tabla 4.10.: Valores límites – inicio de sesión.

Campo	Entradas Validas	Entradas Invalidas
Nombre de Usuario	Cadena de texto	Caracteres especiales como por ejemplo: espacios
Contraseña o Clave	Cadena de texto alfanumérico	De igual forma no se permiten caracteres de espacios

Fuente: Elaboración Propia.

Tabla 4.11.: Prueba de Caja Negra – Inicio de Sesión.

Entrada de Datos		salida	Resultados
usuario	Contraseña		
estanly	123Whugt	Se despliega un mensaje de bienvenida al sistema.	El sistema verifica la existencia del usuario y verifica si la clave o contraseña coinciden con los registros de la base de datos, permitiendo así identificar el tipo de usuario y la carga de los respectivos permisos de acceso al sistema.
Marcos	654asfaee	Despliega un mensaje acerca del error que se está cometiendo con las credenciales.	El sistema verifica la existencia del usuario y verifica si la clave o contraseña coinciden con los registros de la base de datos, si se halla algún error se despliega el mensaje respectivo indicando el error cometido.

Fuente: Elaboración Propia.

Como se puede observar la pantalla de inicio de sesión que despliega el sistema cumple con la función programada para un determinado usuario se identifique y pueda acceder al sistema.

Prueba de Caja Negra - Reserva

A continuación, se realiza la prueba de caja negra para la reserva de cupos.

Figura 4.8.: Prueba de Caja Negra – Reserva.

Reservas

Registra tus datos y nos pondremos en contacto contigo.

AYUDA | 75812742

PREGUNTAS | WinnerGimnasio@gmail.com

NUESTRA DIRECCION | La Paz - El Alto Zona. Villa Tunari, AV. Juan pablo II, Edificio San Juan de Dios 4to piso.

Nombres * (requerido)

E-mail * (requerido)

Teléfono * (requerido)

Mensaje * (requerido)

Enviar

Fuente: Elaboración Propia.

Tabla 4.12.: Valores Límites – Reserva.

Campo	Entrada valida	Entrada invalida
Nombre	Cadena de texto	Cadena alfa numérico
E-mail	Cadena de texto alfanumérico	
Teléfono	Tipo numérico	Cadena de texto
Mensaje	Cadena de texto alfanumérico	

Fuente: Elaboración Propia.

Tabla 4.13.: Prueba de Caja Negra – Reserva.

Entrada	Nombre	Juan
	E-mail	<u>juan@gmail.com</u>
	Teléfono	78646885
	Mensaje	Quisiera reservar un cupo
Salida	Si la información ingresada es correcta se desplegará un mensaje de éxito, caso contrario el sistema despliega los errores cometidos en la inserción.	
Resultados	El sistema valida y verifica la información introducida creando un registro de reserva, y procede a actualizar la tabla de lista de reserva, caso contrario se despliega un mensaje de error	

Fuente: Elaboración Propia.

La prueba de caja negra evidencia que el formulario cumple con la función programada de registros de reserva de cupo.

4.8 CONCLUSIONES Y RECOMENDACIONES

4.8.1 CONCLUSIONES

- **Respecto al proyecto**

Se ha cumplido con el objetivo de desarrollar un Sistema Web para el mejor manejo y administración del Gimnasio Winner y de esa manera poder ofrecer nuestros servicios.

- **Respecto a los objetivos específicos**

- Se logró diseñar una base de datos adecuado y confiable según los requerimientos para el almacenamiento de la información que maneja el gimnasio WINNER.

- Se implementó un módulo para la administración de la información del gimnasio.
- Se logró reducir el tiempo de inscripción de los clientes por parte de la administración del Gimnasio.
- Se realizó la creación de un módulo para el control de información de inscritos por parte de la administración del gimnasio.
- Se realizó un módulo para obtener los reportes e información de clientes inscritos en el gimnasio.
- Se realizó la publicación del sistema web donde el cliente pueda visualizar los productos que ofrece el Gimnasio Winner.

- **Respecto al Objetivo General**

Se logró cubrir con las necesidades de la Radio emisora U.P.E.A., y alcanzar nuestras expectativas ya que se ha terminado su desarrollo contemplando todos los módulos que se ha propuesto en el proyecto.

4.8.2 RECOMENDACIONES.

- **Respecto al Hardware**

Estas recomendaciones se orientan al servidor como al equipo que se dispone en el Gimnasio Winner

- El uso de un UPS (Sistema de alimentación ininterrumpida) también es un factor muy importante debido a las posibles cortes de energía eléctrica.
- El equipo de computación del Gimnasio a futuro debe contar con mejores características (procesador Core i3 o i5, RAM de 4 GB a 8 GB, disco duro de 500 GB o superior).

- **Respecto al Software.**

- Se debe realizar backups mensualmente de la base de datos.
- Se debe hacer revisión a los archivos tanto del servidor web y la base de datos todo esto para que el disco duro del servidor no se colapse.

BIBLIOGRAFÍA

Pressman, S.R. (2002). Ingeniería de software, 5ta edición.

Hernández Jiménez, (2003), Administración de la Función Informática: Una Nueva Profesión autor Ricardo Editorial Limusa.

Arias Chaves, M. (2007). La Ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software. Revista InterSedes Universidad de Costa Rica, 13.

Constanzo, M. A. (2014). Comparacion de modelos de calidad, Factores y Métricas en el Ámbito de la Ingeniería de Software. Rio gallegos: GISP.

Kendall y Kendall, "Análisis y Diseño de Sistemas", Tercera Edición, Editorial Prentice Hall, México, (Kendall y Kendall, 1997).

Laudon K. y Laudon J., "Administración de los Sistemas de Información. Organización y Tecnología", Tercera Edición, Editorial Prentice Hall, México (Laudon K. y Laudon J, 1996.)

Pere Barnola Augé, "Introducción a la creación de páginas web", Universidad Oberta de Catalunya, Barcelona, 2008.

Senn, James A., "ANALISIS Y DISEÑO DE SISTEMAS DE INFORMACION". Segunda Edición, México, Editorial McGrawHill, 1992.

Escalona, M. J., & Koch, N. Ingenieria de Requisitos en Aplicaciones para la Web - Un estilo comparativo. Sevilla España: Universidad de Sevilla (Escalona, M. J., y Koch 2002).

Nieves Guerrero, C. G., Ucan Pech, J. P., & Menendez Dominguez, V. H. (2014). UWE en sistemas de recomendacion de objetos de aprendisaje. Aplicando Ingenieria Web: Un metodo en Caso de Estudio. Revista Latinoamericana de Ingeniera de Software.

P. Rotta, D., S. Pallota, G., E. Klikailo, H., & A. Belloni, E. (2016). Un caso de estudio sobre la aplicación de UWE para la generación de Sistemas Web. Salta, Argentina: Universidad Gastón Dachary.

Research Unit of Programming and Software Engineering, (10 de 18 de 2016), UWE – UML-based Web Engineering. Recuperado de <https://uwe.pst.ifi.lmu.de/index.html>.

GitHub (10 de 07 de 2020) Introducción – Phalcon, Recuperado de
<https://docs.phalcon.io/4.0/es-es/introduction>.

Desarrolloweb.com (30 de 11 de 2020) Html Recuperado de
<https://desarrolloweb.com/home/html>.

4r Soluciones (10 de 07 de 2013) JQuery Mobil framework. Recuperado de
<https://www.4rsoluciones.com/blog/ventajas-y-desventajas-de-jquery-mobile-2/>.

IMECAF (15 de 10 de 12), Php. Recuperado de:
<https://imecaf.com/blog/tag/caracteristicas-de-php/>.

Tonino, J. (10 de 07 de 2020). MariaDB vs MySQL, un Resumen sobre las Tecnologías de Bases de Datos. Obtenido de MariaDB vs MySQL, un Resumen sobre las Tecnologías de Bases de Datos. Recuperado de:

<https://kinsta.com/es/blog/mariadb-vs-mysql/#mysql>.

SlideShareSistemas, F. d. (24 de enero de 2020). Sistemas. Recuperado de:
https://es.slideshare.net/uni_fcys_sistemas/uwe-129633253.

Gestiopolis (05 de 09 de 2017) Administración de la Informática. Recuperado de:
<https://www.gestiopolis.com/concepto-de-administracion-de-la-funcion-informatica-ensayo/>.

ANEXO

1. MANUAL DE USUARIO

1.1. INTRODUCCIÓN

El Sistema de Información Web de Control Administrativo, está pensada y diseñada para facilitar el uso correcto del Sistema del Gimnasio Winner, donde se guardará toda la información, como ser archivos, videos, guía de rutinas, etc. y registro de clientes que tendrá el Gimnasio. Actualmente existen tres roles para acceder al Sistema web: Usuario/Cliente, Usuario/Instructor y Usuario /Administrador.

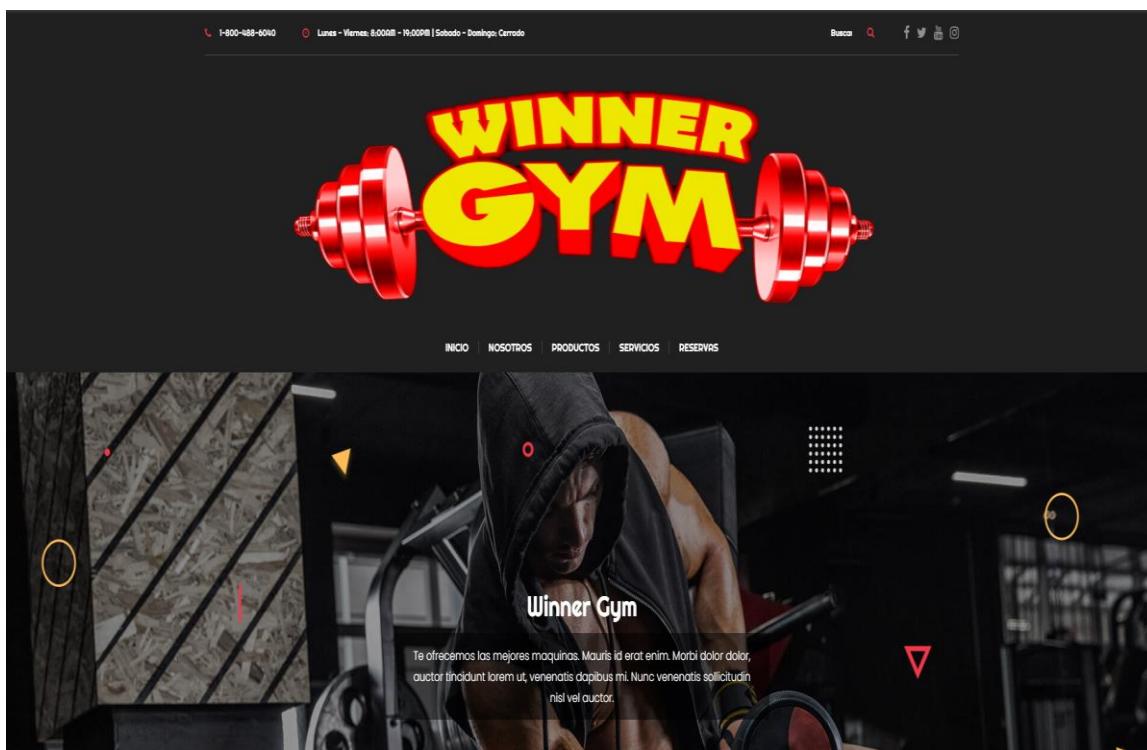
1.2. OBJETIVO DEL SISTEMA

Guia a la administración del Gimnasio Winner, a través de imágenes capturadas del sistema web, hacia el uso correcto del Sistema de Información web de control administrativo.

1.3. GUÍA DE USO

1.3.1. SITIO WEB (FRONT END)

En esta parte podemos ver el sitio web donde contamos con 5 opciones (Nosotros, Productos, Servicios, Reservas e Ingresar).



Nosotros

En esta parte podremos ver la historia como también la misión y visión que tiene el Gimnasio.

The screenshot shows a dark-themed mobile page titled 'NOSOTROS' (About Us). At the top right is a 'Inicio / Nosotros' link. The main content area is titled 'HISTORIA' (History). It contains text about the gym's founding in 2004 and its mission and vision statements. The page features decorative arrows at the top and bottom.

HISTORIA

El gimnasio fitness WINNER fue fundada el 10 de septiembre de 2004, se encuentra ubicada, en la ciudad de La Paz - El Alto Zona, Villa Tunari, AV. Juan pablo II, Edificio San Juan de Dios 4to piso.

Misión: Brindar a nuestros miembros una buena salud física y mental, para ayudarles a alcanzar sus objetivos individuales; con nuestra amplia experiencia les proveemos bienestar en base a un esmerado servicio como un ambiente agradable y con un personal entrenado en los últimos conocimientos disponibles.

Visión: Para el 2021 abrir un nuevo ambiente y ofrecer nuestros servicios a las personas y así poder ser el gimnasio líder de la ciudad, brindando bienestar a nuestros miembros, generando valor a nuestro gimnasio, a nuestros colaboradores y a nuestra comunidad.

Productos

En esta opción el cliente podrá ver todos los productos que podemos adquirir en el gimnasio y así también que beneficio leído al cuerpo el dicho producto.

The screenshot shows a dark-themed mobile page titled 'PRODUCTOS' (Products). At the top right is a 'Inicio / Productos' link. The main content area displays two product images: 'HYDROXYCUT HARDCORE' and 'NEXT GENERATION'. Both products are labeled as 'MUSCLETECH PERFORMANCE SERIES'. The 'HYDROXYCUT HARDCORE' section includes a detailed description of its ingredients and benefits, mentioning Ophiopogon, Amla, and Black Ginger Root. The 'NEXT GENERATION' section also includes a detailed description of its ingredients and benefits, mentioning C. canephora robusta, Coleus Extract, and Caffeine. The page features decorative arrows at the top and bottom.

HYDROXYCUT HARDCORE

Ingrediente clave para la pérdida de peso probado científicamente * Fórmula premium con ophiopogon, amla y raíz de jengibre negro. Fórmula Hardcore sin sobre estimulación I En dos estudios separados, los sujetos que tomaron 200 mg de extracto de café robusta (estandarizado para 45% de ácidos clorogénicos) como se encuentra en HYDROXYCUT HARDCORE® SIGUIENTE GEN NO ESTIMULANTE, perdieron un promedio de 10.95lbs. en 60 días siguiendo una dieta baja en calorías y 3.7 libras. en 8 semanas siguiendo una dieta baja en calorías y realizando ejercicio moderado.

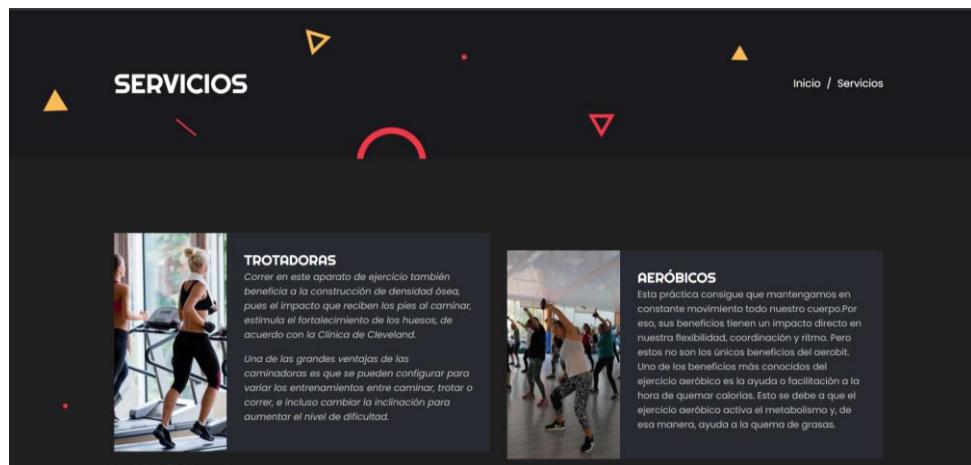
Bs. 230

NEXT GENERATION

FÓRMULA AVANZADA DE PÉRDIDA DE PESO * Los sujetos que tomaron el Ingrediente clave en Hydroxycut Hardcore Next Gen (C. canephora robusta) perdieron 10.95 lbs. en 60 días con una dieta baja en calorías y 3.7 lbs. en un estudio de 8 semanas con una dieta baja en calorías y ejercicio moderado. * OBTENGA LOS HECHOS 400MG C. CANEPhORA ROBUSTA 100MG COLEUS EXTRACT 25MG OPHIOPOGON JAPONICUS 290MG CAFFEINE NEXT GENERATION PÉRDIDA DE PESO Y SENSORIO EXTREMO Hydroxycut Hardcore Next Gen es una fórmula poderosa basada en el ingrediente clave científicamente estudiado C. canephora robusta para ayudar a apoyar la pérdida de peso.

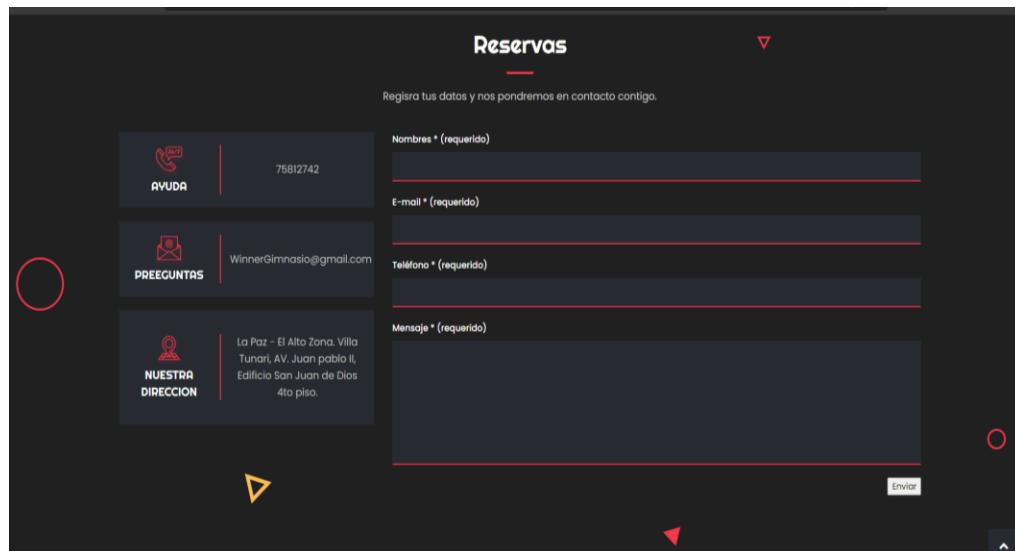
Servicios

Es donde el cliente podrá ver todos los servicios que ofrece el gimnasio y sus beneficios de cada una de ellas.



Reservas

En esta parte el cliente podrá realizar su reserva preguntando todo lo que necesite acerca del gimnasio.



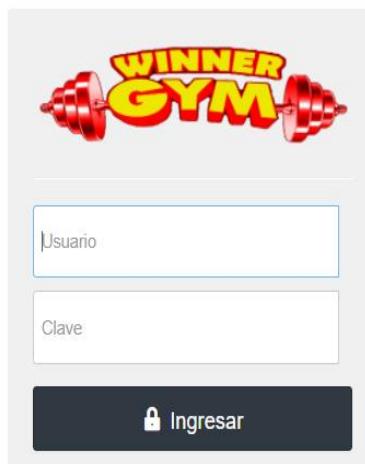
Ingresar

En esta parte el (cliente, Instructor y administrador) una vez obtenido su usuario y contraseña) podrán ingresar al sistema.



1.3.2. INGRESO AL SISTEMA (BACKEND)

El administrador, instructor, usuario deben ingresar su Usuario y contraseña.



Inicio:

Es donde podemos ver los contenidos como ser (nuevos productos, contenidos y categoría) también podemos ingresar al personal donde se puede ver al (administrador, instructor, cliente).

Menú Panel:

Nos ayuda a administrar los módulos dentro del sistema donde podemos añadir nuevos menús.

Asignar Menú:

Una vez creada el menú lo podemos agregar a la barra del sistema

Roles/Cargos:

Podemos asignar roles necesarios que podrán ver los (Usuarios/clientes, instructor y administrador) dentro del sistema.

Configuración del Sitio:

Es importante porque es donde definimos los metadatos, donde podremos configurar los (servidores/puertos/nombres).

Salir:

Una opción que nos permite salir del sistema.



Personal:

Administra todo lo que es el personal administrativo, instructores y Usuario/clientes, también podemos realizar los dichos registros de cada uno de ellos.

Usuarios/Clientes:

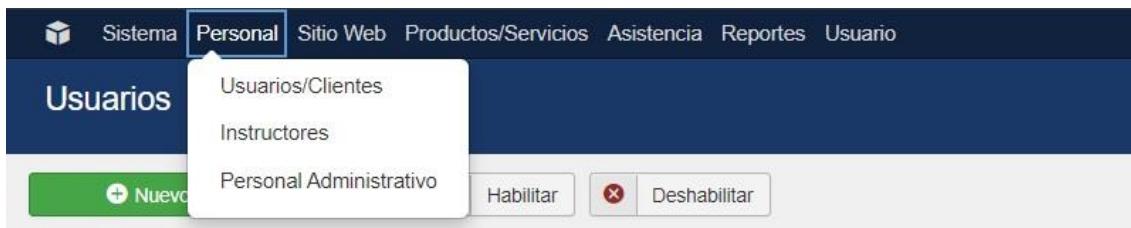
Se podrá ver a todos los Usuarios/Clientes registrados en el gimnasio y así también podemos realizar su dicho registro.

Instructores:

Se podrá ver a todos los Instructores registrados en el gimnasio y así también podemos realizar su dicho registro.

Personal Administrativo:

Se podrá ver a todos los Personal Administrativo registrados en el gimnasio y así también podemos realizar su dicho registro.



Listado de Usuarios						
Nro.	<input type="checkbox"/>	Nombre	Nombre de usuario	Sitio[Area]	Estado	Función
1	<input type="checkbox"/>	Cliente Nuevo Pedro	pedro	Gimnasio[Todos]	Activo	Usuario
2	<input type="checkbox"/>	Estanly admin	estanly	Gimnasio[Todos]	Activo	Administrador

Sitio web:

En este módulo contamos con el siguiente menú (Multimedia, Menús, Categorías, Formulario de Registro Contenidos y Archivos Multimedia).

Multimedia:

En este menú podemos guardar todos nuestros archivos, fotos, videos, etc. y poder subir y administrar nuestro sitio web.

Menús:

En este parte podemos administrar todos nuestros menús de nuestro sitio web.

Categorías:

Es donde administraremos el contenido del sitio web como ser (Información, Productos, Servicios e Inicio).

Formulario de registro:

Una vez que el Usuarios/Clientes realice su reserva en el sitio web en este menú podemos ver el formulario de reserva que realizo el Usuario o cliente.

Contenidos:

Una vez cargada nuestra categoría podemos ver nuestros contenidos y asignar en el Sitio web.

Archivos, Multimedia:

Este menú es especial mente para los Usuarios/Clientes que estén inscritos al gimnasio donde podrá ver todos los archivos, fotos, videos, etc. pero eso si no podrán agregar ni editar ninguno de los contenidos mencionados.

Nro.	?	Nombre Menu	Descripción	Enlace(s) del menu	#Publicado
1	<input type="radio"/>	Menu Principal	Menu principal		4

Productos/Servicios:

En este Modulo Contamos con dos menús importantes para el gimnasio que son (Productos y Servicios) que son muy importantes.

Productos:

Es donde administraremos en el sitio web los productos que ofrece el gimnasio donde pondremos una foto y una pequeña información del producto del producto.

Servicios:

Es donde administraremos en el sitio web los servicios que ofrece el gimnasio donde pondremos el servicio que ofrecemos y un pequeño comentario en que nos beneficia el servicio.

#	<input type="checkbox"/>	Título	Publicado
1	<input type="checkbox"/>	Servicio Completo	✓
2	<input type="checkbox"/>	Trotadoras	✓
3	<input type="checkbox"/>	Spinning	✓
4	<input type="checkbox"/>	Máquinas	✓
5	<input type="checkbox"/>	Aeróbicos	✓

Asistencia:

En esta parte veremos a todos los Usuarios/Clientes activos, como también podremos ver a los Instructores y Personal Administrativo.

Nro.	<input type="checkbox"/>	Nombre	Nombre de usuario	Sitio[Area]	Estado	Funció
1	<input type="checkbox"/>	Cliente Nuevo Pedro	pedro	Gimnasio[Todos]	Activo	Usuario
2	<input type="checkbox"/>	Estanly admin	estanly	Gimnasio[Todos]	Activo	Administrador

Reportes:

En este menú realizaremos la descarga de un pdf mostrando el reporte de todos los Usuario/clientes registrados.

The screenshot shows a software interface with a dark blue header bar containing navigation links: Sistema, Personal, Sitio Web, Productos/Servicios, Asistencia, Reportes (which is highlighted in blue), and Usuario. Below the header, a sub-menu titled 'Reportes' is open, showing the option 'Reporte de Asistencia'. The main content area is titled 'Usuarios/Clientes'. At the top of this area are four buttons: '+ Nuevo', 'Editar', 'Habilitar' (with a green checkmark icon), and 'Deshabilitar' (with a red crossed-out circle icon). Below these buttons is a search bar with the placeholder 'Buscar:' and an 'Aceptar' button. A table lists two users:

Nro.	Nombre	Nombre de usuario
1	Cliente Nuevo Pedro	pedro
2	Estanly admin	estanly

Usuario:

En esta parte tenemos los siguientes menús (Mis Datos Y Cambiar Clave).

Mis Datos:

En esta parte el (Usuario/cliente, Instructor y Personal Administrativo) podrán ver todos sus datos personales.

Cambiar Clave:

En esta parte es donde el (Usuario/cliente, Instructor y Personal Administrativo) podrán realizar el cambio de contraseña.

The screenshot shows a sub-menu under the 'Usuario' tab. The options available are 'Mis datos' and 'Cambiar Clave'. The 'Cambiar clave' option is selected and expanded, showing a form titled 'Cambiar clave'. The form contains four input fields:

- Nombre de Usuario : estanly
- Clave actual: [Empty Input Field]
- Nueva clave : [Empty Input Field]
- Repetir nueva clave : [Empty Input Field]

2. MANUAL TÉCNICO

2.1. INTRODUCCIÓN

El Sistema de Información Web de Control Administrativo, es un sistema diseñada para la web que tiene por objeto realizar las operaciones de Registro y como también Administración del Gimnasio Winner. Cuenta con un sistema de seguridad dinámico a través de la asignación de registros por la administración del Gimnasio.

2.2. OBJETIVOS

Se ha creado dicho documento con el propósito de mostrar cómo fue diseñado el sistema, y al mismo tiempo dar referencias de como interactuar con el sistema para que sea actualizado o al mismo tiempo se le da un mantenimiento adecuado en el caso de un fallo.

2.3. OBJETIVOS ESPECÍFICOS

- Guía de instalación del sistema.
- Requisitos para la ejecución de dicho sistema.

2.4. REQUERIMIENTOS TÉCNICOS

Software

- Navegador web (Firefox, Opera, chrome) para el manejo del sistema web.
- Gestor de base de datos(MySQL) para la administración de los registros almacenados.
- Un emulador de servidor: (XAMPP) para el funcionamiento de la base de datos.

Hardware

- Una computadora completa, esto incluye Mouse, teclado, CPU, etc.
- Como complemento se podría utilizar un equipo touch (computadoras con monitores táctiles para evitar el uso del mouse).
- Conexión a internet via Wifi o Cable Ethernet.

2.4.1. REQUERIMIENTOS MÍNIMOS DE HARDWARE

Procesador. Intel Core 2duo

Memoria RAM (Mínimo): 512mb

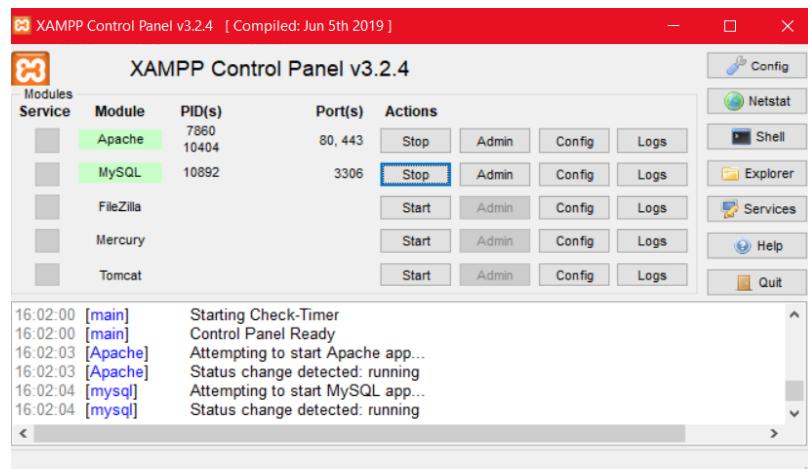
Disco Duro: 64gb

2.4.2. REQUERIMIENTO MÍNIMO DE SOFTWARE

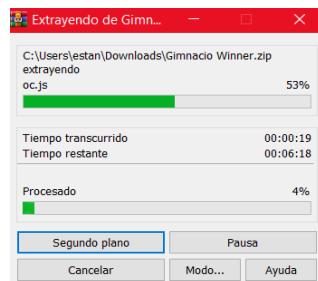
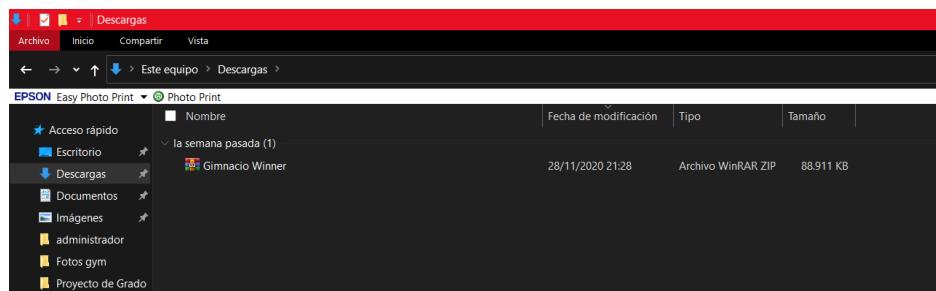
- Privilegios de Administrador: Si
- Sistema operativo: Windows 7 en adelante

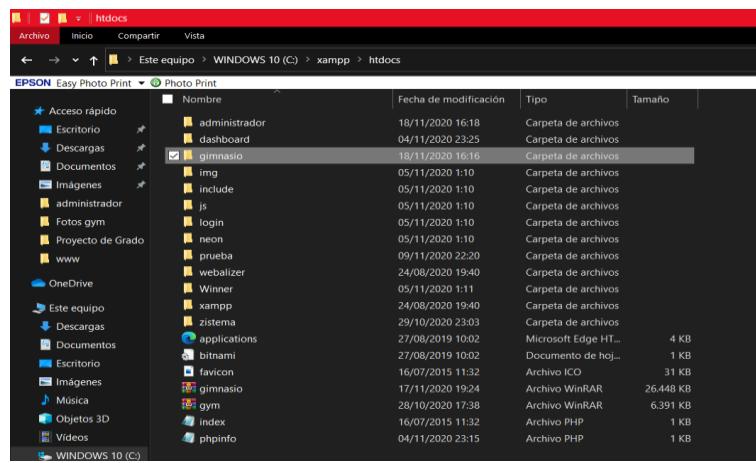
2.5. INSTALACIÓN

Primero que nada, la ejecución la ejecución del servidor XAMPP activando los módulos apache y MySQL.

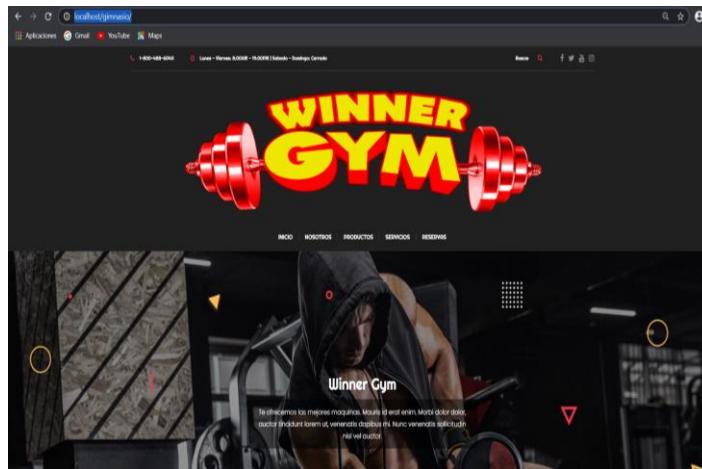


Después buscar la ubicación del archivo comprimido, extraer el archivo comprimido donde viene el sistema y copiar el archivo del sistema en la ruta C:\xampp\htdocs





Seguido de estos pasos escribir en el navegador la línea localhost/gimnasio/. Terminando esto acabaríamos la instalación de nuestro sistema.



ENTREVISTA AL ADMINISTRADOR DEL GIMNASIO WINNER GYM

1. ¿Cuáles son los problemas actualmente del gimnasio WINNER?

R.- se realiza de manera manual todos los procesos y eso causa la pérdida de tiempo

2. ¿Cómo se realiza la venta de sus productos?

R.- De manera presencial para poder explicar el uso como los efectos que se tendrá.

3. ¿Cómo se realiza las inscripciones de los clientes?

R.- De manera manual, lo que ocasiona la pérdida de tiempo

4. ¿Cómo visualizan los horarios?

R.- Esta pegado en el muro de avisos del GYM

5. ¿De qué manera se sabe quiénes son los instructores?

R.- Al momento de la inscripción se les indica que instructores tendrá

6. ¿Cómo realizan reservas en las inscripciones?

R.- De manera presencial en administración

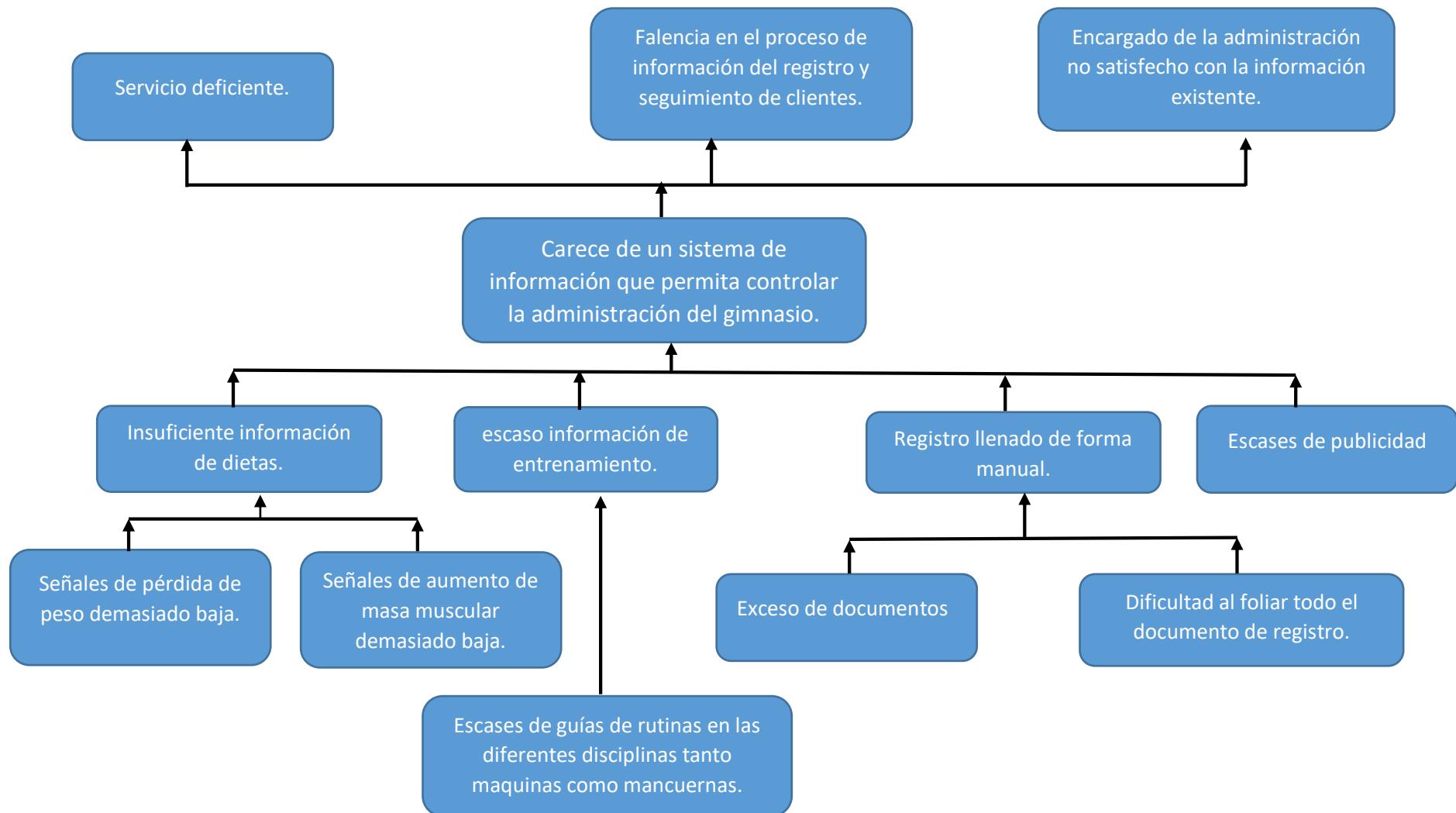
7. ¿De qué manera los clientes o Usuarios pueden seguir las rutinas necesarias para su entrenamiento?

R.- Mediante preguntas al instructor de turno.

8. ¿Cómo los clientes pueden seguir una buena alimentación para la que ma de grasa o aumento de masa muscular?

R.- Según en el entrenamiento que esta realizando con el instructor que le supervisa su entrenamiento y alimentación.

ARBOL DE PROBLEMAS



ARBOL DE OBJETIVOS

