

UNIVERSIDAD PÚBLICA DE EL ALTO

CARRERA INGENIERÍA DE SISTEMAS



PROYECTO DE GRADO

“SISTEMA DE INFORMACIÓN WEB PARA LA ADMINISTRACIÓN Y CONTROL DE MEMBRESÍA MEDIANTE LA TECNOLOGÍA RFID” CASO: ASOCIACIÓN MISIÓN LUZ DE VIDA

**Para Optar al Título de Licenciatura en Ingeniería de Sistemas
MENCIÓN: GESTIÓN Y PRODUCCIÓN**

Postulante: Mary Abigail Chipana Tarqui
Tutor Metodológico: M.Sc. Lic. Ing. Marisol Arguedas Balladares
Tutor Especialista: Ing. Rolando Ulo Mollo
Tutor Revisor: Lic. Gladys Francisca Chuquimia Mamani
EL ALTO - BOLIVIA

2024

DECLARACIÓN JURADA DE AUTENTICIDAD Y RESPONSABILIDAD

Yo **Mary Abigail Chipana Tarqui** con **C.I. 9086358 LP** mediante la presente **declaro** de manera pública que la propuesta del **TRABAJO DE GRADO** titulada **“SISTEMA DE INFORMACIÓN WEB PARA LA ADMINISTRACIÓN Y CONTROL DE MEMBRESÍA MEDIANTE LA TECNOLOGÍA RFID.” CASO: ASOCIACIÓN MISIÓN LUZ DE VIDA** es original, siendo resultado de mi trabajo personal y no constituye una copia o replica de trabajos similares elaborados.

Autorizo la publicación del resumen de mi propuesta en internet y me comprometo a responder a todos los cuestionamientos que se desprenden de su lectura.

Asimismo, me hago responsable ante la universidad o terceros, de cualquiera irregularidad o daño que pudiera ocasionar, por el incumplimiento de lo declarado.

De identificarse falsificación, plagio, fraude, o que el **TRABAJO DE GRADO** haya sido publicado anteriormente; asumo las consecuencias y sanciones que de mi acción se deriven, responsabilizándome por todas las cargas legales que se deriven de ello sometiéndome a las normas establecidas y vigentes de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

El Alto, noviembre de 2024

Mary Abigail Chipana Tarqui

C.I.: 9086358 LP

e-mail: abi.mary579@gmail.com

DEDICATORIA

El presente proyecto va dedicado con mucho cariño a mi familia, a mi Padre: Maximo Ramiro Chipana Tarqui, a mi Madre: Nelly Julia Tarqui de Chipana, quienes han sido mi roca, mi guía y mi inspiración en cada paso de esta larga travesía académica, quienes me han enseñado que el éxito se logra con la ayuda de Dios y el esfuerzo continuo, a mis hermanos Javier Eduardo y Lic. Nilda Rocio quienes me han apoyado incondicionalmente, también a mi sobrina Leydi y mi cuñada Jhovana.

A mi amiga y amigos, por la ayuda constante en el transcurso de nuestro tiempo en la carrera.

AGRADECIMIENTO

Agradezco a Dios por guiarme y darme sabiduría durante toda esta trayectoria.

Agradezco a toda mi familia por apoyarme y estar presente en cada paso de mi vida, por ser ejemplo para seguir adelante y por inculcarme valores, los cuales siempre los tendré presente.

A mi Pastor Caleb y Misionera Esther por darme la confianza de poder realizar mi proyecto en la Asociación Misión Luz de Vida: Iglesia Presbiteriana BET-EL e Iglesia Presbiteriana Luz de Vida, en el cual crecí aprendiendo mucho sobre la palabra de Dios. También agradecer al hermano que me ayudo y a las hermanas quienes me apoyaron en momentos de dificultad y confiaron en mi desempeño para poder desarrollar mis capacidades.

Agradecer a mis distinguidos tutores:

A mi tutor Metodológico M. Sc. Lic. Ingeniera Marisol Arguedas Balladares por haberme asesorado y aconsejado durante la elaboración del proyecto.

A mi tutor Especialista Ing. Rolando Ulo Mollo por los consejos, orientación tiempo, paciencia y confianza que me dio para mejorar las expectativas del presente proyecto.

A mi tutor Revisor Lic. Gladys Francisca Chuquimia Mamani que me guio a la hora de realizar las revisiones del presente proyecto absolviendo mis dudas en todo.

A mis amigos y amigas quienes me brindaron su apoyo incondicional en todo este tiempo. A la carrera de Ingeniería de Sistemas y la Universidad Pública de El Alto por haberme acogido y brindado una formación académica.

Todo lo puedo en Cristo que me fortalece. Filipenses 4:13

INDICE

1. MARCO REFERENCIAL.....	1
1.1 INTRODUCCIÓN.....	1
1.2 ANTECEDENTES.....	3
1.2.1 Antecedentes Institucionales	3
1.2.2 Misión	5
1.2.3 Visión.....	5
1.2.4 Antecedentes a fines al proyecto	5
1.2.4.1 Antecedentes internacionales.....	5
1.2.4.2 Antecedentes Nacionales.....	7
1.2.4.3 Antecedentes Locales.....	8
1.3 PLANTEAMIENTO DEL PROBLEMA	9
1.3.1 Problema Principal.....	10
1.3.2 Problemas Secundarios.....	11
1.4 OBJETIVOS.....	11
1.4.1 Objetivo General.....	11
1.4.2 Objetivos Específicos.....	12
1.5 JUSTIFICACIÓN.....	12
1.5.1 Justificación Técnica.....	12
1.5.2 Justificación Económica.....	13
1.5.3 Justificación Social.....	13
1.6 METODOLOGÍA	13
1.6.1 UWE	13
1.6.2 Método de Ingeniería	14
1.6.2.1 Metodología UWE (UML-BASED WEB ENGINEERRING).....	14
1.6.2.2 Fases o Etapas de la metodología	14
1.7 MÉTRICAS DE CALIDAD	15
1.8 HERRAMIENTAS	15
1.8.1 Hardware	16
1.8.2 Software	16
1.9 LÍMITES Y ALCANCES	19
1.9.1 Limites	19
1.9.2 Alcances.....	20
1.10 APORTES	20
2. MARCO TEÓRICO	21
2.1 INTRODUCCIÓN.....	21

2.2	DEFINICIÓN DE TÉRMINOS	21
2.2.1	Sistema.....	21
2.2.2	Información.....	22
2.2.3	Sistema de información	22
2.2.4	Administración	23
2.2.5	Control.....	24
2.2.6	Membresía.....	24
2.2.7	Tecnología de radio frecuencia RFID.....	25
2.3	INGENIERÍA DE REQUERIMIENTOS	26
2.3.1	Requerimientos funcionales.....	26
2.3.2	Requerimientos no funcionales.....	27
2.4	SOFTWARE	27
2.5	INGENIERÍA DE SOFTWARE	28
2.5.1	Proceso	29
2.5.2	Métodos.....	30
2.5.3	Herramientas	30
2.6	INGENIERÍA WEB.....	30
2.6.1	Etapas de la metodología	31
2.6.2	Características de la Ingeniería web	32
2.7	FASES O ETAPAS DE LA METODOLOGÍA UWE.....	34
2.7.1	Captura, análisis y especificación de requisitos	34
2.7.2	Diseño del sistema.....	35
2.7.2.1	Diagrama de Casos de Uso.....	35
2.7.2.2	Modelo Conceptual.....	36
2.7.2.3	Modelo de Navegación	37
2.7.2.4	Modelo de Presentación	39
2.7.2.5	Modelo de Procesos.....	40
2.7.3	Codificación del Software.....	42
2.7.4	Pruebas	42
2.7.5	La instalación o fase de implementación.....	42
2.7.6	Mantenimiento	42
2.8	ARQUITECTURA CLIENTE-SERVIDOR	42
2.8.1	Características.....	44
2.8.2	Ventajas.....	45
2.8.3	Desventajas.....	46
2.8.4	Seguridad de modelo cliente - servidor.....	47

2.8.5	Arquitectura de dos y tres niveles cliente-servidor	47
2.8.5.1	Arquitectura de dos niveles	47
2.8.5.2	Arquitectura tres niveles	48
2.8.5.3	Comparación entre ambos tipos de arquitecturas	49
2.9	MODELO VISTA CONTROLADOR (MVC)	50
2.9.1	Modelo	51
2.9.2	Vista	51
2.9.3	Controlador	51
2.9.4	Característica del modelo, vista, controlador	52
2.10	MÉTRICAS DE CALIDAD	53
2.10.1	Norma ISO/IEC 25000	53
2.10.2	Funcionabilidad	55
2.10.3	Fiabilidad	55
2.10.4	Usabilidad	56
2.10.5	Portabilidad	56
2.10.6	Mantenibilidad	56
2.10.7	Eficiencia	56
2.10.8	Métricas basadas en la función	57
2.11	MÉTODO DE ESTIMACIÓN DEL SOFTWARE	60
2.11.1	Cocomo II	60
2.12	SEGURIDAD DE APLICACIONES WEB	61
2.12.1	Seguridad del Software ISO-27001	62
2.13	PRUEBAS AL SOFTWARE	64
2.13.1	Prueba de caja negra	64
2.13.2	Pruebas de caja blanca	66
2.14	HERRAMIENTAS	67
2.14.1	Xampp	67
2.14.1.1	Ventajas	67
2.14.1.2	Desventajas	67
2.14.2	MySQL	68
2.14.2.1	Ventajas	68
2.14.2.2	Desventajas	68
2.14.3	PHP	69
2.14.3.1	Ventajas	69
2.14.3.2	Desventajas	70
2.14.4	JavaScript	70

2.14.4.1	Ventajas	71
2.14.4.2	Desventajas	71
2.14.5	HTML	72
2.14.5.1	Ventajas	72
2.14.5.2	Desventajas	72
2.14.6	Css	72
2.14.6.1	Ventajas	72
2.14.6.2	Desventajas	73
2.14.7	Laravel	73
2.14.7.1	Ventajas	74
2.14.7.2	Desventajas	74
2.14.8	Tarjetas RFID pasivas de solo lectura como transpondedor	74
2.14.9	Lector RFID RC522 como transceptor	75
2.14.10	Modulo Node esp8266 como subsistema de procedimientos de datos	76
2.14.11	Software Arduino	78
2.14.12	Librerías Arduino	79
3.	MARCO APLICATIVO	80
3.1	INTRODUCCIÓN	80
3.2	DESARROLLO MODELADO MEDIANTE UWE	80
3.2.1	Captura, análisis y especificación de requisitos	80
3.2.2	Requerimientos Funcionales	82
3.2.3	Requerimientos No Funcionales	83
3.2.4	Diseño del Sistema	85
3.2.4.1	Descripción de actores	85
3.2.4.2	Diagrama de caso de uso general	87
3.2.4.3	Especificaciones de casos de uso	89
3.2.4.4	Diseño conceptual	100
3.2.4.5	Modelo de navegación	101
3.2.4.6	Modelo de presentación	105
3.2.4.7	Modelo de implementación	110
3.2.5	Codificación	117
4.	PRUEBAS DE EVALUACIÓN Y RESULTADOS	124
4.1	MÉTRICAS DE CALIDAD ISO 25001	124
4.1.1	Funcionabilidad	124
4.1.2	Confiabilidad	131

4.1.3	Usabilidad.....	132
4.1.4	Eficiencia	134
4.1.5	Mantenibilidad.....	135
4.1.6	Portabilidad.....	136
4.2	Seguridad del Software ISO 27001.....	138
4.2.1	Seguridad lógica	138
4.2.2	Seguridad física	138
4.3	PRUEBAS DE FUNCIONAMIENTO	139
4.3.1	Prueba de caja blanca	139
4.3.2	Prueba de caja negra.....	141
4.3.3	Pruebas de estrés.....	142
5.	ESTIMACIÓN DE COSTOS.....	143
5.1	COCOMO II	143
6.	CONCLUSIONES Y RECOMENDACIONES	149
6.1	CONCLUSIONES	149
6.1.1	Respecto al proyecto	149
6.1.2	Respecto a los objetivos específicos. Respecto al objetivo general	149
6.2	RECOMENDACIONES.....	151
7.	Referencias Bibliográficas.....	152
ANEXO		156

ÍNDICE DE TABLAS

Tabla 1	<i>Descripción de estereotipos</i>	38
Tabla 2	<i>Descripción de estereotipos</i>	39
Tabla 3	<i>Comparación de arquitecturas</i>	50
Tabla 4	<i>Tabla factor de ponderación</i>	57
Tabla 5	<i>Factor de Escala</i>	58
Tabla 6	<i>Valoración métrica punto función</i>	59
Tabla 7	<i>Captura, análisis y especificación de requisitos</i>	81
Tabla 8	<i>Requerimientos funcionales</i>	82
Tabla 9	<i>Requerimientos no funcionales</i>	84
Tabla 10	<i>Descripción de actores</i>	86
Tabla 11	<i>Descripción de actores de casos de uso Asociación Misión Luz de Vida</i>	89
Tabla 12	<i>Descripción de actores de caso de uso autenticación</i>	91
Tabla 13	<i>Descripción de actores de caso de uso - Ingreso al sistema</i>	92
Tabla 14	<i>Descripción de actores de caso de uso - cierre de sesión</i>	92
Tabla 15	<i>Descripción Caso de uso administrar beneficiario/patrocinio</i>	94
Tabla 16	<i>Descripción caso de uso administrar horario de asistencia beneficiarios/patrocinados</i>	95
Tabla 17	<i>Descripción del Tutor/Maestro</i>	98
Tabla 18	<i>Descripción de beneficiario/patrocinado</i>	100
Tabla 19	<i>Número de entradas de usuarios</i>	125
Tabla 20	<i>Numero de salidas de usuario</i>	125
Tabla 21	<i>Número de peticiones de usuario</i>	126
Tabla 22	<i>Punto función sin ajustar</i>	127
Tabla 23	<i>Factores de ponderación</i>	128
Tabla 24	<i>Factores de Complejidad</i>	129
Tabla 25	<i>Escala de calificación</i>	132
Tabla 26	<i>Evaluación de preguntas para calcula la usabilidad</i>	133
Tabla 27	<i>Escala de valores de eficiencia</i>	134
Tabla 28	<i>Valoración en la eficiencia del sistema</i>	135
Tabla 29	<i>Resultados total de las Métricas de calidad</i>	137
Tabla 30	<i>Aplicación del Modelo Intermedio</i>	143
Tabla 31	<i>Ecuaciones de Cocomo II</i>	144

Tabla 32 <i>Cálculo de atributos de FAE</i>	145
Tabla 33 <i>Resultado total de costos</i>	148
Tabla 34 <i>Tabla comparativa antes y después del Sistema</i>	150

ÍNDICE DE FIGURAS

Figura 1	<i>Requerimientos Funcionales</i>	27
Figura 2	<i>Proceso de Software</i>	29
Figura 3	<i>Modelo de caso de uso de la aplicación portal musical</i>	36
Figura 4	<i>Diagrama Conceptual</i>	37
Figura 5	<i>Modelo de navegación de la aplicación portal musical.</i>	38
Figura 6	<i>Diagrama de presentación</i>	40
Figura 7	<i>Modelo de proceso de la aplicación portal musical.</i>	41
Figura 8	<i>Modelo Cliente-Servidor</i>	44
Figura 9	<i>Arquitectura en dos niveles</i>	48
Figura 10	<i>Arquitectura de tres niveles</i>	49
Figura 11	<i>Modelo Vista Controlador</i>	51
Figura 12	<i>Arquitectura MVC</i>	52
Figura 13	<i>Modelo ISO/ICE 25000</i>	54
Figura 14	<i>Niveles de Cocomo</i>	61
Figura 15	<i>Estructura caja negra</i>	65
Figura 16	<i>Estructura caja blanca</i>	66
Figura 17	<i>Tags de tarjetas de radio frecuencia</i>	75
Figura 18	<i>Modulo lector RFID RC522</i>	76
Figura 19	<i>Modulo Node v2 ESP8266</i>	77
Figura 20	<i>Logo de Arduino</i>	78
Figura 21	<i>Diagrama de caso de uso general</i>	88
Figura 22	<i>Diagrama de caso de Uso - Ingreso al sistema</i>	89
Figura 23	<i>Casos de uso Supervisor/Patrocinio</i>	93
Figura 24	<i>Diagrama de caso de uso – administrar tutor/maestro</i>	97
Figura 25	<i>Caso de uso beneficiario/patrocinado</i>	99
Figura 26	<i>Diseño conceptual</i>	101
Figura 27	<i>Modelo de navegación - Administrador</i>	102
Figura 28	<i>Modelo de navegación - Supervisor/Patrocinio</i>	103
Figura 29	<i>Modelo de navegación - Tutor</i>	104
Figura 30	<i>Modelo de navegación – Beneficiario/Patrocinado</i>	104
Figura 31	<i>Modelo de presentación página inicial del Sistema – Login</i>	105
Figura 32	<i>Modelo de presentación del sistema – Inicio</i>	106

Figura 33 <i>Modelo de presentación – sistemas</i>	106
Figura 34 <i>Modelo de presentación – Reuniones</i>	107
Figura 35 <i>Modelo de presentación del sistema - Pagos</i>	107
Figura 36 <i>Modelo de presentación del sistema - Reportes Asistencias</i>	108
Figura 37 <i>Modelo de presentación del Sistema - Reporte de Cuotas</i>	108
Figura 38 <i>Modelo de presentación del sistema - Lectores</i>	109
Figura 39 <i>Login</i>	110
Figura 40 <i>Menú Inicio</i>	110
Figura 41 <i>Administrar usuarios</i>	111
Figura 42 <i>Menú de Roles</i>	111
Figura 43 <i>Planificar reuniones</i>	112
Figura 44 <i>Genera reportes de Asistencias</i>	112
Figura 45 <i>Módulo de pagos o cobros de cuotas</i>	113
Figura 46 <i>Generar reporte de cuotas</i>	113
Figura 47 <i>Administrar lector</i>	114
Figura 48 <i>Circuito del lector de radio frecuencia</i>	114
Figura 49 <i>Codificación del modelo ESP para envío de datos al sistema</i>	115
Figura 50 <i>Codificación para el envío de datos al sistema</i>	116
Figura 51 <i>Codificación conexión Wifi</i>	117
Figura 52 <i>Controlador de inicio de sesión</i>	118
Figura 53 <i>Codificación del controlador de usuario</i>	119
Figura 54 <i>Codificación del controlador de cuotas</i>	120
Figura 55 <i>Codificación de reporte de Asistencia</i>	121
Figura 56 <i>Codificación usuario</i>	122
Figura 57 <i>Codificación de reporte de asistencia</i>	123
Figura 58 <i>Prueba caja blanca - usuario supervisor/patrocinio</i>	139
Figura 59 <i>Prueba de caja negra - registro de beneficiario/patrocinado</i>	141
Figura 60 <i>Prueba de caja negra - registro de beneficiario/patrocinado</i>	142

RESUMEN

El presente proyecto de grado tiene como objetivo desarrollar un Sistema de Información Web para la administración y Control de Membresía mediante la tecnología RFID. La Asociación Misión Luz de Vida enfrentaba problemas significativos, al no contar con un sistema, el proceso que se realizaba eran los registros de Asistencias en tarjetas físicas que eran selladas al final de cada reunión o cada culto, lo que generaba molestia en los padres y madres de familia al momento de recoger sus respectivas tarjetas. Cada padre o madre de familia de cada beneficiario realiza un pago de 10bs mensualmente el cual es desarrollado manualmente entregando recibos, el cual se pierde información de dichos papeles. El proyecto se utilizó las tecnologías emergentes como ser, MySQL para el almacenamiento de datos a grandes escalas, que de tal manera se puede distribuir los datos en varios nodos para mejorar el rendimiento. Para el desarrollo del software se aplicó la metodología UWE basada en UML, el cual modela las distintas etapas de evolución del nuevo sistema de software. Además, se utilizó herramientas PHP como lenguaje de programación. También se buscó garantizar una buena calidad del sistema adecuado al estándar del ISO 25000, esta que nos proporcionó realizar el funcionamiento de asegurar al cumplimiento de los estándares de calidad. El proyecto se desarrolló con una interfaz dinámica y amigable para tener una buena experiencia intuitiva y fácil de usar para los usuarios que tienen acceso al sistema.

Palabras Clave: ISO 25000, Modelo UWE, Sistema, Membresía.

ABSTRACT

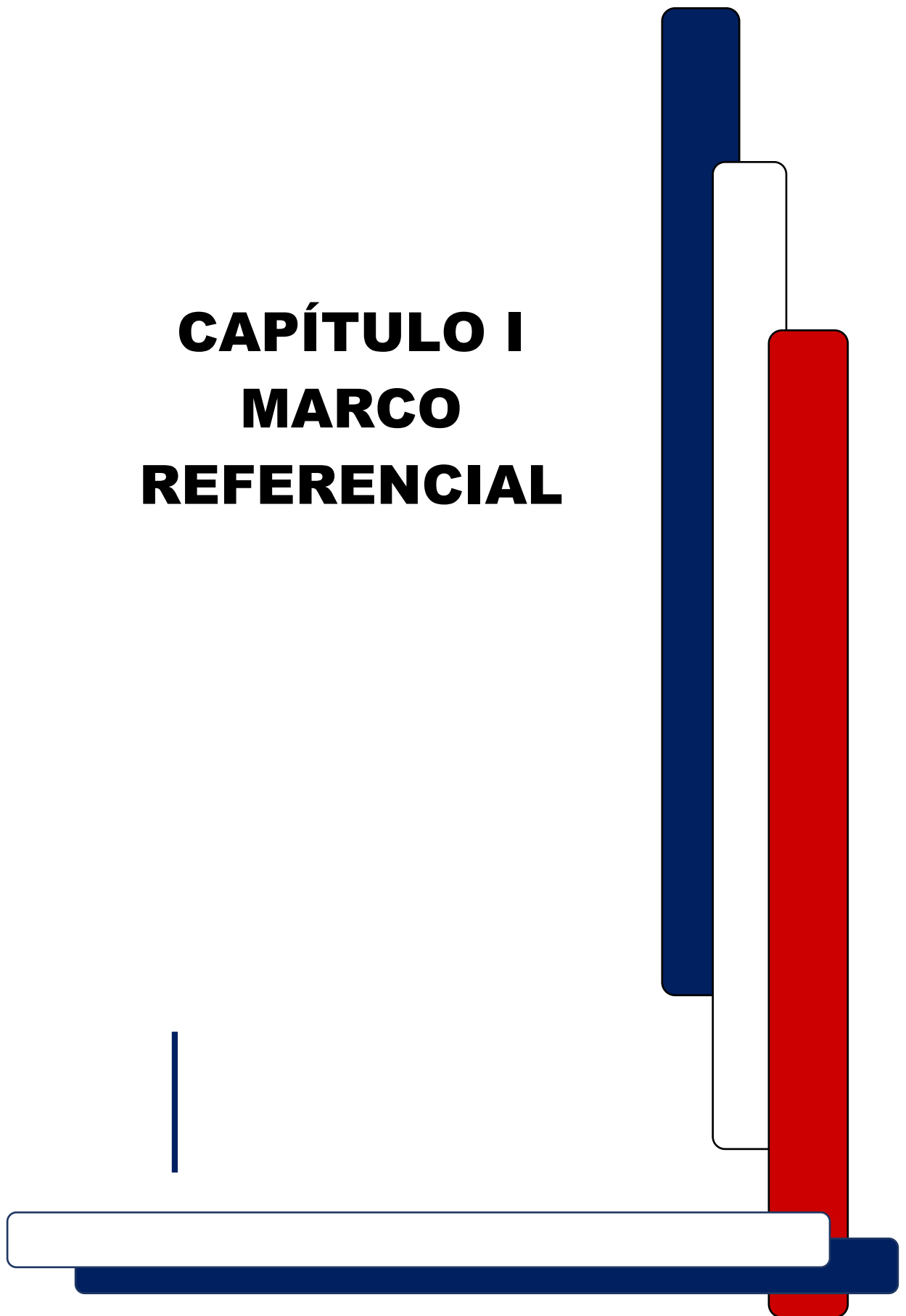
The objective of this degree project is to develop a Web Information System for the administration and Control of Membership using RFID technology. The Light of Life Mission Association faced significant problems, as it did not have a system, the process carried out was attendance records on physical cards that were sealed at the end of each meeting or each worship service, which generated annoyance among parents and mothers of families when collecting their respective cards. Each father or mother of each beneficiary makes a monthly payment of 10bs which is developed manually by submitting receipts, which results in the loss of information from said papers. The project used emerging technologies such as MySQL for large-scale data storage, which can distribute data across multiple nodes to improve performance. For the development of the software, the UWE methodology based on UML was applied, which models the different stages of evolution of the new software system. In addition, PHP tools were used as a programming language. We also sought to guarantee good quality of the system appropriate to the ISO 25000 standard, which allowed us to perform the task of ensuring compliance with quality standards. The project was developed with a dynamic and friendly interface to have a good intuitive and easy-to-use experience for users who have access to the system.

Keywords: ISO 25000, UWE Model, System, Membership.

CAPÍTULO I

MARCO

REFERENCIAL



1. MARCO REFERENCIAL

1.1 INTRODUCCIÓN

La tecnología de RFID identificación por radio Frecuencia como una forma de comunicación inalámbrica que incorpora el uso de acoplamiento electromagnético o electrostático en la porción de radio frecuencia del espectro electromagnético para identificar de manera única un objeto, animal o persona. Actualmente los sistemas se han establecido como herramientas tecnológicas que ordenan y aceleran el proceso de almacenamiento de la información, logrando que sean más rápidos y eficientes en el manejo de grandes cantidades de información, ya que las tareas que eran realizadas manualmente antes ahora son realizadas por medio de sistemas web.

En este documento se describe la problemática que motivo para el desarrollo del proyecto y las razones que lo justifican. La Iglesia Asociación Misión Luz de Vida trabaja con familias de escasos recursos y el proceso de la administración tanto de la membresía ¹y beneficiarios el control de registros se realiza manualmente generando una gran cantidad de hojas de Excel, tarjetas físicas y pérdida de información valiosa como: asistencias a los cultos, reuniones, trabajos y mensualidades o aportes padres en el cual no se puede tener información al instante para realizar las entregas de los beneficios que se les da a cada uno de los beneficiarios o patrocinadores.

El presente proyecto tiene como objetivo principal desarrollar un SISTEMA DE INFORMACIÓN WEB PARA LA ADMINISTRACIÓN Y CONTROL DE MEMBRESÍA MEDIANTE LA TECNOLOGÍA RFID para la Asociación Misión Luz de Vida con el propósito de mejorar la administración de la membresía, brindando información actualizada.

También expresa la metodología utilizada y el plan de trabajo desarrollado. Por lo tanto, el Sistema Web para la Administración de Información y Control de Membresía mediante la

¹ Membresía: Pertenecer o ser parte de la iglesia de Jesucristo.

Tecnología RFID se desarrollará utilizando la metodología de desarrollo UWE (UML-Based Web Engineering, en español Web basada en UML), y las herramientas para el desarrollo a utilizar son: el framework denominado Laravel 9, PHP un lenguaje de programación considerado uno de los más utilizados para el desarrollo de aplicaciones Web ofreciendo seguridad y estabilidad en el funcionamiento, PostgreSQL como el gestor de base de datos, JavaScript para el desarrollo de Frontend, Bootstrap, Arduino, Módulo Node ESP8266, Módulo Lector RFID RC8266 y Tarjetas de Radio Frecuencia, también se hará el uso de arquitectura modelo, Vista, Controlador.

1.2 ANTECEDENTES

Las Iglesias Cristianas funcionan como organizaciones donde centenares de personas participan activamente para rendir culto al Dios todo Poderoso, y gracias a la libertad de culto han logrado darse a conocer públicamente, teniendo como prioridad llevar su mensaje de salvación a la mayor cantidad de personas en el mundo como lo expresa San Mateo 28: 19-20. (Reina Valera, 1960). En las Iglesias Cristianas, al igual que cualquier organización, el éxito o el fracaso depende de varios factores, en su mayoría referidos al recurso humano, esto conlleva la necesidad de almacenar información referente al seguimiento que se realiza con cada persona en particular, es allí donde el manejo de dicha información juega un papel importante. Cada actividad que se realiza genera información que puede apoyar al buen funcionamiento de las distintas tareas que se llevan a cabo. Cualquier empresa que no registra sus actividades se puede ver envuelta en un mismo error constantemente hasta que no se percate de dicho error y lo documente para utilizarlo en situaciones futuras.

1.2.1 Antecedentes Institucionales

La Iglesia Presbiteriana Luz de Vida fue fundada el 6 marzo del 2015, ubicado en la zona de Playa Verde del distrito 7 de la ciudad de El Alto, perteneciente a la red 6 La Paz, el proyecto cuenta con 655 miembros. La Iglesia Presbiteriana Luz de Vida viene trabajando en sociedad con Compassion Internacional hace 7 años. La iglesia inicio como avanzada de la Iglesia Presbiteriana Bet-el Bo-419 misma que viene desde hace 18 años en sociedad con Compassion International.

La iglesia Presbiteriana Luz de Vida realiza el servicio social a través de patrocinio en beneficio de las familias de escasos recursos. Son familias que tienen gran cantidad de miembros y el padre o la madre no pueden solventar los gastos que hay en un hogar. También en algunas familias solo son padres o madres solos que deben de solventar los estudios la alimentación de cada uno de los integrantes de su familia.

La zona no contaba con todos los servicios básicos tales como: agua potable, alcantarillado, internet, áreas verdes, canchas y otros, además no existía centros médicos que puedan atender a la población. Ahora en la actualidad ya cuenta con todos los servicios básicos. Poco después de años fue Asociación Misión Luz de Vida.

Las familias generalmente están conformadas por 7 integrantes con padres jóvenes. La mayoría de los padres se dedican al comercio informal, algunos son choferes asalariados o ayudantes de albañil, las madres se dedican al tejido manual, costura y en su mayoría no tienen un salario digno para sostener sus hogares. También se encuentran madres o padres que son viudos en el cual se esfuerzan para solventar a sus hijos. En algunos casos por la falta de fuentes laborales a causa de la pandemia COVID-19 en las gestiones pasadas tuvieron que volver a sus pueblos para sembrar y cultivar con el propósito de sacar adelante a sus familias. En su tiempo fue un momento muy difícil para las familias, ahora hoy en día se ve poco a poco el progreso de cada familia y con la ayuda del proyecto van sobresaliendo.

Para todo esto la Iglesia Asociación Misión Luz de Vida busca niños, jóvenes que estén escritos en el proyecto para ser seguidores de Jesucristo, el funcionamiento de la Asociación cuenta con asistencias de los padres o madres de familias ya que se busca llegar a cambiar vidas comenzando desde su hogar, formar niños, jóvenes y padres ejemplares para que puedan llevar el evangelio a todo lugar. Las funciones que también se realizan son las reuniones para poder informar aspectos tanto como de la iglesia o comunicar sobre la educación o comportamiento de sus hijos e hijas, se realiza también un aporte mensual para poder complementar y solventar algunos gastos que se realizan en el Proyecto durante cada año como ser: la infraestructura, gastos generales, etc. Cada fin de mes cada beneficiario o patrocinado recibe víveres o productos secos que ayudan a las familias, también 1 vez al año reciben material de higiene, material escolar y algunos beneficios más que ayudan a cada hogar de las familias de escasos recursos. Para el recojo de todos los beneficios cada padre o

madre del beneficiario tiene un control de asistencia y para ello se utilizan tarjetas físicas que son selladas al finalizar de cada reunión o culto, también los cobros de las mensualidades o cuotas que se registran bajo listas en papel o recibos.

1.2.2 Misión

Predicar el evangelio de Jesucristo a las personas de la comunidad moldeando vidas y formando líderes comprometidos para la gloria de Dios. (Mateo 9:35)

1.2.3 Visión

Para el 2024 seremos una iglesia con ministerios y líderes formados teológicamente, con un centro modelo de desarrollo integral, un centro alternativo de formación técnica.

1.2.4 Antecedentes a fines al proyecto

El Sistema de Información Web para la Administración y Control de Membresía aplicando Tecnología RFID busca antecedentes prácticos de metodologías utilizados destacando los siguientes:

1.2.4.1 Antecedentes internacionales

(Torres Mora & Barahona Ávila Kevin Antonio, 2022) "Diseño e implementación de un prototipo RFID para control de inventarios en laboratorio de la UPS". Implementar un prototipo RFID con la finalidad de realizar un control de los equipos electrónicos; multímetros, laptops, osciloscopios que se encuentran en el Laboratorio de Electrónica Analógica de la UPS, mediante una aplicación que de registro de las entradas y salidas de los instrumentos electrónicos en tiempo real. Las herramientas que se utilizó son: etiquetas, antenas, tipos de lectores de RFID, rango de frecuencias y velocidades, tag RFID, Tecnología de autoidentificación, Arduino, Arduino Nano, modulo ESP8266, Modulo RFID. (Universidad Politécnica Salesiana Sede Guayaquil).

(Christian israel lópez mora; & jonathan geovanny ormaza calderón, 2023) “Sistema de inventario basado en tecnología de identificación por radiofrecuencia (rfid) en la carrera de computación de la espam MFL”. Se utilizó la metodología Extreme Programing (XP), basada en cuatro etapas: Planificación, Diseño, Desarrollo, Pruebas. En la etapa de planificación se logró obtener todos los requerimientos necesarios para el perfecto funcionamiento del sistema y se estableció la arquitectura de comunicación entre el hardware y el software, luego, en la etapa del diseño se desarrollaron los diagramas correspondientes de clase y de caso de usos, además se elaboró la arquitectura de comunicación de forma visual , posterior a esto en la etapa de desarrollo se elaboró el backend y fronted en ASP.NET y se estableció la conexión con el lector de identificación por radiofrecuencia. (Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López).

(Luis Armando Castro Vega, 2022)“Implementación de un sistema integral de seguridad mediante tecnología rfid en el laboratorio de computo número catorce en la facultad de ciencias técnicas de la carrera tecnología de la información de la universidad estatal del sur de manabí.” Este sistema tuvo como alcance incrementar la seguridad y el control de acceso dadas las limitaciones de los sistemas de seguridad existentes, elegimos una de las tecnologías más eficaces: la RFID (identificación por radiofrecuencia). La identificación por radiofrecuencia (RFID) es un método de identificación automática que reconoce los objetos mediante una etiqueta especial fijada a ellos de antemano, llamada etiqueta RFID. Las herramientas que se utilizaron son: Bandas de frecuencia, control de acceso, Gigahercios (GHz), Hercio (Hz), infraestructura de comunicación, lector RFID, Tecnología RFID, Radiofrecuencia, sistema de control de acceso, sistema de monitoreo, sistema de seguridad. La metodología utilizada en este proyecto de investigación es cualitativa-cuantitativa con el fin de permitir la recogida de la información mediante datos estadísticos. En este proyecto de investigación se utilizaron

métodos descriptivos, explorativas, deductivos y bibliográficos. (Universidad Estatal Del Sur De Manabí).

1.2.4.2 Antecedentes Nacionales

(Tambo Apaza, 2022)“Diseño de un sistema de monitoreo de rutas del transporte público basado en RFID y microcontroladores PIC para gestionar el transporte público en la zona Extranca de Senkata”. Definir los parámetros técnicos que permitan identificar y procesar las salidas y llegadas, en cada parada de los mini buses, que corresponda a la ruta. Realizar un estudio de frecuencias que sean adecuadas a los horarios de uso cotidiano como ser horas de oficina, escolares y horas pico. Establecer un método de control que verifique si los minibuses cumplen efectivamente, con el horario establecido para cada parada dentro de la ruta que efectúan. Las herramientas se utilizó Microcontroladores (MCU), PIC MCU CCS C Compiler, Grabador de Firmware, Modulo Sim 800L, Lector de RFID-Tag, Arduino, servidor, Python, Lenguaje de programación C++, SQLite. Para el presente sistema se ha procedido a utilizar la Metodología. (Universidad Mayor de San Andrés)

(Luis Gustavo Mamani Apaza, 2022) “Diseño e implementación de un sistema de registro para control de asistencia de docentes y personal administrativo basado en tecnología RFID y reconocimiento facial para la red educativa 203” El presente proyecto tiene la finalidad de diseñar e implementar un sistema de registro de asistencia de docentes y personal administrativo, que además permita generar los reportes de asistencia con tan solo presionar un botón indicando parámetros temporales como el mes y el año, utilizando tecnología RFID y ofreciendo una alternativa de identificación como lo es el reconocimiento facial. Las herramientas que se utilizó son: Identificación por radiofrecuencia RFID, Espectro de radiofrecuencia, comunicaciones inalámbricas, antena RFID, estándar EPC, Reconocimiento facial, lenguaje de programación Python, placas de desarrollo electrónicas, placa de desarrollo Arduino UNO Rev 3, placa de desarrollo Raspberry Pi 3, Red LAN, el Router, cable por

trenzado, conmutador (Switch), servidores. El diseño e implementación del presente proyecto se basa en una adaptación de la metodología de desarrollo basada en componentes, la cual está pensada enteramente para el desarrollo de software. El método de la experimentación es la metodología que nos permitirá evaluar técnicas de desarrollo y corrección de tareas, este método se utilizó juntamente con la metodología basada en componentes. (Universidad Mayor de San Andrés).

(Alba Esther Espejo Velasco, 2021) “Sistema prototipo de registro y seguimiento de activos basado en la tecnología de identificación por radiofrecuencia - RFID” Implementar un sistema prototipo de registro y seguimiento de activos materiales basado en la tecnología de Identificación por Radiofrecuencia RFID para el laboratorio de la carrera de Ingeniería Electrónica de la Universidad Mayor de San Andres. Las herramientas que se utilizó en el presente proyecto son: Componentes básicos de un sistema RFID, Etiquetas RFID, Lector RFID RC522, Raspberry PI3 B+. Clasificación de página: Javascript, CSS, Python, Android Studio, Java, XML, medios inalámbricos. La Metodología que se utilizo es el Proceso Racional Unificado o RUP (por sus siglas en inglés de Rational Unified Process), es un proceso de desarrollo de software, cuyo fin es entregar un producto de software. Se constituye en una de las metodologías estándar más implementadas en sistemas orientados a objetos. Esta metodología proporciona un enfoque disciplinado para asignar tareas y responsabilidades dentro de una organización de desarrollo. Su objetivo es asegurar la producción de software de alta calidad que conoce las necesidades de sus usuarios finales. Esta metodología junto con el Lenguaje Unificado de Modelado (UML) (Universidad Mayor De San Andrés).

1.2.4.3 Antecedentes Locales

(Gladys Claudia Cuevas Coyo, 2020) “Portal web para la gestión de información administrativa y académica” Desarrollar un “Sistema de Información Web para el Control y Seguimiento Académico” para el desenvolvimiento del Estudiante en el año escolar,

cumpliendo todas las exigencias de la Unidad Educativa “Huayna Potosí - Mañana”. Brindando información actualizada al Padre de Familia. Ver Anexo B Árbol de Objetivos. La Metodología que se utilizo es UWE es una metodología basada en el Proceso Unificado y UML para el desarrollo de aplicaciones Web, cubre todo el ciclo de vida de las aplicaciones Web. Su proceso de desarrollo se base en tres frases principales: la fase de captura de requisitos, la fase de análisis y diseño y la fase de la implementación. Las PHP (Programación Orientada a Objetos), Visual Studio Code, CSS3, Ajax, Postman, Laravel 7, Vue, Bootstrap, JQuery, Gestor Base de Datos MySQL-MariaDB. (Universidad Pública de El Alto).

1.3 PLANTEAMIENTO DEL PROBLEMA

La Asociación Misión Luz de Vida a experimentado un crecimiento poblacional debido a un incremento de miembros en la Iglesia. La mayoría de las iglesias no lleva un registro de información sistematizado de sus actividades ni del seguimiento específico de cada miembro, su estado actual o si está cumpliendo con lo que se tiene asignado, solo se tiene información de su vida después de ingresar a la organización, esto gracias a los testimonios narrados por la misma persona.

Cada padre de familia de cada uno de los patrocinados o beneficiarios tiene el deber de asistir a los Cultos con el objetivo de conocer sobre la palabra de Dios para poder tener familias transformadas.

Tiene la responsabilidad de pagar aportes o mensualidades para poder solventar los gastos de la iglesia o del proyecto eso se realiza 1 vez al mes de cada año, se debe de conocer su estado actual o si se está cumpliendo con sus aportes.

Se realizan reuniones con los padres de familias para poder informarles sobre: viajes, salidas de excursiones, el comportamiento de sus hijos e hijas.

Se tiene un control para trabajos que se realizaran en la iglesia como: limpieza, cambiado de ventanas, algunas refacciones del proyecto, etc.

Los registros de Asistencias a los cultos, Cobros de mensualidades, asistencias a reuniones y trabajos se realiza de forma manual, utilizando únicamente tarjetas físicas, aunque existen mínimas excepciones donde se recurre a hojas físicas o en el mejor de los casos a hojas de Excel para llevar el registro y control de los miembros; aun así, la información no está disponible en cualquier momento y desde cualquier lugar, creando limitaciones de disponibilidad. También esta situación provoca conflictos de perdida de información de las tarjetas físicas ya que cada uno de los miembros son responsables de cuidar y ver su asistencia, trabajo, reuniones, aportes o mensualidades, sin tener la información de los registros los encargados ya que son Patrocinio y directora, también se tiene una pérdida de dinero ya que solo se tiene el registro de aportes o mensualidades en hojas de Excel, pérdida de tiempo porque al momento de registrar las asistencias se crean conflictos al momento de entregar las tarjetas físicas ya que numerosos miembros. Si no se tiene todo el registro se les dificulta a cada patrocinado o beneficiarios en no poder recoger los beneficios que les llegan tanto como: regalos, víveres, y otros beneficios.

1.3.1 Problema Principal

El proceso manual de la administración de información y control en la Iglesia Asociación Misión Luz de Vida genera una gran cantidad de hojas de registros, tarjetas físicas y perdida de información valiosa de cada uno de los beneficiarios que es de mucha importancia para realizar la entrega de los beneficios que se le da a cada uno de los patrocinadores o beneficiarios, para poder entregar se necesita conocer su registro de asistencias a cultos, reuniones, trabajos y aportes o mensualidades. Se genera discusiones y demora de tiempo entre madres de familias al momento de registrar sus asistencias, realizar cobros y muchas actividades que se realizan en la Iglesia. Perdida de dinero por motivos de que no se tiene un registro adecuado ya que se

tiene el registro en las tarjetas físicas y como los padres de familias lo tienen muchas veces lo hacen extraviar y solo se tiene el respaldo de recibos o registros de hojas en Excel generando dudas, lo que dificulta la elaboración de informes por gestión de las distintas actividades que se tiene en la Asociación al momento de recibir la ayuda externa de distintos Patrocinadores.

1.3.2 Problemas Secundarios

- El control y registro de asistencia de los miembros de la Asociación Misión Luz de Vida se realiza manualmente utilizando tarjetas de cartulina impresas, para ser selladas.
- Los registros y actualizaciones de los datos se guardan en planillas de Excel (en el cual no se tiene una conexión en la nube que pueda guardar la información para su recuperación en cualquier momento).
- Pérdida de documentos o recibos físicos al momento de realizar cobros de cuotas o mensualidades, lo que ocasiona problemas entre los padres de familia y el personal.
- Demora en la búsqueda de información detallada, lo que genera largas filas al momento de realizar reuniones, actividades extracurriculares y muchos más.

Formulación del problema

¿De qué manera se podrá administrar y controlar las asistencias, aportes o mensualidades y trabajos a cada uno de los miembros de la iglesia y coadyuvar los procesos para su óptimo funcionamiento y atención oportuna?

1.4 OBJETIVOS

1.4.1 Objetivo General

Desarrollar un Sistema de Información Web para la Administración y Control de Membresía mediante la tecnología RFID, para la Asociación Misión Luz de Vida, con el

propósito de mejorar la administración eclesiástica dentro de la iglesia, brindando información actualizada y mejorar la calidad de atención a los miembros de la Asociación.

1.4.2 Objetivos Específicos

- Sistematizar el control y registro de asistencia de los miembros de la Asociación Misión Luz de Vida mediante tarjetas RFID, para optimizar el proceso.
- Diseñar una Base de Datos que permita almacenar datos e información para un mejor manejo adecuado en base a los requerimientos de la institución.
- Implementar el módulo de registro de aportes o mensualidades para un buen manejo adecuado entre las finanzas.
- Generar reportes que faciliten el acceso inmediato a la información de los miembros, al momento de realizar reuniones, actividades extracurriculares y mucho más.

1.5 JUSTIFICACIÓN

De acuerdo a los problemas planteados anteriormente se diseñará un Sistema de Información Web para la Administración y Control de Membresía mediante la Tecnología RFID haciendo el uso de una metodología ágil de desarrollo para realizar el modelado de las recaudaciones y examinar la estructura de la institución para identificar los procesos de administración de control de asistencia.

1.5.1 Justificación Técnica

Para la implementación de este sistema se dispone de distintas tecnologías que son necesarias como ser en la parte de Hardware: equipos de computación con una buena capacidad con un procesador Intel(R) Core(TM) i5-8400 CPU, con una memoria instalada 4,00 GB, conexiones de red, servicio de internet y ambiente adecuado, para poder llevar a cabo el desarrollo del sistema, y en la parte de Software se tienen instalado el sistema operativo de 64

bits, procesador basado en x64, con programas como: Microsoft Office Profesional Plus 2019, PDF, antivirus y otros programas.

1.5.2 Justificación Económica

El software reducirá gastos ya que la información se maneja en grandes volúmenes, si estas son automatizadas implica ahorro de tiempo en la búsqueda de los mismos, reducción de material de escritorio (hojas de papel, cartulinas) y reducción del trabajo manual, generando reducción en gastos de material de escritorio y gastos administrativos. Tanto en el software libre no se adquirirá licencias de pago ya que el proyecto se va a desarrollar con productos de licencia de código abierto, así mismo permitirá fácilmente la elaboración de los informes para obtener la ayuda externa de los Patrocinadores.

1.5.3 Justificación Social

El presente proyecto beneficia a los participantes o beneficiarios (niños, Jóvenes) con diferentes dificultades económicas como intelectuales. Esto facilita al control exhaustivos que se hace con este tipo de ayudas, también facilitará a los Misioneros, los Pastores y al personal de la Asociación Misión Luz de Vida que son: Directora, Patrocinio, tutores/maestros y también a los miembros de la iglesia, se podrá tener un control total lo cual como efecto producirá el crecimiento de la iglesia, que va a permitir buscar soluciones a uno de los problemas, también beneficiará a todos los miembros que asisten a la iglesia tanto niños, jóvenes y padres de familia debido a que pueda reducir el tiempo de espera y minimizar las fallas en el control de asistencia.

1.6 METODOLOGÍA

La metodología que utiliza para el desarrollo de este proyecto es UWE.

1.6.1 UWE

Uwe es una metodología que permite especificar de mejor manera la aplicación web en su proceso de creación, mantiene una notación estándar basada en el uso de UML (Unified

Modeling Language) para sus modelos y sus métodos, lo que facilita la transición. (Guerrero, Pech, & Dominguez, 2014)

1.6.2 Método de Ingeniería

1.6.2.1 Metodología UWE (UML-BASED WEB ENGINEERING)

UWE (UML Web Engineering, en español Ingeniería Web basada en UML) es una metodología que permite especificar de mejor manera una aplicación Web, para el proceso de creación de aplicaciones detalla está, con una gran cantidad de definiciones.

1.6.2.2 Fases o Etapas de la metodología

- **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.
- **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.
- **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.
- **Pruebas:** Las pruebas se utilizan para asegurar el correcto funcionamiento de secciones de código.
- **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.

- **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 MÉTRICAS DE CALIDAD

ISO/IEC 25000, conocida como SQuaRE (System and Software Quality Requirements and Evaluation), es una familia de normas que tiene por objetivo la creación de un marco de trabajo común para evaluar la calidad del producto software.

La familia ISO/IEC 25000 es el resultado de la evolución de otras normas anteriores, especialmente de las normas ISO/IEC 9126, que describe las particularidades de un modelo de calidad del producto software, e ISO/IEC 14598, que abordaba el proceso de evaluación de productos software. Esta familia de normas ISO/IEC 25000 se encuentra compuesta por cinco divisiones.

- ISO/IEC 2500n – División de gestión de calidad
- ISO/IEC 2501n – División de modelo de calidad
- ISO/IEC 2502n – División de medición de calidad
- ISO/IEC 2504n – División de evaluación de calidad

ISO 27001 es la norma internacional que proporciona un marco de trabajo para los sistemas de gestión de seguridad de la información (SGSI) con el fin de proporcionar confidencialidad, integridad y disponibilidad continuada de la información, así como cumplimiento legal.

1.8 HERRAMIENTAS

De acuerdo a las técnicas empleadas se utilizarán las siguientes herramientas que se describen a continuación:

1.8.1 Hardware

- **Computadora**

Una computadora personal, computador personal u ordenador, es un tipo de microcomputadora diseñada en principio para que sea utilizada por una sola persona, la computadora personal tiene que estar equipada para cumplir tareas comunes que le permita navegar por Internet.

- **Tarjetas RFID**

RFID (Identificación por radiofrecuencia) es un sistema de almacenamiento y recuperación de datos remoto que usa dispositivos denominados etiquetas, tarjetas, transpondedores o tags RFID.

- **Lector**

Es un lector de interfaz USB sin controladores es un dispositivo de entrada externo de ordenador de alto rendimiento que puede leer el número de serie de la tarjeta RFID y la salida al ordenador a través del puerto USB de acuerdo con el formato especificado sin procedimientos.

1.8.2 Software

- **Base de Datos**

MySQL es un sistema de gestión de bases de datos relacionales (RDBMS) de código abierto respaldado por Oracle y basado en el lenguaje de consulta estructurado (SQL). MySQL funciona prácticamente en todas las plataformas, incluyendo Linux, UNIX y Windows. MySQL se basa en un modelo cliente-servidor. El núcleo de MySQL es el servidor MySQL, que maneja todas las instrucciones (o comandos) de la base de datos. Está disponible como un programa independiente para su uso en un entorno de red cliente-servidor y como una biblioteca que puede ser incrustada (o enlazada) en aplicaciones independientes. (TechTarget, 2024)

- **Framework de desarrollo**

Laravel es un framework de PHP para ayudarnos en un tipo de desarrollo sobre aplicaciones escritas en este lenguaje de programación, nos ayuda en muchas cosas al desarrollar una aplicación, por medio de sus sistema de paquetes y de ser un framework del tipo MVC (Modelo-VistaControlador), cómo instanciar clases y métodos para usarlos en muchas partes de nuestra aplicación sin la necesidad de escribirlo y repetirlos muchas veces con lo que eso conlleva a la hora de modificar algo en el código, usa muchas herramientas o utilidades con distintos fines y podríamos casi escribir un artículo o varios por cada una de ellas y hacer que este artículo casi fuera interminable gracias a este compañero que tantas posibilidades nos permite. (Herrera Rivas, 2022)

- **Framework de Diseño**

Visual Studio Code Es un editor de texto distribución del code–repositorio con personalizaciones específicas de Microsoft publicadas bajo una licencia de producto tradicional de Microsoft. Visual Studio Code combina la simplicidad de un editor de código con lo que los desarrolladores necesitan para su ciclo principal de edición, compilación y depuración de código junto con depuración liviana, un modelo de extensibilidad enriquecido e integración liviana con herramientas existentes. De acuerdo a la información proporcionada por la página web Vulgarin Matamoros, J. C., & Tuaza Burgos, A. R. (2022) es un editor de código fuente que permite trabajar con diversos lenguajes de programación, admite gestionar tus propios atajos de teclado y refactorizar el código. sus extensiones nos otorgan infinidad de opciones, como colorear tabulaciones, etiquetas o recomendaciones de autocompletado. Hay extensiones que nos ayudan con el lenguaje de programación que vayamos a usar, como por ejemplo para Python, C / C++, JavaScript y demás.

Bootstrap es uno de los frameworks más populares para el desarrollo del frontend de páginas web. La herramienta proporciona plantillas para CSS y HTML que facilitan la colocación y el diseño de la página.

- **JQuery**

Jquery conocido como JavaScript, es un software libre y de código abierto que posee un doble licenciamiento bajo la licencia MIT y la Licencia Publica General de GNUv2, permitiendo su uso en proyectos libres y privativos. Es compatible con los navegadores Mozilla Firefox 2.0+, Internet Explorer 6+, Safari 3, Opera 10.6+ y Google Chrome 8+. Boumediane, M. E. A. (2012).

- **PHP**

PHP según la información proporcionada por la página web Pardo, M. R. V., Tapia, J. A. H., Moreno, A. S. G., & Sánchez, L. F. V. (2018). (acrónimo de PHP: Hypertext Preprocessor), es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor. Es un lenguaje multiplataforma. Completamente orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una Base de Datos. Es libre, por lo que se presenta como una alternativa de fácil acceso para todos y permite aplicar técnicas de programación orientada a objetos.

- **Servidor Apache**

Xampp

Es un servidor web HTTP de código abierto para la creación de páginas y servicios web, es un servidor multiplataforma, una de las ventajas más grandes de Apache es que es un servidor web multiplataforma, gratuito muy robusto y que destaca por su seguridad y rendimiento.

- **JavaScript**

El código de programación de JavaScript se ejecuta en los navegadores, ya sean de escritorio o móviles, ya sean Android o iPhone. Sirve para exactamente lo mismo, da igual en el tipo de dispositivo que se ejecute el navegador.

JavaScript es el lenguaje de programación encargado de dotar de mayor interactividad y dinamismo a las páginas web. Cuando JavaScript se ejecuta en el navegador, no necesita de un compilador. El navegador lee directamente el código, sin necesidad de terceros. Por tanto, se le reconoce como uno de los tres lenguajes nativos de la web junto a HTML (contenido y su estructura) y a CSS (diseño del contenido y su estructura).

- **CSS**

CSS son las siglas en inglés para hojas de estilo en cascada (Cascading Style Sheets). Básicamente, es un lenguaje que maneja el diseño y presentación de las páginas web, es decir, cómo lucen cuando un usuario las visita. Funciona junto con el lenguaje HTML que se encarga del contenido básico de las páginas.

- **Arduino**

Es una plataforma de hardware libre, nos referimos a que cualquier persona puede crear y modificar sus propias placas a partir de una misma base y puede darle el uso que quiera libremente, de manera que con una de estas placas se pueden crear diferentes tipos de microordenadores.

1.9 LÍMITES Y ALCANCES

A continuación, se describen los límites y alcances del proyecto de grado:

1.9.1 Límites

- El presente proyecto analizara la situación actual y de acuerdo a las necesidades específicas se desarrollarán en la Asociación Misión Luz de Vida.
- El sistema no contemplara la biblioteca de la Iglesia.

- El sistema beneficiara solo a los miembros de la Asociación Misión Luz de Vida.

1.9.2 Alcances

Con los alcances del presente proyecto, se pretende reducir de manera eficaz la posible pérdida de datos relacionados con los nuevos asistentes a la Iglesia y los miembros bautizados.

El presente sistema considerara los siguientes alcances:

- Para el control de asistencia se utilizará tarjetas RFID.
- Módulo Administrador, podrá registrar los datos personales y se otorgarán roles y permisos a los usuarios.
- Módulo de registro de nuevos miembros, donde se realizará los registros con las tarjetas RFID.
- Módulo de registro de mensualidades o aportes.
- Módulo de registro, actualización y consulta de pedido de oración.
- Módulo de registro, actualización y consulta de reuniones/avisos.
- Módulo de reportes.

1.10 APORTES

El Sistema de Información Web para la Administración y Control de Membresía aplicando Tecnología RFID, permitirá administrar la información de los miembros de la Asociación Misión Luz de Vida, proponiendo además mejorar la calidad gracias al uso de la tecnología.

Ayudar con un sistema de seguimiento de actividades que almacene información histórica de la gestión actual y las futuras gestiones para el acceso oportuno a la información y la constancia del personal administrativo.

CAPITULO II

MARCO TEÓRICO

**INGENIERIA
DE SISTEMAS**
UNIVERSIDAD PUBLICA DE EL ALTO

2. MARCO TEÓRICO

2.1 INTRODUCCIÓN

En este capítulo del marco teórico se darán a conocer los conceptos básicos del presente proyecto. sistema de información web para la administración y control de membresía mediante la tecnología RFID. Así mismo para su completo entendimiento se encontrará los conceptos básicos, complementarios y específicos con las distintas herramientas utilizadas y métodos, con una descripción detallada de cada uno de los elementos que sirvieron para realizar el presente proyecto.

Para su mejor comprensión del título del proyecto, se tiene como referencia las siguientes definiciones y conceptos con el fin de dar soluciones a los problemas presentados.

2.2 DEFINICIÓN DE TÉRMINOS

2.2.1 Sistema

Un sistema se define como " una unión de partes o componentes, conectados en una forma organizada. Las partes se afectan por estar en el sistema y se cambian si lo dejan. La unión de partes hace algo (muestra una conducta dinámica como opuesto a permanecer). Además, un sistema puede existir realmente como un agregado natural de partes componentes encontradas en la naturaleza, o ésa puede ser un agregado inventado por el hombre, una forma de ver el problema que resulta de una decisión deliberada de suponer que un conjunto de elementos está relacionado, y constituyen una cosa llamada "un sistema". (Gigch Van, 1978)

Un sistema es un elemento completo donde cada uno de sus componentes se vincula con otro componente, tanto de forma material como conceptual. Los métodos tienen composición, organización y dominio, pero únicamente los sistemas materiales tienen mecanismos y solamente algunos de ellos tienen figura o configuración. (Martinez, 2023)

El sistema es un conjunto de datos ordenados, conforme a una serie de instrucciones o algoritmos o es un conjunto de elementos o componentes interrelacionados entre sí para lograr un objetivo o realizar una función específica.

2.2.2 Información

La información está constituida por un grupo de datos ya supervisados y ordenados, que sirven para construir un mensaje basado en un cierto fenómeno o ente. La información permite resolver problemas y tomar decisiones, ya que su aprovechamiento racional es la base del conocimiento. (Pérez Porto & Gardey, 2021b)

Información "es un conjunto de datos con un significado, o sea, que reduce la incertidumbre o que aumenta el conocimiento de algo. En verdad, la información es un mensaje con significado en un determinado contexto, disponible para uso inmediato y que proporciona orientación a las acciones por el hecho de reducir el margen de incertidumbre con respecto a nuestras decisiones" (Chiavenato, 2006)

La información es un conjunto de datos que están organizados y procesados tienen como objetivo de administrar y distribuir datos de manera organizada.

2.2.3 Sistema de información

Un sistema de información (SI) automático o manual, que comprende persona, maquinas y/o métodos organizados para agrupar, procesar, transmitir y diseminar datos que representan información para el usuario. (Gonzalez Longatt, p.1)

Un sistema de información es un conjunto de elementos interrelacionados con el propósito de prestar atención a las demandas de información de una organización, para elevar el nivel de conocimientos que permitan un mejor apoyo a la toma de decisiones y desarrollo de acciones.(Peña, 2006)

Los sistemas de información contienen información sobre personas, lugares y cosas importantes dentro de la organización, o en el entorno que la rodea. Por información nos referimos a los datos que se han modelado en una forma significativa y útil para los seres humanos. Por el contrario, los datos son flujos de elementos en bruto que representan los eventos que ocurren en las organizaciones o en el entorno físico antes de ordenarlos e interpretarlos en una forma que las personas puedan comprender y usar. (Ramírez Ivan, 2019)

El sistema de Información es un conjunto de elementos ordenados de personas, procesos y herramientas que se interactúan y permiten que la información esté disponible para satisfacer necesidades cuyo fin es administrar datos e información.

2.2.4 Administración

“La administración es un proceso a través del cual se coordinan y optimizan los recursos de un grupo social con el fin de lograr la máxima eficacia, eficiencia, calidad, productividad y competitividad en la consecución de sus objetivos, para lograr la máxima eficiencia” (Münch Galindo, 2018)

La administración es un proceso continuo y sistemático que implica una serie de actividades impulsoras, como planear, organizar, dirigir y controlar recursos y competencias para alcanzar metas y objetivos organizacionales. Además, incluye fijar los objetivos que se quieren alcanzar, tomar decisiones en el camino para alcanzarlos y liderar todo ese proceso con el fin de alcanzarlos y ofrecer resultados a los públicos estratégicos de la organización. (Chiavenato Idalberto, 2019)

De acuerdo con la definición de Idalberto Chiavenato, la administración es el proceso de planear, organizar, dirigir y controlar el uso de los recursos para lograr los objetivos organizacionales. Por lo tanto, la administración es imprescindible para la existencia, la supervivencia y el éxito de las organizaciones. Sin la administración, las organizaciones no tendrían las condiciones para existir y crecer. (Chiavenato, 2007)

La administración permite la ejecución de actividades, también en red implica organizar, instalar y mantener en funcionamiento los sistemas informáticos

2.2.5 Control

El control puede definirse como "el proceso de regular actividades que aseguren que se están cumpliendo como fueron planificadas y corrigiendo cualquier desviación significativa"(Robbins, 1996, p. 654)

El concepto de control engloba toda la administración de una empresa, pero según Idalberto Chiavenato considera que el "principio de control: controlar el trabajo para cerciorarse de que está ejecutándose de acuerdo con las normas establecidas y según el plan previsto. La gerencia debe cooperar con los empleados para que la ejecución sea la mejor posible".(Chiavenato, 2004)

El control tiene como objetivo cerciorarse de que los hechos vayan de acuerdo con los planes establecidos. (Scalan, 1999)

Control, consiste en controlar el trabajo o regular las actividades y asegurar que se están cumpliendo.

2.2.6 Membresía

La membresía de la iglesia se refiere a pertenecer o ser parte de la iglesia de Jesucristo. Involucra diferentes conceptos de la iglesia, requisitos de ingreso, responsabilidades y privilegios de la membresía. (Allison Gregg, 2012)

Membresía es una adaptación del vocablo inglés membership. La noción, que también puede escribirse como membrecía según lo aceptado por el diccionario de la Real Academia Española (RAE), se emplea en el continente americano para aludir a la condición de integrante de una agrupación u organización. Contar con una membresía de una cierta de entidad, por lo tanto, equivale a ser miembro de la misma. La idea refiere a un grupo específico de individuos

que comparten determinados privilegios y derechos, los cuales no están disponibles para quienes se encuentran afuera del conjunto en cuestión.(Pérez Porto & Gardey, 2021c)

La membresía de la iglesia es una relación formal entre la iglesia y el cristiano, caracterizada por la confirmación y la supervisión eclesial del discipulado del cristiano, y la sumisión del cristiano para vivir su discipulado bajo el cuidado de la iglesia. (Leeman, 2013)

La membresía es un conjunto de personas que tienen relación con la Iglesia y el cristianismo.

2.2.7 Tecnología de radio frecuencia RFID

RFID (Identificación por Radiofrecuencia) es un método de almacenamiento y recuperación remota de datos, basado en el empleo de etiquetas o “tags” en las que reside la información. RFID se basa en un concepto similar al del sistema de código de barras; la principal diferencia entre ambos reside en que el segundo utiliza señales ópticas para transmitir los datos entre la etiqueta y el lector, y RFID, en cambio, emplea señales de radiofrecuencia (en diferentes bandas dependiendo del tipo de sistema, típicamente 125 KHz, 13,56 MHz, 433-860-960 MHz y 2,45 GHz).(Portillo García et al., 2008)

La RFID (identificación por radiofrecuencia) es una tecnología empleada en todas las áreas de captura automática de datos, que permite la identificación sin contacto físico de objetos mediante radiofrecuencia (RF). Sus aplicaciones actuales abarcan desde sistemas industriales automatizados, control de acceso, identificación de animales y pasaportes electrónicos hasta aplicaciones médicas, emisión de billetes y seguimiento de existencias. (Canestri, 2007)

La identificación por radiofrecuencia es una tecnología de captura e identificación automática de información contenida en etiquetas electrónicas (tags). Cuando estas etiquetas entran en el área de cobertura de un lector RFID, éste envía una señal para que la etiqueta le

transmita la información almacenada en su memoria, habitualmente un código de identificación.(Huidobro, n.d.)

La tecnología de identificación por radio frecuencia es un sistema de almacenamiento y recuperación de datos que se usan dispositivos llamados etiquetas, tarjetas o transpondedores RFID.

2.3 INGENIERÍA DE REQUERIMIENTOS

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 de que se transforman en un sistema funcional. Incluye siete tareas diferentes: concepción, indagación, elaboración, negociación, especificación, validación y administración.(R. Pressman, 2010)

“La ingeniería de requerimientos es el proceso de desarrollar una especificación de software. Las especificaciones pretenden comunicar las necesidades del sistema del cliente a los desarrolladores del sistema”. (Sommerville, 2005)

“Ingeniería de Requerimientos ayuda a los ingenieros de software a entender mejor el problema en cuya solución trabajarán. Incluye el conjunto de tareas que conducen a comprender cuál será el impacto del software sobre el negocio, qué es lo que el cliente quiere y cómo interactuarán los usuarios finales con el software”. (R. S. Pressman, 2006)

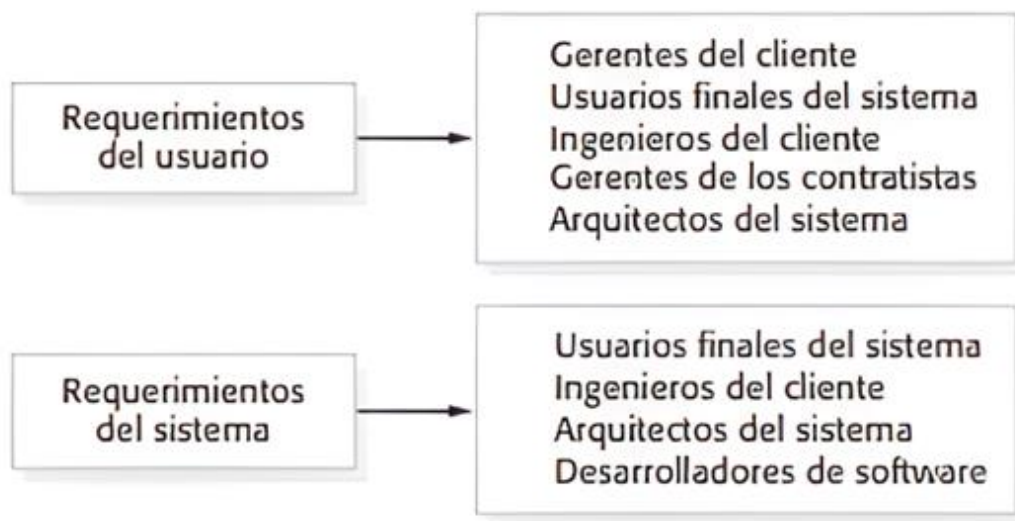
2.3.1 *Requerimientos funcionales*

Los requerimientos funcionales para un sistema refieren lo que el sistema debe hacer. Tales requerimientos dependen del tipo de software que se esté desarrollando, de los usuarios esperados del software y del enfoque general que adopta la organización cuando se escriben los requerimientos. (Sommerville)

Son enunciados acerca de servicios que el sistema debe proveer, de cómo debería reaccionar el sistema a entradas particulares y de cómo debería comportarse el sistema en situaciones específicas. En algunos casos, los requerimientos funcionales también explican lo que no debe hacer el sistema.

Figura 1

Requerimientos Funcionales



Nota. Lectores de diferentes tipos especificación de requerimientos

2.3.2 *Requerimientos no funcionales*

Los requerimientos no funcionales, como indica su nombre, son requerimientos que no se relacionan directamente con los servicios específicos que el sistema entrega a sus usuarios. Pueden relacionarse con propiedades emergentes del sistema, como fiabilidad, tiempo de respuesta y uso de almacenamiento. De forma alternativa, pueden definir restricciones sobre la implementación del sistema, como las capacidades de los dispositivos I/O o las representaciones de datos usados en las interfaces con otros sistemas.

2.4 SOFTWARE

El software es el conjunto de instrucciones y datos en formato binario almacenados en la memoria principal, que le indica a una computadora que debe hacer y como, es decir, el

software dirige al hardware el software es la parte lógica del sistema informático.(Castellano, 2001)

El software de computadora es el producto que construyen los programadores profesionales y al que después le dan mantenimiento durante un largo tiempo. Incluye programas que se ejecutan en una computadora de cualquier tamaño y arquitectura, contenido que se presenta a medida de que se ejecutan los programas de cómputo e información descriptiva tanto en una copia dura como en formatos virtuales que engloban virtualmente a cualesquiera medios electrónicos. La ingeniería de software está formada por un proceso, un conjunto de métodos (prácticas) y un arreglo de herramientas que permite a los profesionales elaborar software de cómputo de alta calidad.(Pressman R, 2010b)

El software es un conjunto de elementos lógicos necesarios para que se pueda realizar las tareas encomendadas al mismo, se puede definir de la siguiente forma: es la parte lógica que dota al equipo físico de capacidad para realizar cualquier tipo de trabajo.(Castellano Casas, 2001)

2.5 INGENIERÍA DE SOFTWARE

La ingeniería de software es una tecnología de varias capas las mismas que deben asegurar la calidad del producto, El fundamento para la ingeniería de software es la capa proceso. 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 R, 2010a)

Ingeniería del software es el establecimiento y uso de principios sólidos de ingeniería, orientados a obtener software económico que sea fiable y trabaje de manera eficiente en

máquinas reales. Fritz Bauer, First NATO Software Engineering Conference, Garmisch (Germany, 1968)

La ingeniería de software es el estudio de los principios y metodologías para el desarrollo y mantenimiento de sistemas de software. (Zelkovitz, 1978)

La ingeniería de software es la aplicación sistemática de conocimientos científico para la construcción de soluciones, aplica principios propios de la ciencia. La ingeniería de software no se limita a aplicar solo principios de la ciencia de computación y las matemáticas sino cualquier principio de que pueda sacar ventajas. El fundamento en el que se apoya la ingeniería de software es el compromiso con la calidad.

Figura 2

Proceso de Software



Nota. Elaboración propia

2.5.1 Proceso

Los métodos de la ingeniería del software indican «cómo» construir técnicamente el software. Los métodos abarcan una gran gama de tareas que incluyen análisis de requisitos, diseño, construcción de programas, pruebas y mantenimiento. Los métodos de la ingeniería del software dependen de un conjunto de principios básicos que gobiernan cada área de la tecnología e incluyen actividades de modelado y otras técnicas descriptivas.

2.5.2 Métodos

El proceso define un marco de trabajo para un conjunto de Areas clave de proceso (ACPs) que se deben establecer para la entrega efectiva de la tecnología de la ingeniería del software. Las áreas claves del proceso forman la base del control de gestión de proyectos del software y establecen el contexto en el que se aplican los métodos técnicos, se obtienen productos del trabajo (modelos, documentos, datos, informes, formularios, etc.), se establecen hitos, se asegura la calidad y el cambio se gestiona adecuadamente.

2.5.3 Herramientas

Las herramientas de la Ingeniería del software proporcionan un enfoque automático o semi-automático para el proceso y para los métodos. Cuando se integran herramientas para que la información creada por una herramienta la pueda utilizar otra, se establece un sistema de soporte para el desarrollo del software llamado ingeniería del software asistida por computadora (CASE).

2.6 INGENIERÍA WEB

Ingeniería Web es la aplicación sistemática, disciplinada y cuantificable de enfoques para el desarrollo, operación y mantenimiento de las aplicaciones basadas en la Web. Es a la vez un enfoque pro-activo y una creciente colección teórica y empírica para la investigación en el desarrollo de aplicaciones Web.

Según (Villezcas, 2011)“la ingeniería web es la aplicación de metodologías sistemáticas, disciplinadas y cuantificables al desarrollo eficiente, operación y evolución de aplicaciones de alta calidad en la worl wide web”.

Por definición la Ingeniería Web para (Murgesan, 2022)“La aplicación de una aproximación sistemática, disciplinada y cuantificable al desarrollo, operación y mantenimiento de aplicaciones basadas en la Web o la aplicación de la ingeniería al software basado en la Web”. (p. 34)

Según del (Valle, 2009) “La Ingeniería Web (IWeb) aplica “sólidos principios científicos, de ingeniería y de administración, y enfoques disciplinados y sistemáticos para el desarrollo, despliegue y mantenimiento exitoso de sistemas y aplicaciones basados en Web de alta calidad”.

2.6.1 Etapas de la metodología

- **Formulación:** Consiste en identificar las metas, y los objetivos del sistema, constituyendo de esta manera el motivo del progreso del sistema, su importancia y los usuarios potenciales.
- **Planificación:** Consiste en el cálculo del costo integral del proyecto y se determinan las amenazas que se relacionan con el impulso del desarrollo además se determina un plan muy detallado para el desarrollo y progresos de la aplicación.
- **Análisis:** Consiste en establecer los requerimientos tecnológicos y de diseño y el reconocimiento de los fundamentos del contenido que se va a agregar. Esta etapa consta de cuatro análisis diferentes: Análisis del contenido, análisis de la interacción, análisis funcional y análisis de la configuración.
- **Ingeniería:** Esta etapa consiste en la realización de diseños tanto del tema como el de la fabricación, en paralelo con los diseños arquitectónicos, navegación e interfaz.
- **Generación de páginas:** Esta etapa consiste en la realización de la estructura, empleando herramientas para el desarrollo de aplicaciones web. Está relacionado con el diseño arquitectónico, de navegación y de interfaz para la elaboración de una web dinámica.
- **Pruebas:** Esta etapa sirve para encontrar las fallas y permite garantizar que la aplicación web funcione perfectamente en distintos campos, utilizando tácticas y tecnologías que son sugeridas para otros sistemas.

- **Evaluación del cliente:** En este punto, se efectúan todas las modificaciones y variaciones que se encontraron en la etapa de pruebas y se incorporan al sistema para el siguiente incremento, de tal modo que se asegure la satisfacción por parte del cliente, según los requerimientos solicitados.

Entonces la ingeniería de la web es la aplicación de metodologías sistemáticas, disciplinadas y cuantificables al desarrollo eficiente, operación y evolución de aplicaciones de alta calidad en la World Wide Web (www). En este sentido la ingeniería de la web hace referencia a las metodologías, técnicas y herramientas que se utilizan en el desarrollo de aplicaciones web complejas y de gran dimensión en las que se apoya la evaluación, diseño, desarrollo, implementación y evolución de dichas aplicaciones.

2.6.2 Características de la Ingeniería web

La ingeniería de la Web no es un clon o subconjunto de la ingeniería de software, aunque ambas incluyen desarrollo de software y programación, pues a pesar de que la ingeniería de la Web utiliza principios de Ingeniería de Software, incluye nuevos enfoques, metodologías, herramientas, técnicas, guías y patrones para cubrir los requisitos únicos de las aplicaciones web. Sin embargo, el término de ingeniería de la web ha sido un término muy controversial especialmente para profesionales en disciplinas tales como la ingeniería del software ya que no la consideran como un campo dentro de la ingeniería. (Salguera, 2012)

Los principales aspectos de la ingeniería de la Web incluyen, entre otros, los siguientes temas:

- Diseño de procesos de negocio para aplicaciones web.
- Herramientas CASE para aplicaciones web.
- Desarrollo web colaborativo.
- Modelado conceptual de aplicaciones web.

- Diseño de Modelos de datos para sistemas de información web.
- Ingeniería web empírica.
- Entornos de desarrollo de aplicaciones web integrados.
- Herramientas de autor para contenido multimedia.
- Pruebas de rendimiento de aplicaciones basadas en web.
- Personalización y adaptación de aplicaciones web.
- Modelado de procesos para aplicaciones web.
- Herramientas y métodos de prototipado.
- Control de calidad y pruebas de sistemas.
- Ingeniería de requisitos para aplicaciones web.
- Aplicaciones para la Web Semántica.
- Factorías de software para la web.
- Métodos, herramientas y automatización de pruebas para aplicaciones web.
- Aplicaciones web móviles y ubicuas.
- Usabilidad de aplicaciones web.
- Accesibilidad para la web.
- Metodologías de diseño web.
- Formación en ingeniería de la web.
- Diseño de interfaces de usuario.
- Métricas para la web, estimación de costes y medición.

- Gestión de proyectos web y gestión de riesgos.
- Desarrollo y despliegue de servicios web

2.7 FASES O ETAPAS DE LA METODOLOGÍA UWE

La metodología UWE está basada en técnicas, notación y los mecanismos de extensión de UML como mencionan (Rotta et al., 2016)

UWE es un proceso del desarrollo para aplicaciones Web enfocado sobre el diseño sistemático, la personalización y la generación semiautomática de escenarios que guíen el proceso de desarrollo de una aplicación Web. UWE describe una metodología de diseño sistemática, basada en las técnicas de UML, la notación de UML y los mecanismos de extensión de UML. (Cabello, 2013b)

UWE es una metodología basada en el proceso unificado y UML para el desarrollo de aplicaciones web. UWE cubre todo el ciclo de vida de las aplicaciones web, su proceso de desarrollo se basa en fases principales.

A continuación, veremos las etapas de la metodología UWE.

2.7.1 *Captura, análisis y especificación de requisitos*

En este proceso, iterativo e incremental, incluye flujos de trabajo y puntos de control, y sus fases coinciden con las propuestas en el proceso unificado de modelado (Guerrero, 2014)

En esta etapa; se adquieren, reúnen y especifican las características que deben cumplir el sistema, a través de requerimientos funcionales, requerimientos no funcionales, identificación de actores.

- **Identificación de actores:** se utiliza tablas, que contendrá el actor y la descripción, para la identificación de actores que interactúen con el sistema. Un actor no necesariamente es una persona.

- **Requerimientos funcionales:** se utilizará para definir el funcionamiento del sistema “lo que debe hacer el sistema”. a través de tablas que contengan: rol “es el número del requerimiento funcional”, descripción “lo que el sistema debe hacer” y función “loque tiene que hacer el sistema al ejecutar ese requerimiento”.
- **Requerimientos no funcionales:** se utilizará para capturar las propiedades o características del sistema como ser: el rendimiento, disponibilidad, seguridad entre otros. A través de tabas que contengan: rol “los requerimientos no funcionales” y función “restricción del sistema”.

2.7.2 Diseño del sistema

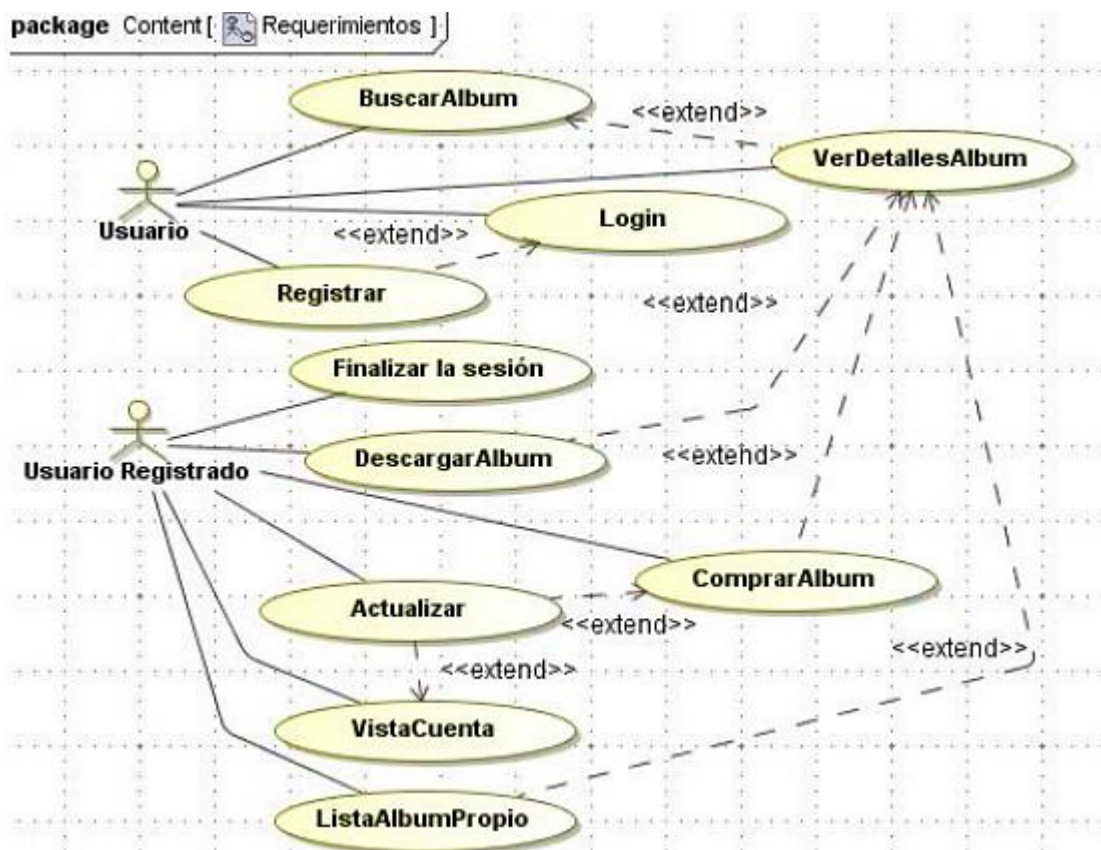
Se basa en la especificación de requisitos producido por el análisis de los requerimientos (fase de análisis), el diseño define cómo estos requisitos se cumplirán, la estructura que debe darse a la aplicación web. (Cabello, 2013a)

2.7.2.1 Diagrama de Casos de Uso

Un diagrama de casos de uso muestra la relación entre los actores y los casos de uso del sistema. Representa la funcionalidad que ofrece el sistema en lo que se refiere a su interacción externa.

Figura 3

Modelo de caso de uso de la aplicación portal musical



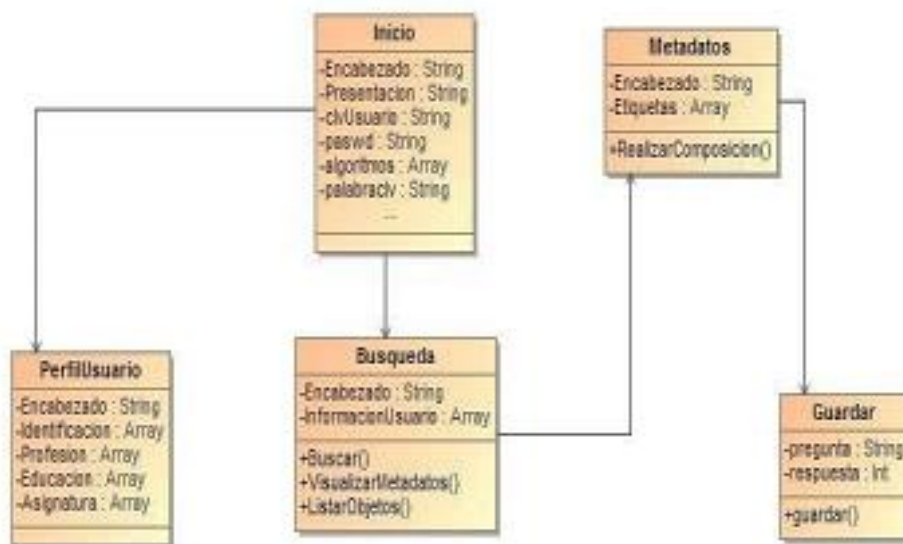
Nota. Por (Engineering, 2012)

En la figura podemos ver que las funciones de un usuario no registrado son limitadas, ya que solo puede buscar información sobre los discos, registrarse y loguearse, para llegar a ser usuario Registrado.

2.7.2.2 Modelo Conceptual

Este modelo especifica cómo se encuentran relacionados los contenidos del sistema, es decir, define la estructura de los datos que se encuentran alojados en el sitio web. (Aurazo, 2017)

Este modelo especifica cómo se encuentra relacionados los contenidos del sistema, es decir, define la estructura de los datos que se encuentran alojados en el sitio web.

Figura 4*Diagrama Conceptual*

Nota. Adaptado de (Proyecto fin de master (pdf), 2016)

2.7.2.3 Modelo de Navegación

El modelo de espacio de navegación es construido con las clases de navegación y las asociaciones de navegación y están representadas gráficamente de clase de UML. (Koch, 2000)

Este diseño especifica qué objetos pueden ser vistos a través de la aplicación web. En el proceso de construir el modelo de espacio de navegación el desarrollador toma decisiones cruciales de diseño, tales como qué vista del modelo conceptual es necesaria para la aplicación y cuáles serán los caminos de navegación requeridos para el aseguramiento de la funcionalidad.

2.7.2.4 Modelo de Presentación

En este modelo se representan las clases de navegación y de procesos que pertenecen a cada página web, estos son los elementos que introduce la metodología UWE en este modelo.

Tabla 2

Descripción de estereotipos.

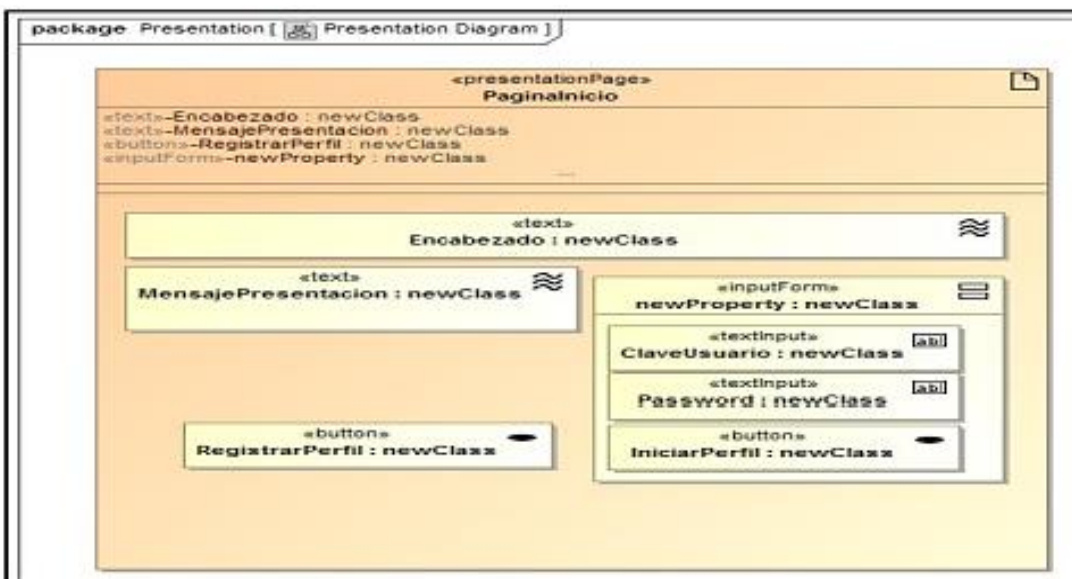
Icono	Descripción	Icono	Descripción
	Grupo de presentación		Página de presentación
	Texto		Entrada de texto
	Ancla		File Upload
	Botón		Imagen
	Formulario		Componente de cliente
	Alternativas de presentación		Selección

Nota. Elaboración propia basado en (KOCH, 2000)

El modelo de presentación ofrece una visión abstracta de la interfaz de usuario de aplicaciones web. Se basa en el modelo de navegación y en los aspectos concretos de la interfaz de usuario.

Figura 6

Diagrama de presentación



Nota. (Flores, 2009)

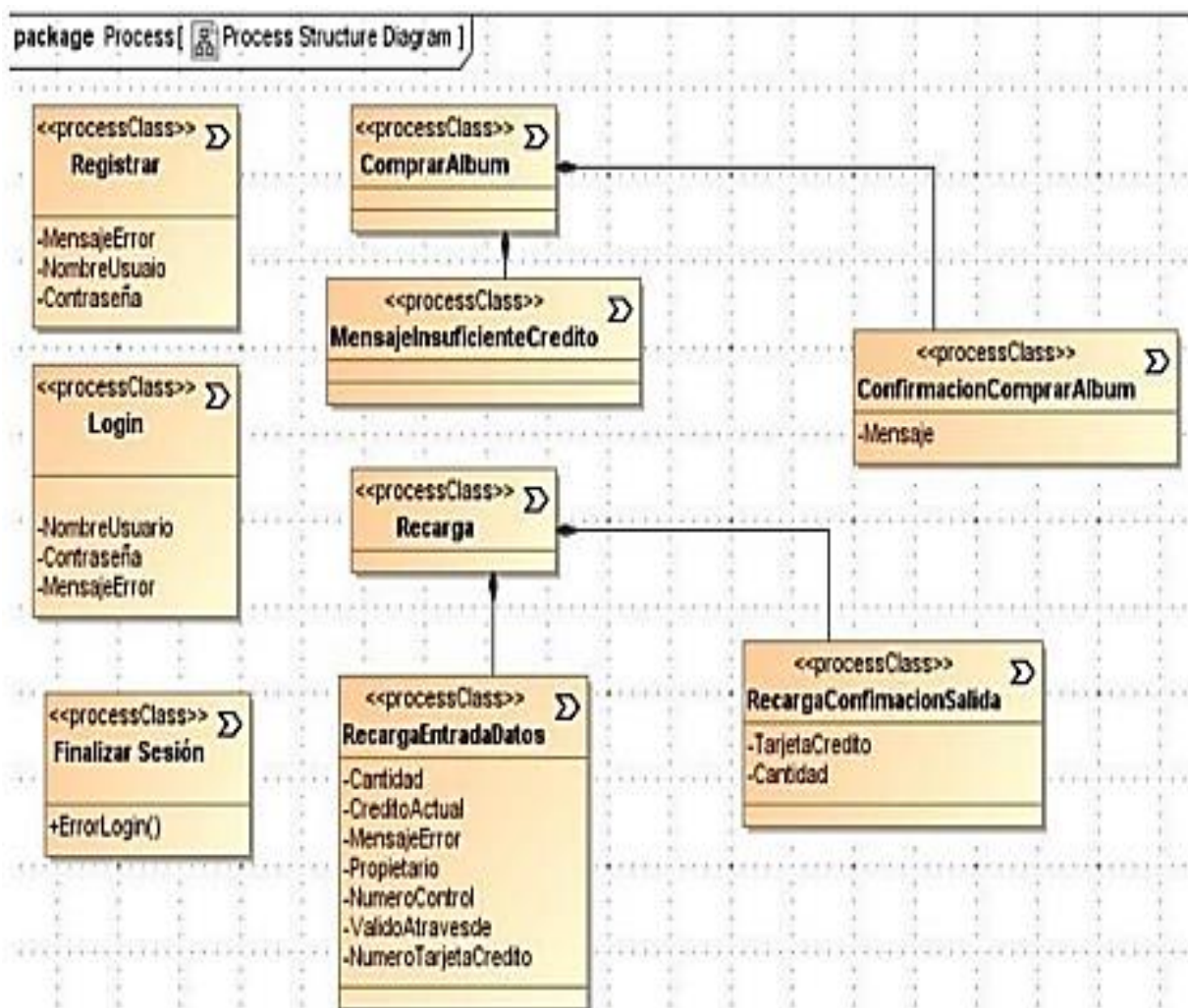
2.7.2.5 Modelo de Procesos

Representa la parte dinámica de la aplicación Web, especificando la funcionalidad de las transiciones y de los flujos de trabajo complejos de las actividades, contrario al modelo navegación, que representa la parte estática de la información [T-ESPE, (2013)].

El modelo de proceso del ejemplo se compone de las clases de procesos que se integran en el modelo de navegación, las clases de proceso adicionales para manejar la entrada del usuario y las actividades que definen el comportamiento de los procesos. El modelo de proceso muestra la relación de la clase de proceso principal y clases de procesos, que se han asociado al flujo del proceso.

Figura 7

Modelo de proceso de la aplicación portal musical.



Nota. Adaptado por (Engineering, 2012)

2.7.3 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, todo lo diseñado en la fase anterior. (Vallejos, n.d.)

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.

2.7.4 Pruebas

Las pruebas se utilizan para asegurar el correcto funcionamiento de secciones de código. (Vallejos, n.d., p.37)

2.7.5 La instalación o fase de implementación

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

2.7.6 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.

Mantenimiento y Seguimiento: Una vez puesta la Página Web a disposición de los usuarios hay que ir cambiando datos y mantener este sitio actualizado, ya que esta página no puede permanecer estática. Los problemas de uso no detectados durante el proceso de desarrollo pueden descubrirse a través de varios métodos, principalmente a través de los mensajes, opiniones de los usuarios, el comportamiento y uso del sitio. (Apaza Bilbao, 2021)

2.8 ARQUITECTURA CLIENTE-SERVIDOR

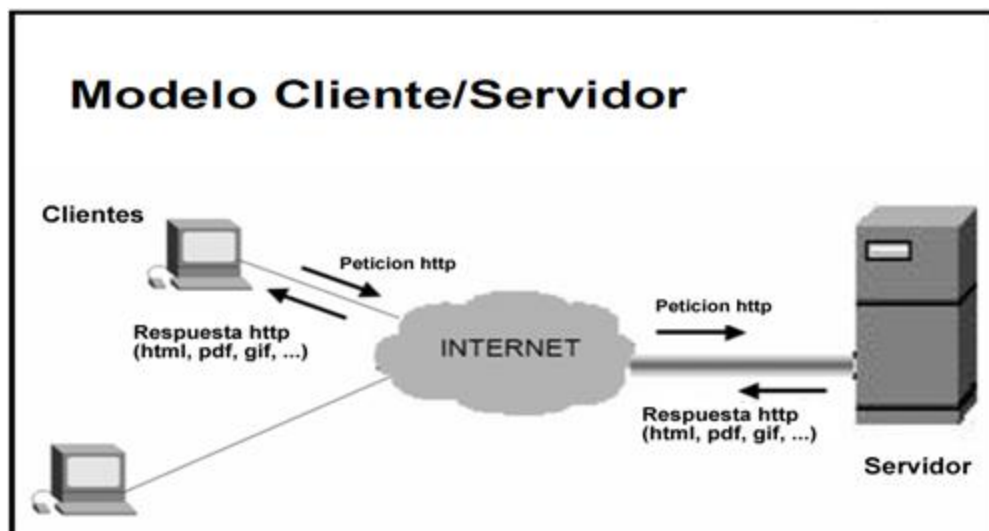
Según (TIC, 2016):

La estructura cliente – servidor es una arquitectura de computación en la que se consigue un procesamiento cooperativo de la información por medio de un conjunto de procesadores, de tal forma que uno o varios clientes, distribuidos geográficamente o no, solicitan servicios de computación a uno o más servidores.

Según (LÓPEZ PONCE, n.d.)

La arquitectura cliente-servidor es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Un cliente realiza peticiones a otro programa, el servidor, quien le da respuesta. Esta idea también se puede aplicar a programas que se ejecutan sobre una sola computadora, aunque es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras.

La red cliente-servidor es una red de comunicaciones en la cual los clientes están conectados a un servidor, en el que se centralizan los diversos recursos y aplicaciones con que se cuenta; y que los pone a disposición de los clientes cada vez que estos son solicitados. Esto significa que todas las gestiones que se realizan se concentran en el servidor, de manera que en él se disponen los requerimientos provenientes de los clientes que tienen prioridad, los archivos que son de uso público y los que son de uso restringido, los archivos que son de sólo lectura y los que, por el contrario, pueden ser modificados, etc. Este tipo de red puede utilizarse conjuntamente en caso de que se esté utilizando en una red mixta.

Figura 8*Modelo Cliente-Servidor*

Nota. Adaptado de modelo cliente servidor, por Flores Gines

2.8.1 Características

En la arquitectura C/S el remitente de una solicitud es conocido como cliente. Sus características son:

- Es quien inicia solicitudes o peticiones, tienen por tanto un papel activo en la comunicación (dispositivo maestro o amo).
- Espera y recibe las respuestas del servidor.
- Por lo general, puede conectarse a varios servidores a la vez.
- Normalmente interactúa directamente con los usuarios finales mediante una interfaz gráfica de usuario.

En la arquitectura C/S sus características generales son:

- El Cliente y el Servidor pueden actuar como una sola entidad y también pueden actuar como entidades separadas, realizando actividades o tareas independientes.
- Las funciones de Cliente y Servidor pueden estar en plataformas separadas, o en la misma plataforma.

- Cada plataforma puede ser escalable independientemente. Los cambios realizados en las plataformas de los Clientes o de los Servidores, ya sean por actualización o por reemplazo tecnológico, se realizan de una manera transparente para el usuario final.
- La interrelación entre el hardware y el software están basados en una infraestructura poderosa, de tal forma que el acceso a los recursos de la red no muestra la complejidad de los diferentes tipos de formatos de datos y de los protocolos.
- Su representación típica es un centro de trabajo (PC), en donde el usuario dispone de sus propias aplicaciones de oficina y sus propias bases de datos, sin dependencia directa del sistema central de información de la organización.

2.8.2 Ventajas

- **Centralización del control:** los accesos, recursos y la integridad de los datos son controlados por el servidor de forma que un programa cliente defectuoso o no autorizado no pueda dañar el sistema. Esta centralización también facilita la tarea de poner al día datos u otros recursos (mejor que en las redes P2P).
- **Escalabilidad:** se puede aumentar la capacidad de clientes y servidores por separado. Cualquier elemento puede ser aumentado (o mejorado) en cualquier momento, o se pueden añadir nuevos nodos a la red (clientes y/o servidores).
- **Fácil mantenimiento:** al estar distribuidas las funciones y responsabilidades entre varios ordenadores independientes, es posible reemplazar, reparar, actualizar, o incluso trasladar un servidor, mientras que sus clientes no se verán afectados por ese cambio (o se afectarán mínimamente). Esta independencia de los cambios también se conoce como encapsulación.

- Existen tecnologías, suficientemente desarrolladas, diseñadas para el paradigma de C/S que aseguran la seguridad en las transacciones, la amigabilidad de la interfaz, y la facilidad de empleo.

2.8.3 Desventajas

- La congestión del tráfico ha sido siempre un problema en el paradigma de C/S. Cuando una gran cantidad de clientes envían peticiones simultáneas al mismo servidor, puede ser que cause muchos problemas para éste (a mayor número de clientes, más problemas para el servidor). Al contrario, en las redes P2P como cada nodo en la red hace también de servidor, cuantos más nodos hay, mejor es el ancho de banda que se tiene.
- El paradigma de C/S clásico no tiene la robustez de una red P2P. Cuando un servidor está caído, las peticiones de los clientes no pueden ser satisfechas. En la mayor parte de redes P2P, los recursos están generalmente distribuidos en varios nodos de la red. Aunque algunos salgan o abandonen la descarga; otros pueden todavía acabar de descargar consiguiendo datos del resto de los nodos en la red.
- El software y el hardware de un servidor son generalmente muy determinantes. Un hardware regular de un ordenador personal puede no poder servir a cierta cantidad de clientes. Normalmente se necesita software y hardware específico, sobre todo en el lado del servidor, para satisfacer el trabajo. Por supuesto, esto aumentará el coste.
- El cliente no dispone de los recursos que puedan existir en el servidor. Por ejemplo, si la aplicación es una Web, no podemos escribir en el disco duro del cliente o imprimir directamente sobre las impresoras sin sacar antes la ventana previa de impresión de los navegadores.

2.8.4 Seguridad de modelo cliente - servidor

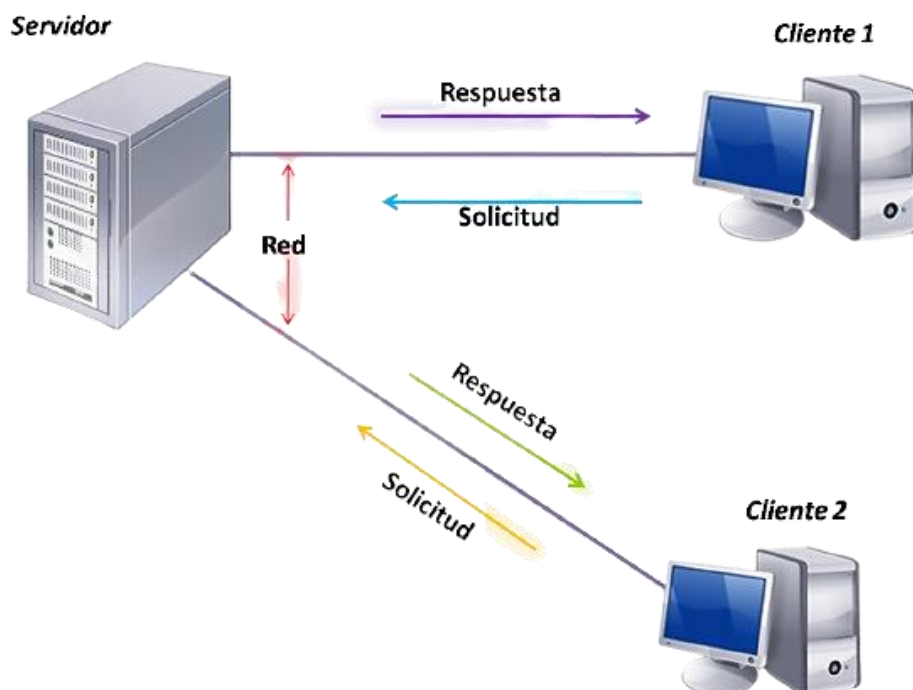
Es de gran importancia por el valor intrínseco para los usuarios. Tiene 3 componentes: Confidencialidad, Integridad y disponibilidad.

- Confidencialidad: Protección contra individuos no autorizados.
- Integridad: Protección contra la alteración o corrupción.
- Disponibilidad: Protección contra interferencia con los procedimientos de accesos a los recursos.

2.8.5 Arquitectura de dos y tres niveles cliente-servidor

2.8.5.1 Arquitectura de dos niveles

La arquitectura en 2 niveles se utiliza para describir los sistemas cliente/servidor en donde el cliente solicita recursos y el servidor responde directamente a la solicitud, con sus propios recursos. Esto significa que el servidor no requiere otra aplicación para proporcionar parte del servicio.

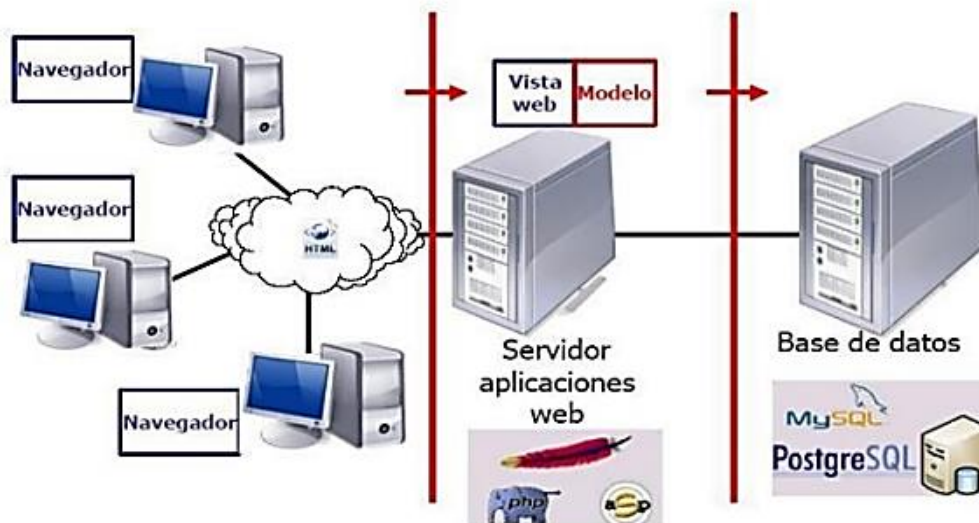
Figura 9*Arquitectura en dos niveles*

Nota.: (Leal Castellano, Leal Molina, & Medina Castiblanco, 2011).

2.8.5.2 Arquitectura tres niveles

En la arquitectura en tres niveles, existe un nivel intermedio. Esto significa que la arquitectura generalmente está compartida por:

- Un cliente, es decir el equipo que solicita los recursos, equipado con una interfaz de usuario (generalmente un navegador web) para la presentación.
- El servidor de aplicaciones (también denominado software intermedio), cuya tarea es proporcionar los recursos solicitados, pero que requiere de otro servidor para hacerlo.
- El servidor de datos, que proporciona al servidor de aplicaciones los datos que requiere.

Figura 10*Arquitectura de tres niveles*

Nota. (Leal Castellano, Leal Molina, & Medina Castiblanco, 2011)

2.8.5.3 Comparación entre ambos tipos de arquitecturas

La arquitectura en dos niveles es, por lo tanto, una arquitectura cliente/servidor en la que el servidor es polivalente, es decir, puede responder directamente a todas las solicitudes de recursos del cliente.

Sin embargo, en la arquitectura en 3 niveles, las aplicaciones al nivel del servidor son descentralizadas de uno a otro, es decir, cada servidor se especializa en una determinada tarea, (por ejemplo: servidor web/servidor de bases de datos). La arquitectura en 3 niveles permite:

- Un mayor grado de flexibilidad.
- Mayor seguridad, ya que la seguridad se puede definir independientemente para cada servicio y en cada nivel.
- Mejor rendimiento, ya que las tareas se comparten entre servidores.

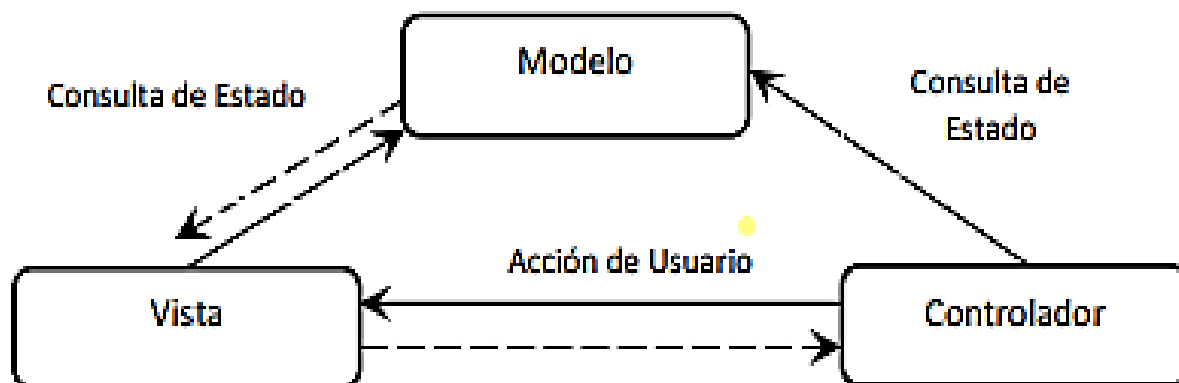
Tabla 3*Comparación de arquitecturas.*

Dos niveles	Tres niveles
✓ Consiste en una capa de presentación y lógica de la aplicación, y la otra a la base de datos. Cuando se requiera poco procesamiento de datos en la organización.	✓ Consiste en una capa de la presentación, otra capa de la lógica de la aplicación y la capa de la base de datos.
✓ Cuando se tiene una base de datos centralizada en un solo servidor.	✓ Cuando se requiera mucho procesamiento de datos en la aplicación
✓ Cuando la base de datos es relativamente estática.	✓ En aplicaciones donde la funcionalidad esté en constante cambio.
✓ Cuando se requiere un mantenimiento mínimo.	✓ Cuando los procesos no están relativamente muy relacionados con los datos

Nota. (Leal Castellano, Leal Molina, & Medina Castiblanco, 2011)

2.9 MODELO VISTA CONTROLADOR (MVC)

Según (Estava Muñoz, 2013) “es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones”

Figura 11*Modelo Vista Controlador***Nota.** Tomado de Flores Fernández (2012)

Para (Pantoja Bascón, 2004) "MVC es bastante utilizado en la actualidad en marcos de aplicación orientados a objeto desarrollados para construir aplicaciones de gran tamaño; Java Swing, Apache Struts, Microsoft ASP.NET, las transformaciones XSL o incluso los documentos LATEX siguen este patrón de diseño".

2.9.1 Modelo

El Modelo representa las estructuras de datos. Típicamente el modelo de clases contendrá funciones para consultar, insertar y actualizar información de la base de datos.

2.9.2 Vista

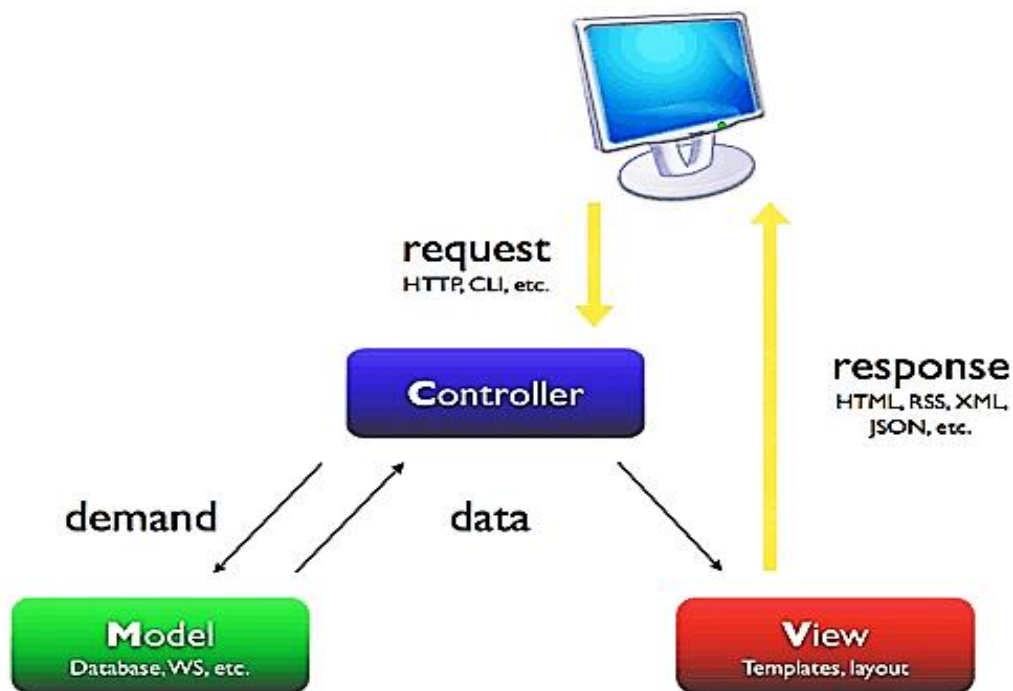
La Vista es la información presentada al usuario. Una vista puede ser una página web o una parte de una página.

2.9.3 Controlador

Este es el puente entre la vista y el modelo ya que el usuario solicitará información mediante la vista y este pasará por el controlador para posteriormente realizar la petición al modelo, habitualmente es llamado la capa de lógica del negocio.

Figura 12

Arquitectura MVC



Nota.: Aguilar, (2008)

2.9.4 Característica del modelo, vista, controlador

Aunque se pueden encontrar diferentes implementaciones de MVC, la característica del flujo que sigue el control generalmente es el siguiente:

- El usuario interactúa con la interfaz de usuario de alguna forma (por ejemplo, el usuario pulsa un botón, enlace, etc.).
- El controlador recibe (por parte de los objetos de la interfaz-vista) la notificación de la acción solicitada por el usuario. El controlador gestiona el evento que llega, frecuentemente a través de un gestor de eventos (handler) o callback.
- El controlador accede al modelo, actualizándose, posiblemente modificándolo de forma adecuada a la acción solicitada por el usuario. Los controladores complejos están a menudo estructurados usando un patrón de comando que encapsula las acciones y simplifica su extensión.

- El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se reflejan los cambios en el modelo. El modelo no debe tener conocimiento directo sobre la vista. Sin embargo, el patrón de observador puede ser utilizado para proveer cierta indirección entre el modelo y la vista, permitiendo al modelo notificar a los interesados de cualquier cambio.
- La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

2.10 MÉTRICAS DE CALIDAD

Pressman clasifica el campo de las métricas en 6 categorías o grupos de métricas distintos, dentro de ellas se encuentran las: Métricas de calidad. Éstas proporcionan una indicación de cómo se ajusta el software a los requisitos implícitos y explícitos del cliente. Es decir cómo se va a medir para que el sistema se adapte a los requisitos que pide el cliente. (PRESSMAN, 2009)

El proceso para intentar mejorarlo, el producto se mide intentar aumentar su facilidad. para la necesidad de la medición es algo evidente después que todo es lo que nos permite cuantificar y por consiguiente gestionar de forma más efectiva.

2.10.1 Norma ISO/IEC 25000

La familia ISO/IEC 25000 es el resultado de la evolución de otras normas anteriores, especialmente de las normas ISO/IEC 9126, que describe las particularidades de un modelo de calidad del producto software, e ISO/IEC 14598, que abordaba el proceso de evaluación de productos software. Esta familia de normas ISO/IEC 25000 se encuentra compuesta por cinco divisiones.

Según (Roa, Morales, & Gutiérrez, Norma ISO/IEC 25000 ISO/IEC 25000 Standard, 2015) Las normas ISO/IEC 25000 también llamadas SQuare (Requisitos y Evaluación de Calidad de Productos de Software) están conformadas por las normas ISO/IEC 9126 e ISO/IEC 14598, surgen para crear modelos, métricas, procesos y herramientas de evaluación de calidad del software como producto, por medio de la especificación de los requisitos. Es una familia de normas que tiene por objetivo la creación de un marco de trabajo común para evaluar la calidad del producto software

Estos estándares establecen criterios para la especificación de requisitos de calidad de productos software, sus métricas y su evaluación, e incluyen un modelo de calidad para unificar las definiciones de calidad de los clientes con los atributos en el proceso de desarrollo.

Figura 13

Modelo ISO/ICE 25000



Nota: Luigi Mendoza Jiibaja

2.10.2 Funcionabilidad

La norma ISO/IEC 25000, también conocida como SQuaRE (System and Software Quality Requirements and Evaluation), establece criterios para evaluar la calidad del software. Su funcionalidad es la de:

- Proporcionar una guía para el uso de las nuevas series de estándares
- Establecer criterios para la especificación de requisitos de calidad de productos software
- Unificar las definiciones de calidad de los clientes con los atributos en el proceso de desarrollo
- Recoger los requisitos, recomendaciones y guías para el proceso de evaluación de la calidad de un software

2.10.3 Fiabilidad

Capacidad de un sistema o componente para desempeñar las funciones especificadas, cuando se usa bajo unas condiciones y periodo de tiempo determinados sin interrupciones o fallos. Esta característica se subdivide a su vez en las siguientes sub características:

- **Ausencia de fallos.** Capacidad del sistema de llevar a cabo sus funciones sin fallos bajo condiciones normales de operación.
- **Disponibilidad.** Capacidad del sistema o componente de estar operativo y accesible para su uso cuando se requiere.
- **Tolerancia a fallos.** Capacidad del sistema o componente para operar según lo previsto en presencia de fallos hardware o software.

- **Capacidad de recuperación.** Capacidad del producto software para recuperar los datos directamente afectados y reestablecer el estado deseado del sistema en caso de interrupción o fallo

2.10.4 Usabilidad

Algunas de las características que la conforman son: comprensibilidad, aprendibilidad, operabilidad, atractivo, cumplimiento de usabilidad, capacidad para reconocer su adecuación, capacidad de aprendizaje, capacidad de ser usado, protección contra errores de usuario, estética de la interfaz de usuario, accesibilidad. También fácil de usar, fácil de aprender, atractivo para el usuario, conforme a normas, uso intuitivo. (Roa, Morales, & Gutiérrez, Norma ISO/IEC 25000, 2015)

2.10.5 Portabilidad

Hace referencia a la capacidad del software para ser transferido o adaptado fácilmente a diferentes entornos, sistemas operativos, plataformas y configuraciones sin necesidad de realizar modificaciones significativas. Es fácil de instalar y desinstalar, además permite ser adaptado de forma efectiva a diferentes entornos de hardware o software

2.10.6 Mantenibilidad

Es la medida del esfuerzo requerido para realizar cambios en los componentes de un sistema de manera efectiva y eficiente. Algunas de sus características son: analizabilidad, modificabilidad, estabilidad, testabilidad, cumplimiento de mantenibilidad, modularidad. Puede ser modificado, actualizado, corregido y mejorado de manera efectiva y eficiente, sin afectar su funcionalidad, rendimiento o fiabilidad.

2.10.7 Eficiencia

Es el comportamiento del sistema: funcionalidad, capacidad, utilización de recursos y respuesta temporal. Dentro de sus características se encuentra que el sistema requiere la

utilización de un mínimo de recursos (por ejemplo tiempo de CPU) para ejecutar una tarea determinada.

2.10.8 Métricas basadas en la función

La métrica punto función (PF) se usa de manera efectiva como medio para medir la funcionalidad que entrega un sistema. PF se deriva empleando una relación empírica basada en medidas contables del dominio de la información del software y las evaluaciones de la complejidad de este. Sea los valores de dominio de la información según la tabla.

Tabla 4

Tabla factor de ponderación

	PARÁMETROS DE MEDICIÓN	CUENTA	FACTORES DE PONDERACIÓN			TOTAL
			Simple	Medio	complejo	
1	Nro. de entradas	X	3	4	5	=
2	Nro. de salidas de usuario	X	4	5	7	=
3	Nro. de peticiones de usuario	X	3	4	6	=
4	Nro. de archivos	X	7	10	15	=
5	Nro. de interfaces externas	X	5	7	10	=
Cuenta total						

Nota. (Pressman, 2002).

- Número de entradas de usuario: Se cuenta cada entrada de usuario que proporcione al software diferentes datos aplicados a la aplicación, las entradas deben ser distinguidas de las peticiones que se contabilizan por separado.
- Número de salidas del usuario: Se encuentra cada salida que proporciona al usuario información orientada a la aplicación. En este contexto se refieren a informes, pantallas, mensajes de error.

- Número de peticiones al usuario: Una petición está definida como una entrada interactiva que resulta de la generación de algún tipo de respuesta en forma de salida interactiva se cuenta cada petición por separado.
- Número de archivos: Se cuenta cada archivo maestro lógico es decir una agrupación lógica de datos que puede ser una parte en una gran base de datos o un archivo independiente.
- Número de interfaces externas: Se cuentan todas las interfaces legibles por la máquina, por ejemplo: archivo de datos, en cintas o discos que son utilizados para transmitir información a otro sistema.

A cada conteo se le asocia un valor de complejidad, no obstante, la determinación de la complejidad es un poco subjetiva. Para calcular los puntos de función se usa la siguiente ecuación.

$$PF = \text{Cuenta Total} * (0.65 + 0.01 * \sum Fi)$$

Cuenta total es la suma de todas las entradas de PF obtenidos de la tabla anterior. Fi donde i puede ser de uno hasta 14 valores de ajuste de complejidad basados en las respuestas a las cuestiones señaladas de la tabla.

Tabla 5

Factor de Escala

	Parámetros de medición	Cuenta	Factores de ponderación			total
			Simple	Medio	Complejo	
1	Nro. de entradas de usuario	X	3	4	6	

Nota. Pressman, (2002)

Tabla 6*Valoración métrica punto función*

Nro.	Preguntas
1	¿Requiere el sistema copia de seguridad y recuperación?
2	¿Requiere comunicación de datos?
3	¿Existen funciones de procesos distribuidos?
4	¿El rendimiento es crítico?
5	¿Sera ejecutado el sistema en entorno existente y fuertemente utilizado?
6	¿entrada de datos en línea?
7	¿Requiere la entrada de datos interactiva que las transiciones de entrada?
8	¿se llevan a cabo sobre múltiples pantallas o varias opciones?
9	¿se actualizan los archivos maestros de forma interactiva?
10	¿Son complejos de las entradas de salidas de archivos?
11	¿lógica del proceso interno compleja?
12	¿se diseña el código para ser reutilizable?
13	¿están incluidas en el diseño conversiones de instalación?
14	¿instalaciones múltiples?
15	¿facilidad de cambios?

Nota. Pressman, (2002)

Los valores constantes de la ecuación anterior y los factores de peso aplicados en las encuestas de los ámbitos de información han sido determinados empíricamente.

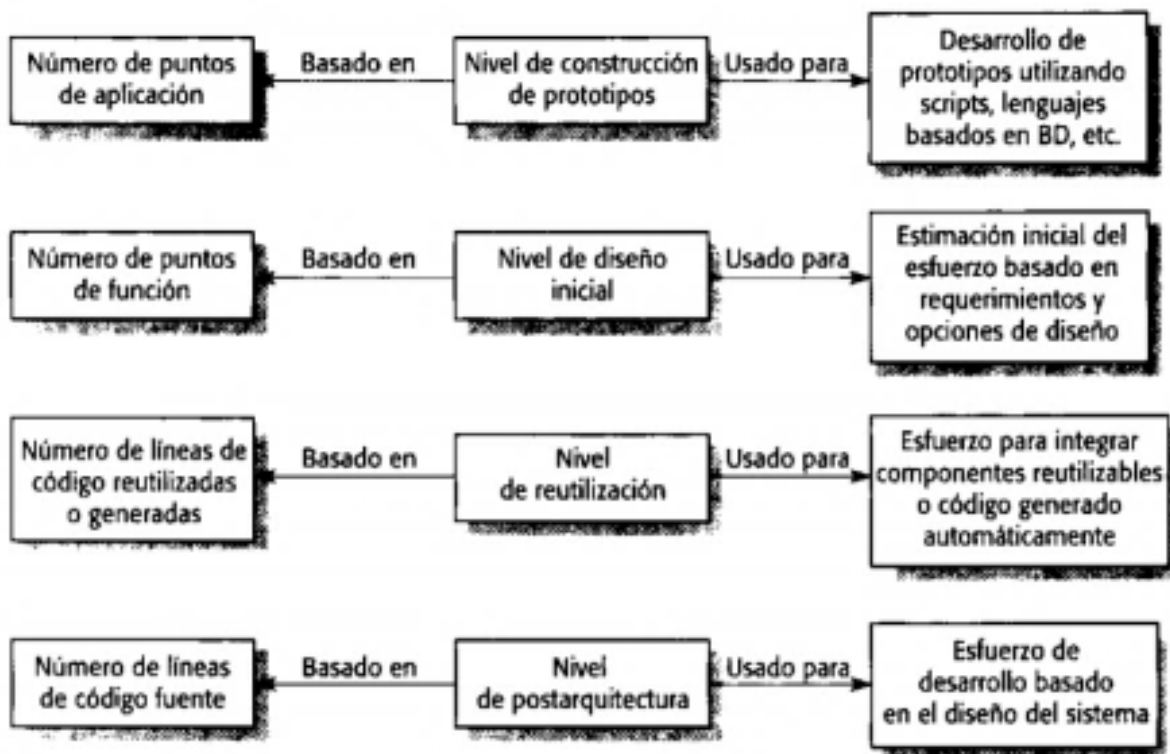
2.11 MÉTODO DE ESTIMACIÓN DEL SOFTWARE

2.11.1 *Cocomo II*

COCOMO es un modelo de formulación matemática con un fuerte componente de base empírica, principalmente utilizado para estimación de costos en los proyectos de software. (Garita, 2014)

Los modelos algorítmicos de estimación de costo en proyectos de software tienen alcances específicos a contextos limitados. No hay uniformidad en la aplicación efectiva de un determinado modelo sobre otro. La comparación de los resultados obtenidos en varios modelos parece ser la forma adecuada de obtener estimaciones realistas. (Garita-González & Lizano-Madriz, 2018)

COCOMO II es un modelo que permite estimar el coste, esfuerzo y tiempo cuando se planifica una nueva actividad de desarrollo software. Está asociado a los ciclos de vida modernos. COCOMO II apunta hacia los proyectos software de los 90 y de la primera década del 2000, y continuará evolucionando durante los próximos años. (Somerville, 2005)

Figura 14*Niveles de Cocomo***Nota.:** Somerville, 2005, pág. 573

2.12 SEGURIDAD DE APLICACIONES WEB

Según (Pressman, 2005):

Las pruebas de seguridad están diseñadas para probar la vulnerabilidad en el ambiente del lado del cliente, las comunicaciones de red que ocurren mientras los datos pasan del cliente al servidor y de vuelta y el ambiente del lado del servidor. Cada uno de estos dominios puede recibir ataques y es labor de quien prueba la seguridad descubrir las debilidades que pueden explotar quienes tengan la intención de hacerlo. A continuación, se muestra las vulnerabilidades al lado del cliente

- Vulnerabilidad en el navegador.
- Vulnerabilidad en los correos electrónicos.

- Acceso no autorizado de cookies.
- Simulación en la comunicación con el servidor.

Por lo tanto, se puede aplicar las siguientes medidas

Cortafuegos: Mecanismo de filtrado combinación de hardware y software para examinar cada paquete de información entrante para garantizar que no sea un dato sospechoso.

Uso de autenticación: Una contraseña o clave es una forma de autenticación que utiliza información secreta para controlar el acceso hacia algún recurso. A aquellos que desean acceder a la información se les solicita una clave, si conocen o no conocen la contraseña, se concede o se niega el acceso a la información según sea el caso.

Modelo de Criptografía: La encriptación está asociada con la transformación de un mensaje inteligente a una forma no inteligente con la ayuda de una clave secreta antes de que sea colocada en un medio inseguro.

Algoritmo básico del hash: Una función criptográfica hash- usualmente conocida como "hash"- es un algoritmo matemático que transforma cualquier bloque arbitrario de datos en una nueva serie de caracteres con una longitud fija. Independientemente de la longitud de los datos de entrada, el valor hash de salida tendrá siempre la misma longitud.

2.12.1 Seguridad del Software ISO-27001

Es un sistema de gestión para la seguridad de información que sirve para brindar soporte de datos que se encuentran registrados en la organización. Para ello se implementa un SGSI con estándar 27001 para el cuidado de información brindando confidencialidad, disponibilidad e integridad de los datos para el buen uso de la información y no divulgación del mismo en organizaciones ya sean grandes o pequeñas.

Es una norma desarrollada como modelo para el establecimiento, implementación, operación, monitorización, revisión, mantenimiento y mejora de un SGSI para cualquier tipo de organización. Dado que la norma ISO 27001:2013 trabaja sobre un enfoque a procesos, una aplicación detallada y puntual de la misma, requiere de un manual de procesos, completa y debidamente establecidos, definidos, documentados y validados. El correcto diseño, establecimiento y operación de un SGSI permite a la organización realizar más eficientemente sus actividades, sin embargo, si no se ha aplicado una reingeniería de procesos a la organización previa al establecimiento del SGSI, esta ralentiza sus operaciones. Esta norma tiene un total de 14 Dominios, 35 Objetivos de Control y 114 Controles. (MANTILLA, 2017)

La seguridad de la información según la ISO/IEC 27001 se basa en tres pilares fundamentales la confidencialidad, la disponibilidad y la integridad, estos factores se deben garantizar en una adecuada gestión de seguridad de la información. (López, 2017)

- **Confidencialidad:** La confidencialidad es uno de los factores de mayor riesgo y más complejos de asegurar ya que la información es manejada por el eslabón más débil de la cadena, el usuario, quien por error, omisión o con conocimiento de causa puede dejar expuesta la información vulnerando la organización.
- **Disponibilidad:** hace referencia al almacenamiento de la información y su accesibilidad al usuario, es la posibilidad de tener la información en el tiempo y espacio requerido. Brindar la información, mantener actualizado el sistema y facilitar su acceso asegura enormes beneficios para la empresa. Sin embargo, debe optimizarse el control de riesgos y disminuir la vulnerabilidad de este flujo. Es en este punto, que la seguridad a través de contraseñas y permisos toma relevancia.
- **Integridad:** es el factor más importante de la seguridad de la información ya que de nada sirve una información disponible y confidencial si su integridad ha sido vulnerada es decir no es exacta ni válida.

2.13 PRUEBAS AL SOFTWARE

Como bien se sabe, el desarrollo de software es una actividad que está sujeta a los errores humanos, por tanto es probable encontrarse en cualquier desarrollo con defectos, errores y fallas. Para tratar de evitar esas situaciones al máximo existen ya definidas metodologías de desarrollo que si bien no garantizan la eliminación total de errores si disminuyen altamente la probabilidad de falla. (Camilo & Ochoa, 2010, pág. 5)

2.13.1 *Prueba de caja negra*

Consiste en estudiar la especificación de las funciones, la entrada y la salida para derivar los casos. La prueba ideal del software consiste en probar todas las posibles entradas y salidas del programa. Consideran la función específica para la cual fue creado el producto (qué es lo que hace).

Las pruebas se llevan a cabo sobre la interfaz del sistema.

No es una alternativa a la prueba de caja blanca.

Complementan a las pruebas de caja blanca

La prueba de caja negra, también denominados pruebas de comportamiento encuentra errores de:

- Funciones incorrectas o ausentes
- Errores de interfaz
- Errores en estructuras de datos o en accesos a bases de datos externas
- Errores de rendimiento
- Errores de inicialización y de terminación

Figura 15

Estructura caja negra



Nota. Ingeniería de Software, (2011).

Técnicas de prueba de Caja negra

- Partición equivalente.
- Análisis de valores límite.

Prueba de partición equivalente

Se basa en la división de los parámetros de entrada en un conjunto de clases de datos denominadas clases de equivalencia.

- Se parte de la premisa de que cualquier elemento de una clase de equivalencia es representativo del resto del conjunto.
- A partir de las clases de equivalencia se obtienen las clases de equivalencias válidas e inválidas.

Prueba de Valores Límite (VL)

- Se basa en la evidencia experimental de que los errores suelen aparecer con mayor probabilidad en los extremos de los campos de entrada.
- Un análisis de las condiciones límite de las clases de equivalencia aumenta la eficiencia de las pruebas.

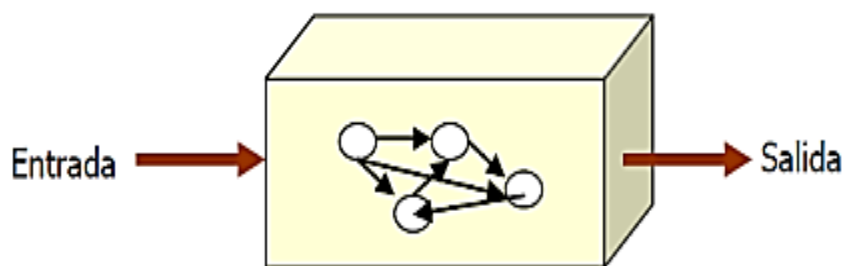
- Condiciones límite: valores justo por encima y por debajo de los márgenes de la clase de equivalencia. (A. Goñi, 2018)

2.13.2 Pruebas de caja blanca

Esta prueba se centra en la estructura interna del programa, en este caso la prueba consiste en probar todos los posibles caminos de ejecución a través de las instrucciones del código, que puedan trazarse.

Figura 16

Estructura caja blanca



Nota. Ingeniería de Software, (2011)

Mediante esta prueba, el ingeniero desarrollador del software puede:

- Garantizar que se recorre por lo menos una vez todos los caminos independientes de cada módulo.
- Recorrer todas las decisiones lógicas en sus condiciones verdadera y falsa.
- Recorrer todos los bucles en sus límites y con sus límites operacionales.
- Recorrer las estructuras internas de datos para asegurar su validez

2.14 HERRAMIENTAS

2.14.1 Xampp

Es XAMPP, es un servidor de plataforma libre, es un software que integra en una sola aplicación, un servidor web Apache, intérpretes de lenguaje de scripts PHP, un servidor de base de datos MySQL, un servidor de FTP FileZilla, el popular administrador de base de datos escrito en PHP, MySQL, entre otros módulos.

Te permite instalar de forma sencilla Apache en tu propio ordenador, sin importar tu sistema operativo (Linux, Windows, MAC o Solaris). Y lo mejor de todo es que su uso es gratuito. (Idalia, n.d.)

2.14.1.1 Ventajas

Xampp es una herramienta que permite instalar el entorno de Postgres, Apache y PHP, para empezar proyectos web o revisar proyectos localmente

2.14.1.2 Desventajas

- No soporta MySQL desde la consola. 64 Xampp trae PhpMyAdmin para administrar las bases de datos de MySQL, sin embargo, para tareas más específicas es mejor utilizar la consola (línea de comandos) y Xampp no la soporta.
- No se pueden actualizar individualmente las versiones de los programas que instala. Xampp trae las últimas versiones de las aplicaciones que instala, sin embargo, cuando pasa el tiempo y salen nuevas versiones de las mismas, no queda otra salida que reinstalar todo Xampp.
- Dificultad para configurar aplicaciones de terceros.
- Una desventaja es la garantía. Este programa se distribuye con la esperanza de que sea útil, más sin ninguna garantía, sin siquiera la garantía implícita de su mercadabilidad y aptitud para un propósito determinado.

2.14.2 MySQL

MySQL es más que un simple software de gestión de bases de datos; es una solución integral que ha sido adoptada en todo el mundo, desde aplicaciones web pequeñas hasta grandes proyectos corporativos. (MySQL: Características, Ventajas y Desventajas, 2024)

MySQL es un gestor de bases de datos, y en la actualidad uno de los más utilizados y conocidos del mercado. Principalmente en el ámbito de creación de sitios web, está catalogada como la base de datos open source más popular del mundo. (SQL, 2024)

2.14.2.1 Ventajas

- MySQL es de distribución libre y gratuita, podemos descargarlo del sitio web oficial de MySQL sin ningún costo.
- MySQL es Open Source, es decir que su código fuente puede ser obtenido y modificado por cualquier desarrollador de software.
- Es veloz al realizar operaciones, y garantiza un buen rendimiento de las aplicaciones.
- Es multiplataforma, es decir que soporta sistemas operáticos como Linux, Solaris, Windows, etc.
- Es fácil de instalar y configurar.
- Es seguro y confiable, las contraseñas están encriptadas en MySQL.
- Es rápido: se considera uno de los gestores de base de datos más rápidos del mercado.
- Puede ser ejecutado en computadoras con pocos recursos, ya que no exige mayores requerimientos.

2.14.2.2 Desventajas

- Limitaciones de seguridad: MySQL puede tener algunas limitaciones de seguridad, especialmente si no se configura adecuadamente. Deberás asegurarte de que la base de datos esté bien protegida para evitar vulnerabilidades.
- Varias de las utilidades de MySQL no están documentadas.
- No es del todo intuitivo, en comparación con otros programas.
- No maneja de manera tan eficiente una base de datos con un tamaño muy grande.

2.14.3 PHP

Una de las principales razones de la popularidad de PHP como lenguaje de creación de scripts para Web es su amplio soporte a diferentes bases de datos, facilitando que los desarrolladores creen sitios sustentados en bases de datos, y que se hagan nuevos prototipos de aplicaciones Web de manera rápida y eficiente sin demasiada complejidad. (Eslava, 2013)

Se caracteriza por su potencia, versatilidad, robustez y modularidad. Al igual que ocurre con tecnologías similares, los programas son integrados directamente dentro del código HTML. El lenguaje PHP es uno de los lenguajes más utilizados actualmente en el desarrollo de aplicaciones web y viene experimentando un constante crecimiento en su nivel de utilización en Interno. (Zambrano & Almeida Zambrano, 2022)

2.14.3.1 Ventajas

- Lenguaje totalmente libre y abierto.
- Posee una curva de aprendizaje muy baja.
- Los entornos de desarrollo son de rápida y fácil configuración.
- Fácil de instalar: existen paquetes autoinstalables que integran PHP rápidamente.
- Fácil acceso e integración con las bases de datos. 67
- Posee una comunidad muy grande.

- Es el lenguaje con mayor usabilidad en el mundo.
- Es un lenguaje multiplataforma.
- Completamente orientado al desarrollo de aplicaciones web dinámicas y/o páginas web con acceso a una Base de Datos.
- El código escrito en PHP es invisible al navegador ya que se ejecuta al lado del servidor y los resultados en el navegador es HTML.
- Posee una versatilidad para la conexión con la mayoría de base de datos que existen en la actualidad.

2.14.3.2 Desventajas

- El inconveniente es que el código fuente no pueda ser ocultado de una manera eficiente.
- La ofuscación es una técnica que puede dificultar la lectura del código y, en ciertos aspectos representa tiempos de ejecución.
- Nuestro código estará seguro para ejecutar si es nuestro propio servidor. Por lo tanto, si un cliente requiere su código en su pc, tendríamos que dejar el código fuente, sin manera de ocultarlo, aunque hay muchas aplicaciones para PHP que nos ayuda a encriptar el código fuente.
- Si no lo configuras y/o proteges correctamente dejas abiertas muchas brechas de seguridad que a la larga tendremos problemas.
- Solo se ejecuta en un servidor y se necesita un servidor web para que funcione.

2.14.4 JavaScript

El uso del lenguaje de programación JavaScript es estratégico y fundamental para manejar datos en tu página web y de realizar conexiones entre el sitio y las API que también

desarrollan actividades sobre el sitio o página web. Este es el motivo de por qué su uso es tan importante en el desarrollo web. (KeepCoding, 2023)

2.14.4.1 Ventajas

- JavaScript es un lenguaje de programación necesario para el desarrollo de páginas web, pues integra muchas opciones para agregarle valor a tus páginas web. Además, está claro que, en cualquier momento, su uso se hace muy necesario.
- Ahora, una de las ventajas más importantes de porqué usar JavaScript es que trabaja en la parte del cliente, por lo que ahorra en ancho de banda. Por otro lado, acelera la ejecución del programa de código y del sitio web.
- Ya que una página tendría que funcionar en cualquier tipo de navegador (ya sea Chrome, Firefox o Safari) y eso es permitido gracias a JavaScript, ya que funciona en cualquier tipo de navegador, aunque existen problemas respecto a la lectura. Además, tiene toda la compatibilidad con muchas aplicaciones.
- Dado su popularidad entre el sector de los programadores, hay muchas personas interesadas en compartir su conocimiento sobre el lenguaje de programación, además de presentar sus propios proyectos en repositorios remotos, para el acceso a la información de todo el interesado.

2.14.4.2 Desventajas

- Como te mencionamos hace un momento en el punto sobre los navegadores, aunque sea compatible con la mayoría de lenguajes de programación, hay diferencias respecto al soporte del navegador.
- Para muchos expertos en la materia, el JavaScript DOM (o Document Object Model) puede llegar a ser muy lento al trabajar con HTML.

- Además, dado que en el campo se relaciona con elementos como HTML, muchas veces hay problemas con la depuración y la localización sencilla de errores en los programas de código.

2.14.5 HTML

“HTML es un lenguaje de marcado que se utiliza para el desarrollo de páginas de Internet. Se trata de las siglas que corresponden a HyperText Markup Language, es decir, Lenguaje de Marcas de Hipertexto”. (Pérez Porto & Gardey, 2021a)

2.14.5.1 Ventajas

Permite describir hipertexto, tiene un despliegue rápido, lo reconoce y admite cualquier tipo de explorador y permite archivos pequeños.

2.14.5.2 Desventajas

El diseño es más lento, tiene un lenguaje estático y las etiquetas son limitadas.

2.14.6 Css

Las hojas de estilo en cascada (CSS, Cascading Style Sheets) es un lenguaje de estilos que define el aspecto, la presentación y la posición que tendrán los diferentes elementos que componen una página web. El protagonismo que ha ido adquiriendo el uso de las hojas de estilo en el mundo del diseño web ha ido creciendo año tras año hasta convertirse en un mecanismo totalmente imprescindible. (Gomez, 2013)

2.14.6.1 Ventajas

- CSS juega un papel importante, al usar CSS simplemente debe especificar un estilo repetido para el elemento una vez y usarlo varias veces porque CSS aplicará automáticamente los estilos requeridos.
- La principal ventaja de CSS es que el estilo se aplica de manera consistente en una variedad de sitios. Una instrucción puede controlar varias áreas, lo que es ventajoso.

- Los diseñadores web necesitan usar pocas líneas de programación para cada página, lo que mejora la velocidad del sitio.
- La hoja en cascada no solo simplifica el desarrollo del sitio web, sino que también simplifica el mantenimiento, ya que un cambio de una línea de código afecta todo el sitio web y el tiempo de mantenimiento.
- Es menos complejo por lo que el esfuerzo se reduce significativamente.
- Ayuda a formar cambios espontáneos y consistentes.

2.14.6.2 Desventajas

- CSS, CSS 1 hasta CSS3, crean confusión entre los navegadores web.
- Con CSS, lo que funciona con un navegador puede no funcionar siempre con otro. Los desarrolladores web deben probar la compatibilidad, ejecutando el programa en varios navegadores.
- Existe una escasez de seguridad.
- Después de realizar los cambios, necesitamos confirmar la compatibilidad si aparecen. El cambio similar afecta a todos los navegadores.
- El mundo de los lenguajes de programación es complicado para los no desarrolladores y los principiantes. Los diferentes niveles de CSS, es decir, CSS, CSS 2, CSS 3, suelen ser bastante confusos.

2.14.7 Laravel

Laravel es un framework de PHP para ayudarnos en un tipo de desarrollo sobre aplicaciones escritas en este lenguaje de programación. Este framework o más bien podría llamarlo compañero de ahora en adelante, nos ayuda en muchas cosas al desarrollar una aplicación, por medio de sus sistema de paquetes y de ser un framework del tipo MVC

(Modelo-Vista-Controlador) da como resultado que podamos “despreocuparnos” (por así decirlo) en ciertos aspectos del desarrollo, cómo instanciar clases y métodos para usarlos en muchas partes de nuestra aplicación sin la necesidad de escribirlo y repetirlo muchas veces con lo que eso conlleva a la hora de modificar algo en el código. (Altube Vera, 2021)

2.14.7.1 Ventajas

Entre las ventajas de Laravel quizás la que más destaque es que es muy fácil de usar y aprender como framework de PHP, la famosa curva de aprendizaje, y si eso lo sumamos a que todas las demás como las características del anterior punto y que en el mercado es bastante usado y no está para nada mal pagado si tienes algo de experiencia, hace que en realidad esté framework sean casi todas ventajas más que desventajas si te tienes que decantar por aprender alguno o quieres usar alguno para algún proyecto.

2.14.7.2 Desventajas

Puede tener es que para algunas librerías y demás depende de Symfony que es otro framework del mismo estilo, dicen que con la curva de aprendizaje no tan liviana, aunque yo en particular he usado los dos y los he aprendido desde cero y si sabes cómo funciona un MVC, sabes cómo funcionan todos, lo único que tienes que hacer es mirar la documentación para saber cómo llama cada uno a sus métodos en particular y entre estos dos saber usar Blade o Twig, diferentes pero similares motores de plantillas que utilizan uno u otro framework.

2.14.8 Tarjetas RFID pasivas de solo lectura como transpondedor

Es un sistema de almacenamiento y recuperación de datos remoto que usa dispositivos denominados etiquetas, tarjetas, transpondedores o tags RFID. El propósito fundamental de la tecnología RFID es transmitir la identidad de un objeto (similar a un número de serie único) mediante ondas de radio. Compuesta por una antena, un transductor radio y un material encapsulado o chip. El propósito de la antena es permitirle al chip, el cual contiene la información, transmitir la información de identificación de la etiqueta. Existen varios tipos de

etiquetas. El chip posee una memoria interna con una capacidad que depende del modelo y varía de una decena a millares de bytes.

Figura 17

Tags de tarjetas de radio frecuencia



Nota. Elaboración Propia

2.14.9 Lector RFID RC522 como transceptor

El lector envía periódicamente señales para ver si hay alguna etiqueta en sus inmediaciones. Cuando capta una señal de una etiqueta (la cual contiene la información de identificación de esta), extrae la información y se la pasa al subsistema de procesamiento de datos. El lector RFID está compuesto por una antena, un transceptor y un decodificador.

Figura 18*Modulo lector RFID RC522**Nota. Elaboración propia*

2.14.10 *Modulo Node esp8266 como subsistema de procedimientos de datos*

NodeMCU ESP8266 es una plataforma de desarrollo similar a Arduino especialmente orientada al Internet de las cosas (IoT). La placa NodeMcu v2 ESP8266 tiene como núcleo al SoM ESP-12E que a su vez está basado en el SoC Wi-Fi ESP8266, integra además el conversor USB-Serial TTL CP2102 y conector micro-USB necesario para la programación y comunicación a PC. NodeMcu v2 ESP8266 está diseñado especialmente para trabajar montado en protoboard o soldado sobre una placa. Posee un regulador de voltaje de 3.3V en placa, esto permite alimentar la placa directamente del puerto micro-USB o por los pines 5V y GND.

NodeMCU viene con un firmware pre-instalado el cual nos permite trabajar con el lenguaje interpretado LUA, enviándole comandos mediante el puerto serial (CP2102). Las tarjetas NodeMCU y Wemos D1 mini son las plataformas más usadas en proyectos de Internet de las cosas (IoT). No compite con Arduino, pues cubren objetivos distintos.

Figura 19

Modulo Node v2 ESP8266



Nota. Elaboración propia

Especificaciones Técnicas

- Voltaje de Alimentación: 5V DC
- Voltaje de Entradas/Salidas: 3.3V DC (No usar 5V)
- Placa: NodeMCU v2 (Amica)
- Chip conversor USB-serial: CP2102
- SoM: ESP-12E (Ai-Thinker)
- SoC: ESP8266 (Espressif)
- CPU: Tensilica Xtensa LX3 (32 bit)
- Frecuencia de Reloj: 80MHz/160MHz
- Instruction RAM: 32KB
- Data RAM: 96KB
- Memoria Flash Externa: 4MB
- Wi-Fi Direct (P2P), soft-AP

- Stack de Protocolo TCP/IP integrado
- Corriente de fuga menor a 10uA
- Pulsador RESET y FLASH
- Leds indicadores: 2
- Dimensiones: 49*26*12 mm

2.14.11 Software Arduino

Arduino soporta varios lenguajes de programación de alto nivel derivados de C, haciendo de esto una ventaja para los diseñadores que trabajan en varios o en un solo entorno de desarrollo de programación.

(Arduino, 2016) Para poder trabajar desde el nivel programación del procesador debe descargarse el software que incluye las librerías necesarias para poder utilizar el lenguaje de manera completa. Otra ventaja es que este software puede descargarse desde el sitio web oficial de Arduino, ya que opera bajo licencia libre y está disponible a todo público. Su versión más reciente para todos los sistemas operativos es la versión Arduino 1.6

Figura 20

Logo de Arduino



Nota. Adaptado de Arduino.cc

2.14.12 *Librerías Arduino*

Las librerías proveen funcionalidad extra a nuestro sketch, por ejemplo: al trabajar con hardware o al manipular datos. Para usar una librería dentro de un sketch, puedes seleccionarla desde Sketch >Import Library (Importar Librería). Si deseas usar librerías que no vienen junto con Arduino, necesitarás instalarlas. Para hacerlo, descarga la librería y descomprimirla. Debería localizarse en una carpeta propia, y normalmente, contener dos archivos, uno con sufijo “.h” y otro con sufijo “.cpp”. Abre tu carpeta sketchbook de Arduino, si ya existe una carpeta llamada “libraries”, coloca la carpeta de la librería ahí dentro. Reiniciar el IDE de Arduino para encontrar la nueva librería en el menú Sketch >Import Library

CAPÍTULO III

MARCO

APLICATIVO

**INGENIERIA
DE SISTEMAS**
UNIVERSIDAD PUBLICA DE EL ALTO

3. MARCO APLICATIVO

3.1 INTRODUCCIÓN

En esta etapa el desarrollo del proyecto Sistema de Información web para la administración y control de membresía mediante la tecnología RFID caso: Asociación Misión Luz de Vida. Se aplicó las distintas fases del modelo UWE (Ingeniería Web basado en UML), las fases de la metodología son: análisis de requerimiento, diseño conceptual, diseño de navegación y diseño de presentación que presentan diversos diagramas y esquemas.

3.2 DESARROLLO MODELADO MEDIANTE UWE

3.2.1 *Captura, análisis y especificación de requisitos*

Como base de requerimientos la ingeniería de requerimientos funcionales y no funcionales, se obtiene la información clara y precisa, en este sentido para el presente proyecto se realizaron las siguientes actividades que se detalla bajo la siguiente tabla.

Tabla 7*Captura, análisis y especificación de requisitos*

HERRAMIENTA	CARACTERÍSTICAS
Observación	En la Asociación Misión Luz de Vida se pudo observar las dificultades que ocasiona a la hora de registrar las asistencias, el control de mensualidades, reuniones y trabajos que se realiza en la institución.
Documentación	Se reviso en dirección los documentos que se manejan dentro de la Asociación.
Entrevista	Se realizo la entrevista con el personal involucrado dentro de la Asociación Misión Luz de Vida. Quien expresa la necesidad de un Sistema Web para mejorar el registro de los patrocinadores. • Pastores • Directora • Patrocinio • Tutores • Participantes • Padres de familia

Nota. Se denota en la tabla del cómo se realizó la captura de requerimientos.

3.2.2 Requerimientos Funcionales

Un requerimiento funcional define una función del sistema de software o sus componentes, Los requerimientos funcionales (RF), se enfocan principalmente en lo que necesita el Sistema para los diferentes procesos, se detalla en la siguiente tabla:

Tabla 8

Requerimientos funcionales

ROL	DESCRIPCIÓN	FUNCIÓN
RF_1	Base de datos	Almacenamiento de datos, gestor de base de datos.
RF_2	Acceso al Sistema	Acceder al sistema por roles de usuarios (director, patrocinio, tutores y Participantes/beneficiarios).
RF_3	Gestionar Usuarios	Registrar a los usuarios con datos necesarios de la persona asignando el dato que tiene, modificar, activar y desactivar a los participantes/beneficiarios.
RF_4	Gestionar roles	Registrar y asignar los permisos y roles a los distintos cargos.
RF_5	Gestionar permisos	Asignar los permisos dependiendo del rol a los usuarios.
RF_6	interfaz amigable	La interfaz gráfica debe ser fácil y entendible para los usuarios.
RF_7	Visualización de la página web	Visualización y funcionamiento correcto en los diferentes navegadores.
RF_8	perfil del usuario	Muestra la información, actualización de foto de perfil y modificación de la contraseña.

ROL	DESCRIPCIÓN	FUNCIÓN
RF_9	Registro y Control de los procesos del sistema	Registra todos los procesos que realizan los usuarios.
RF_10	Registro de control de cobros de cuotas/mensualidades	El administrador (director y patrocinio) tienen el acceso a la apertura de cobros de las mensualidades que se realizaran mensualmente y costo total en cuanto se está recaudando.
RF_11	Generar reportes del control de Asistencia	Permite mostrar la información de asistencias en el transcurso del año
RF_12	Generar reportes de las reuniones	Permite mostrar las reuniones realizadas una vez al mes.
RF_13	Generar reportes de las mensualidades pendientes	Permite mostrar las cuotas/mensualidades realizadas durante la gestión y deudas pendientes.
RF_14	Generar un reporte General	Permite listar un reporte general de Asistencias a Cultos, reuniones y mensualidades.

Nota. Se realizo la identificación de los requerimientos funcionales.

3.2.3 Requerimientos No Funcionales

Son restricciones de los servicios o funciones ofrecidos por el sistema. Incluyen restricciones de tiempo, sobre el proceso de desarrollo, estándares, etc. Los requerimientos no funcionales (RFN), son cualidades y características del Sistema que el usuario puede visualizar, el sistema proporciona que no se relacionan de forma directa con el comportamiento funcional.

Tabla 9*Requerimientos no funcionales*

ROL	FUNCIÓN
RNF_1	El Sistema debe de visualizarse y funcionar correctamente en cualquier dispositivo que tenga disponibilidad la conexión a internet desde los navegadores modernos como ser: Chrome, Mozilla, Edge, Brave y otros.
RNF_2	El sistema debe tener seguridad en el acceso a la información del sistema.
RNF_3	El sistema debe tener un entorno amigable con el usuario, para un fácil entendimiento de las funcionalidades y operaciones que se pueden realizar.
RNF_4	El acceso debe ser restringido mediante autenticación de usuario y contraseña.
RNF_5	El sistema debe tener pestañas de navegación para tener un mejor seguimiento de las páginas web.
RNF_6	Los datos modificados se actualizan para todos los usuarios
RNF_7	El Sistema tendrá que encriptar las contraseñas de los usuarios para mayor seguridad.
RNF_8	La sesión tendrá un determinado periodo de tiempo activo para que después el usuario vuelva a iniciar sesión.
RNF_9	El URL del sistema debe ser encriptado para que No pueda ser visible para posibles accesos no autorizados.
RNF_10	El sistema debe permitir agregar nuevos módulos y funciones a predicción futuro actualización a grandes escalas de seguridad confiabilidad.
RNF_11	El sistema debe de funcionar en base a la ISO 25000 Métricas de calidad.

Nota. Se realizo la identificación de los requerimientos No funcionales

3.2.4 *Diseño del Sistema*

Se utilizará las siguientes fases: Descripción de los autores, diagramas de casos de uso, diagramas de clase, modelo conceptual y modelo navegacional.

3.2.4.1 Descripción de actores.

Nos permitió conocer a las personas involucradas en el proceso de las funciones: asistencia, revisión y reporte de la información a las actividades programadas dentro de la Asociación Misión Luz de Vida.

Tabla 10

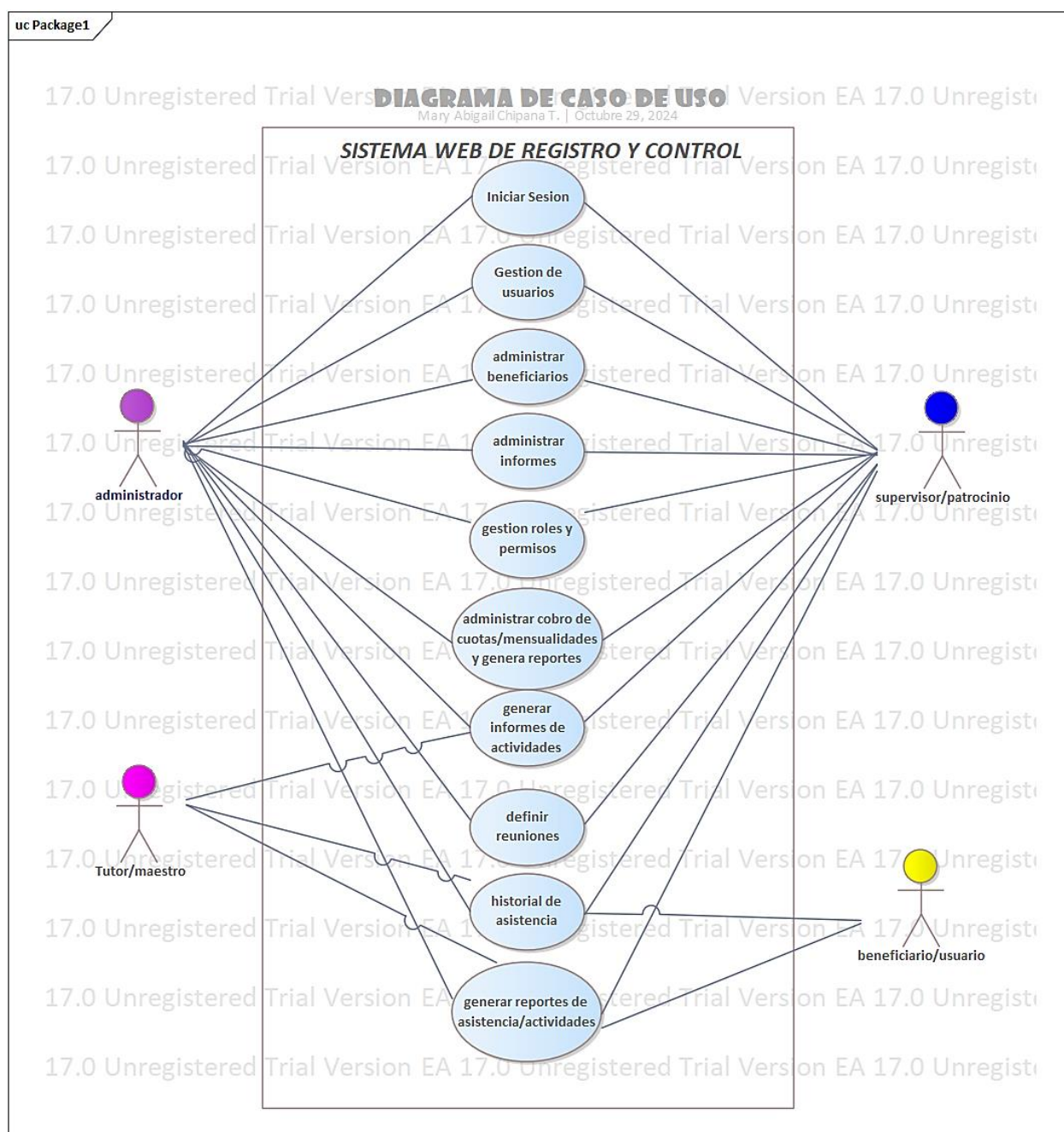
Descripción de actores

ROLES	DESCRIPCIÓN
Administrador/director	Es encargado de administrar el sistema, realizando gestiones de usuarios: El rol como administrador cumple las funciones de asignar roles y permisos. • Nombre de Usuario, contraseña • Asignar tarjeta RFID • crear nuevas Reuniones • Nivel de acceso
Administrador o responsable de Patrocinio	Para un control seguro y diferenciado, se encarga de las configuraciones del sistema. Su funcionalidad es: • Asignar tarjeta RFID • Asignar las nuevas reuniones • Cobro de mensualidades/cuotas • Asignar o controlar las Asistencias. • Generar reportes
Tutores/maestros	Cumple la función de generar reportes de asistencias a Cultos, reuniones, mensualidades.
Usuario (Patrocinador/participante)	Se encargará de visualizar su reporte de todas las actividades que realiza.

Nota. Se realizó la descripción de cada uno de los actores y roles.

3.2.4.2 Diagrama de caso de uso general

En este punto se plasma el análisis de requerimientos de sistema mediante el diseño de casos de uso, que describe el comportamiento del sistema frente a las acciones de los actores, funcionamiento del sistema y además elementos que permiten la abstracción del problema. A continuación, se realiza el modelamiento donde se puede apreciar como interactúan los actores sobre los casos de uso.

Figura 21*Diagrama de caso de uso general*

Nota. Se realizó la descripción del diagrama de caso de uso general.

Tabla 11

Descripción de actores de casos de uso Asociación Misión Luz de Vida

ACTORES	DESCRIPCIÓN
Administrador	Encargado de administrar el sistema, con acceso a módulos, gestionar usuarios, administrar informes, historial de asistencia, administrar cobro de mensualidades, definir reuniones.
Supervisor/patrocinio	Encargado de administrar a los beneficiarios o usuarios, teniendo el acceso a gestión de usuarios, administrar beneficiarios, administrar cobro de mensualidades, generar informes de actividades, definir reuniones, historial de asistencia.
Tutor/maestro	Podrá ingresar al Sistema y generar informes de actividades, historial de asistencia, y generar informes del historial de asistencia y actividades.
beneficiario/usuario	Podrán ingresar al sistema para poder ver el historial de asistencia y generar reportes de actividades e historial de asistencia.

Nota. Elaboración de casos de uso

3.2.4.3 Especificaciones de casos de uso

A continuación, se realizará la descripción de las especificaciones de la administración del sistema.

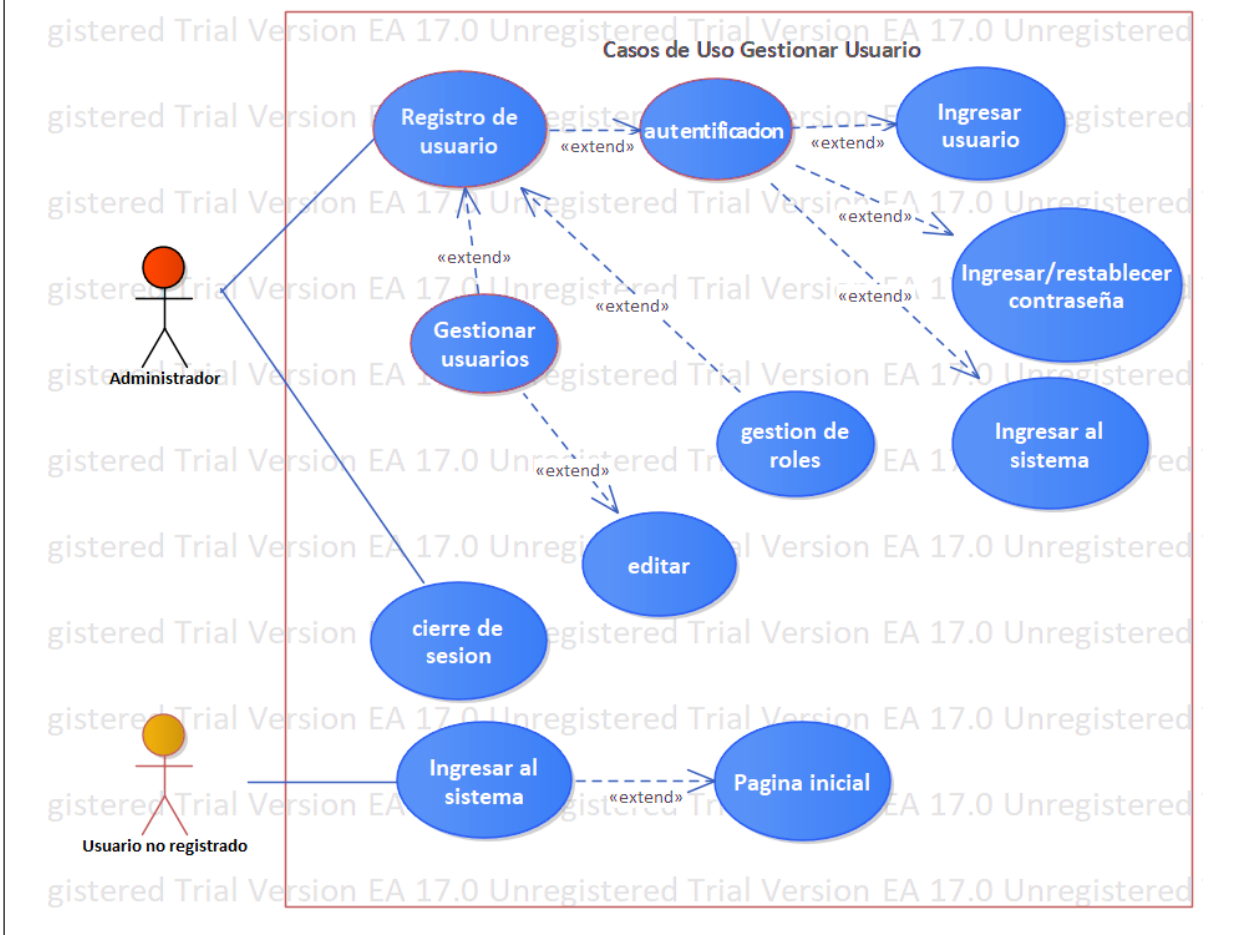
Figura 22

Diagrama de caso de Uso - Ingreso al sistema

uc Package1

DIAGRAMA DE CASOS DE USO - ESPECIFICO

Mary Abigail Chipana T. | Octubre 29, 2024



Nota. Representa como un usuario ingresa al sistema (elaboración propia)

Tabla 12*Descripción de actores de caso de uso autenticación*

CASO DE USO	AUTENTICACIÓN
Número del caso de uso:	DCU01
Descripción	Describe el proceso que permite identificar, autenticar y autorizar el ingreso al sistema
Precondiciones	El usuario tiene que estar registrado en el sistema Nombre de usuario Contraseña de usuario
Actores	Administrador/director supervisor/patrocinio tutor/maestro beneficiario/patrocinador
Secuencia	Usuario ingresa a la página inicial al sistema. Ingresa nombre usuario. Ingresa contraseña de usuario. Autenticación de datos ingresados y autorización de ingreso al Sistema.
Secuencia alternativa	Ingresa a la página inicial del sistema. Ingresa nombre de usuario. Ingresa contraseña de usuario. Autenticación de datos ingresados y si no accede, el sistema muestra un mensaje de error de datos.

Nota. Elaboración de actores de casos de uso.

Tabla 13

Descripción de actores de caso de uso - Ingreso al sistema

CASO DE USO	INGRESAR AL SISTEMA
Descripción	Permite el ingreso a la interfaz inicial al Sistema
Precondiciones	La dirección web de acceso al sistema
Actores	Administrador/director, supervisor/patrocinio, tutor/maestro, beneficiario/patrocinador
Secuencia	Ingreso del usuario a la página principal
Secuencia alternativa	ingreso al portal web para elegir

Nota. Elaboración de casos de uso del ingreso al sistema

Tabla 14

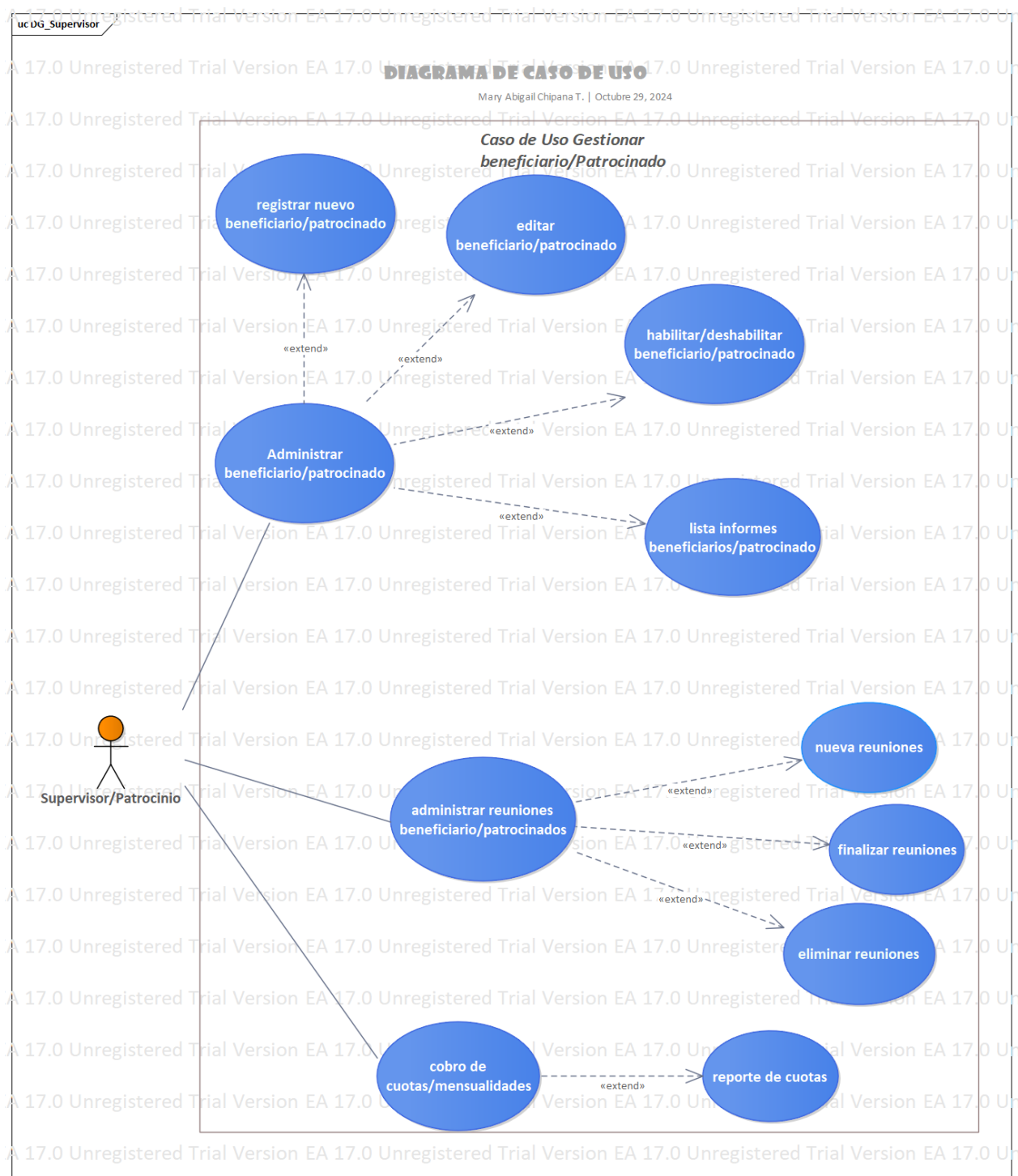
Descripción de actores de caso de uso - cierre de sesión

CASO DE USO	CIERRE DE SESIÓN
Descripción	Describe el proceso que permite el cierre de la sesión abierta por el usuario.
precondiciones	Ingreso al sistema
actores	Administrador/director, supervisor/patrocinio, tutor/maestro, beneficiario/patrocinador
secuencia	Usuario ingresa a la página inicial al sistema. Ingresa nombre usuario. Ingresa contraseña de usuario. Autenticación de datos ingresados y autorización de ingreso al Sistema.
secuencia alternativa	Ninguna

Nota. Elaboración propia

Figura 23

Casos de uso Supervisor/Patrocinio



Nota. Se denota el proceso de supervisor/patrocinio.

Tabla 15*Descripción Caso de uso administrar beneficiario/patrocinio*

CASO DE USO		CIERRE DE SESIÓN	
Número del caso de uso		DCU02	
Descripción	Describe el proceso que permite administrar a los beneficiarios/patrocinados de la Asociación Misión Luz de Vida.		
Precondiciones	Proceso de autenticación (login) del usuario en el sistema.		
Actores	Administrador/director, supervisor/patrocinio. Selecciona la opción administrar beneficiario/patrocinado.		
Secuencia	• Selecciona la opción agregar beneficiario/patrocinado. • Registra los datos requeridos. • Registra y Asigna una tarjeta con el cual registrara su asistencia. • Crea nuevas reuniones. • Registra los cobros de las mensualidades. • El Sistema muestra la lista de los beneficiarios/patrocinados registrados en la base de datos. • Selecciona la opción de editar para modificar los datos.		
Secuencia alternativa	Ninguna		

Nota. Elaboración de caso de uso del beneficiario/patrocinio

Tabla 16

Descripción caso de uso administrar horario de asistencia beneficiarios/patrocinados

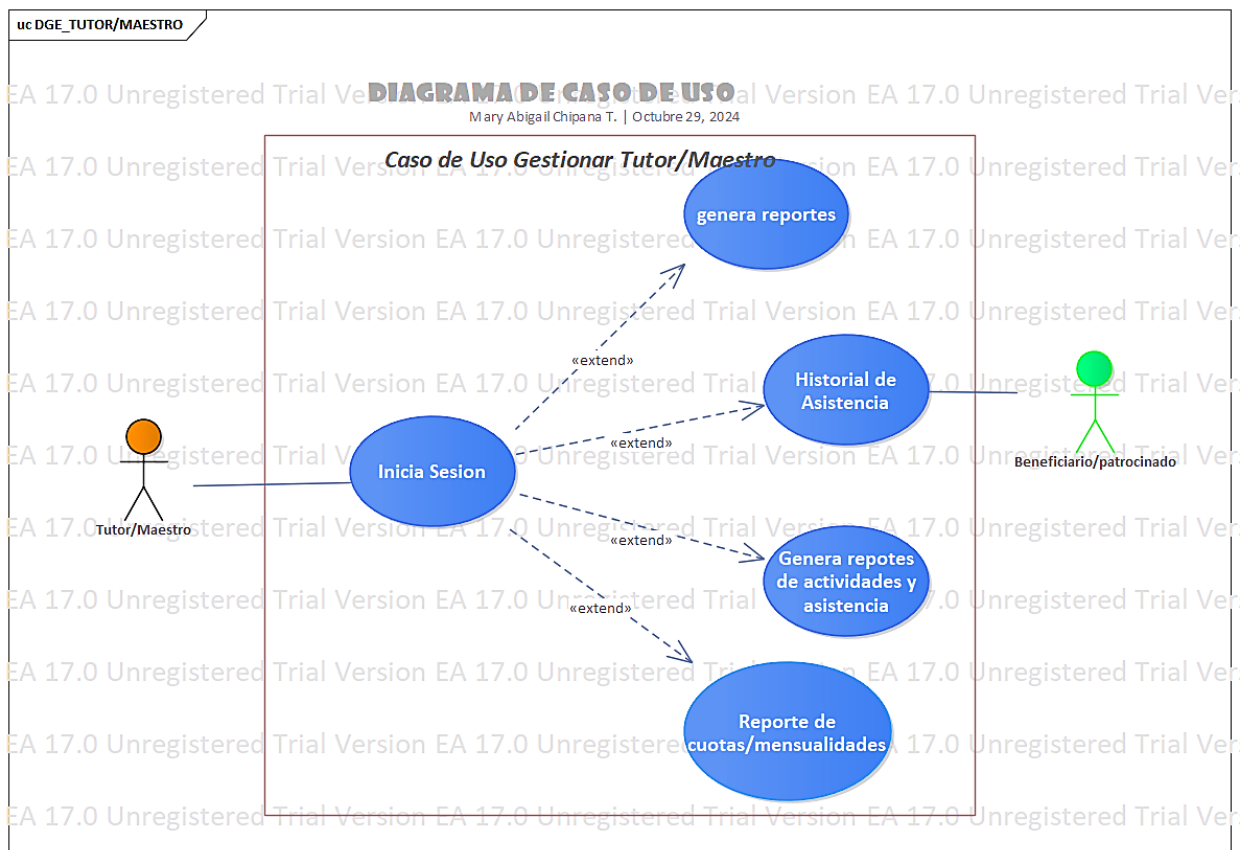
CASO DE USO	ADMINISTRAR EL HORARIO DE BENEFICIARIOS/PATROCINADOS
Descripción	Describe el proceso que permite administrar la carga horaria o asistencias al mes de los beneficiarios/patrocinados de la Asociación Misión Luz de Vida.
Precondiciones	Proceso de autenticación (login) del usuario en el sistema

CASO DE USO	ADMINISTRAR EL HORARIO DE BENEFICIARIOS/PATROCINADOS
Actores	Administrador/director, supervisor/patrocinio
	Selecciona la opción asistencia beneficiarios/patrocinados.
	El Sistema muestra la lista de los beneficiarios/patrocinados en la base de datos.
	• Selecciona Control de Asistencia. • El sistema muestra el historial de asistencia del beneficiario/patrocinado: • Selecciona la opción agregar asistencia. Llena los datos requeridos
Secuencia	Selecciona guardar datos. • Selecciona la opción editar para modificar los datos. Selecciona guardar datos o cancelar. • Selecciona la opción eliminar. Confirma la eliminación de un registro de asistencia.
secuencia alternativa	Ninguna

Nota. Caso de uso administrar horario

Figura 24

Diagrama de caso de uso – administrar tutor/maestro



Nota. Se denota el proceso de tutor/maestro.

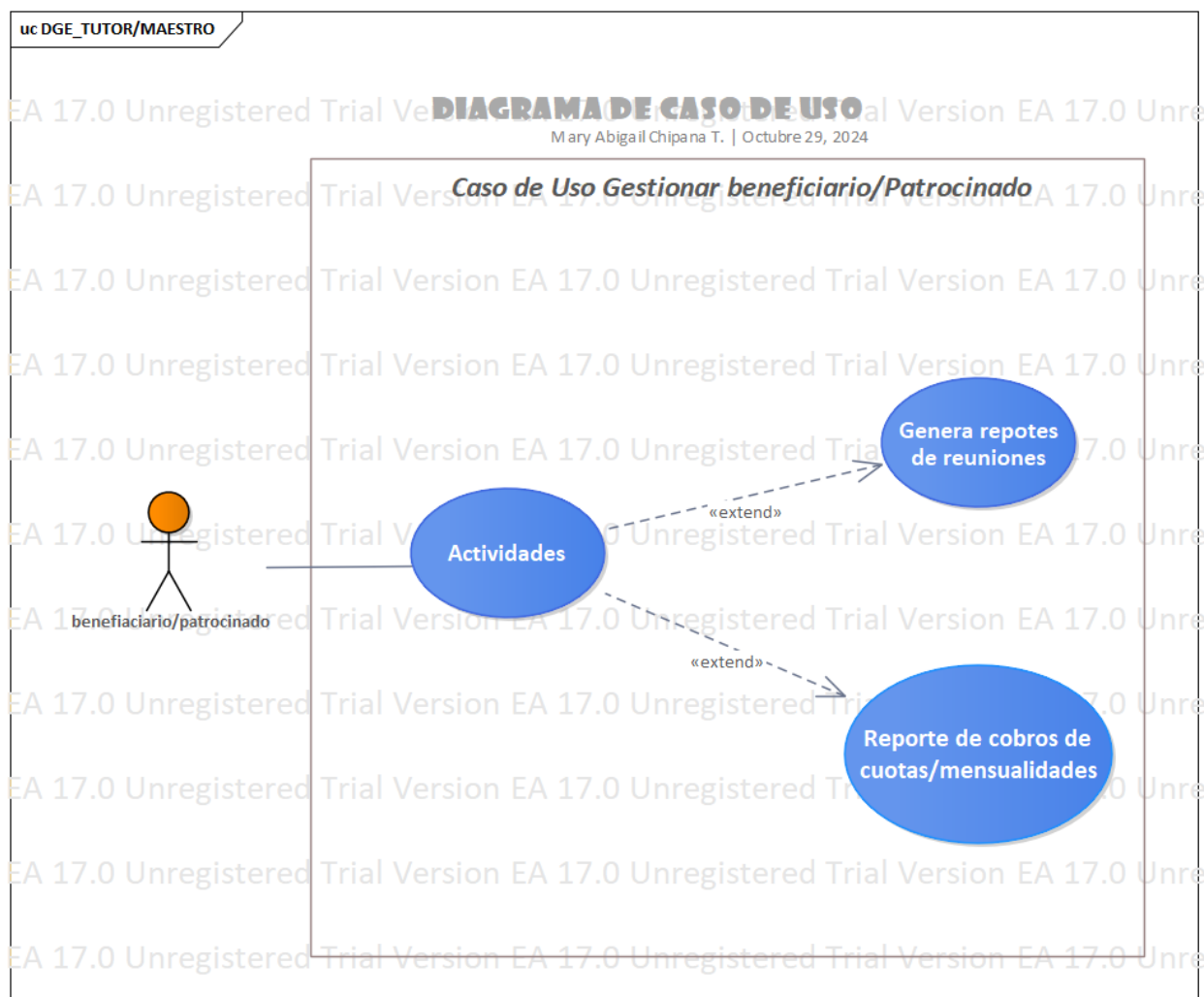
Tabla 17*Descripción del Tutor/Maestro*

CASO DE USO	DESCRIPCIÓN DE TUTOR/MAESTRO
Descripción	Describe el proceso que permite visualizar el control de las distintas actividades de beneficiarios/patrocinados de la Asociación Misión Luz de Vida.
precondiciones	Proceso de autenticación (login) del usuario en el sistema
actores	tutor/maestro
secuencia	El Sistema muestra la lista de reportes de las actividades que se realizan, como ser: • Historial de Asistencias a Cultos. • Reporte del control de Asistencia a reuniones. • Visualizar reportes de las cuotas o mensualidades
secuencia alternativa	Ninguna

Nota. Este caso de uso denota el proceso del tutor/maestro.

Figura 25

Caso de uso beneficiario/patrocinado



Nota. Caso de uso donde se realiza el control del beneficiario/patrocinado

Tabla 18*Descripción de beneficiario/patrocinado*

CASO DE USO	ADMINISTRAR BENEFICIARIO/PATROCINADO
Descripción	Describe el proceso que permite visualizar las distintas actividades de beneficiarios/patrocinados de la Asociación Misión Luz de Vida.
Precondiciones	Proceso de autenticación (login) del usuario en el sistema, el usuario tiene que haber iniciado sesión en el sistema para poder gestionar a los productos.
Actores	beneficiario/patrocinado
Secuencia	El Sistema muestra la lista de reportes individuales de las actividades que se realizaron, como ser: • Historial de Asistencias a Cultos. • Reporte del control de Asistencia a reuniones. • Visualizar reportes de las cuotas o mensualidades
Secuencia alternativa	Ninguna

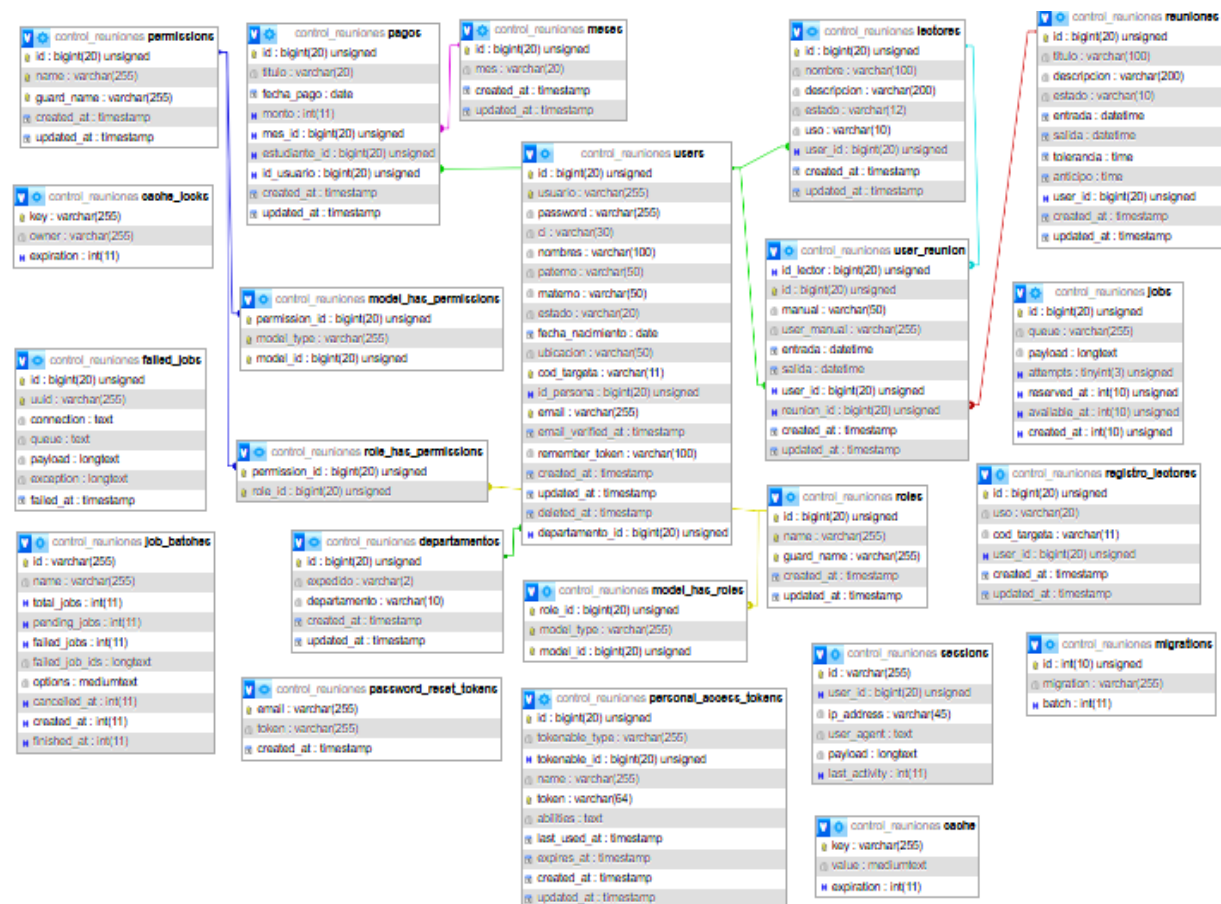
Nota. Diagrama de Caso de uso, describe del beneficiario/patrocinado

3.2.4.4 Diseño conceptual

En esta etapa, se construirá un modelo Conceptual, A través de este modelo, se podrá observar con claridad cuáles son los alcances del proyecto, para luego trabajar sobre ellos, la presentación grafica del Modelo Conceptual es la siguiente.

Figura 26

Diseño conceptual



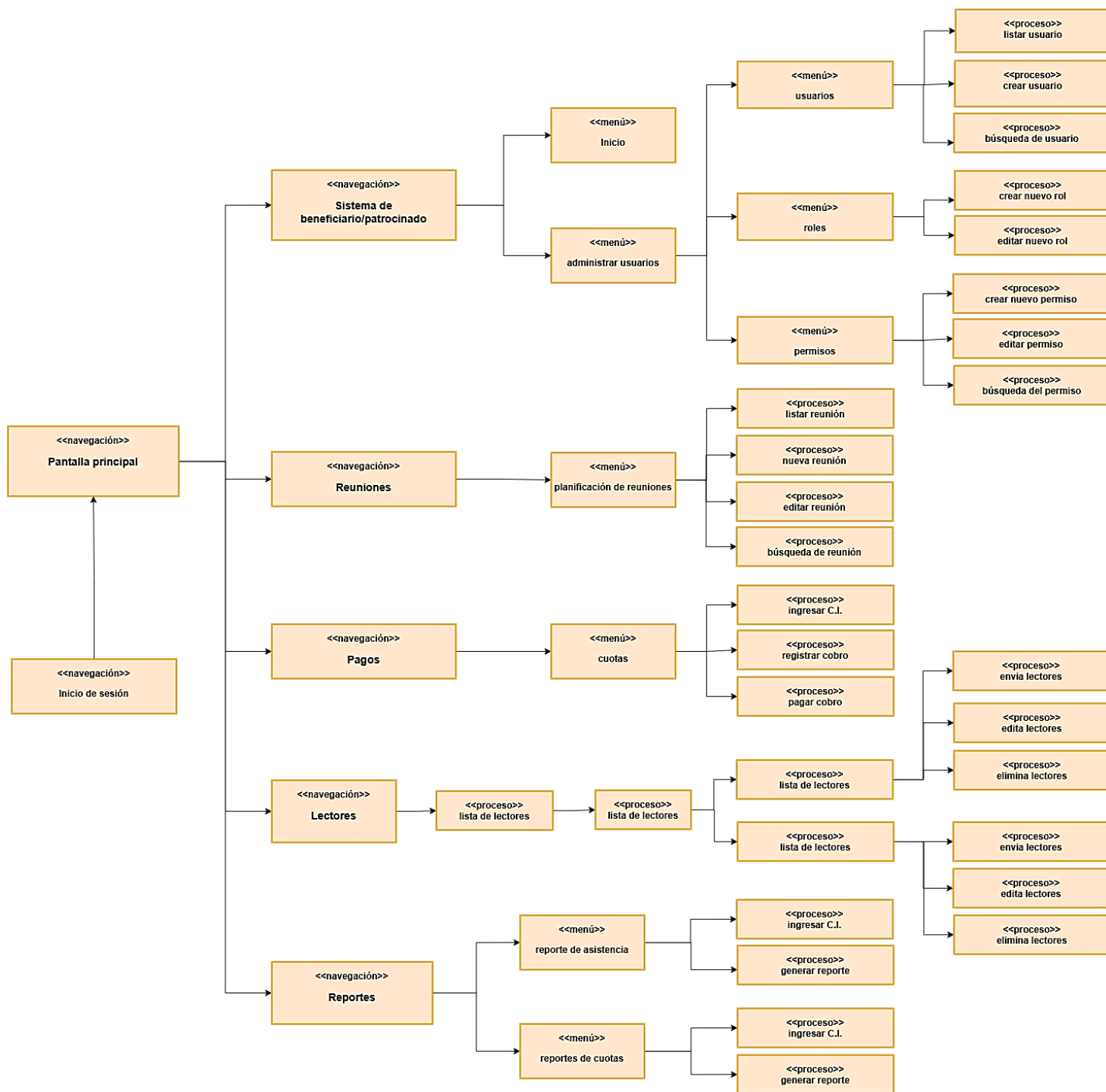
Nota. Se realizo el modelo conceptual de la base de datos

3.2.4.5 Modelo de navegación

Se muestra a continuación el diseño del sistema de interfaz de toma en cuenta una interfaz amigable, visible que permite seleccionar acciones fácilmente.

Figura 27

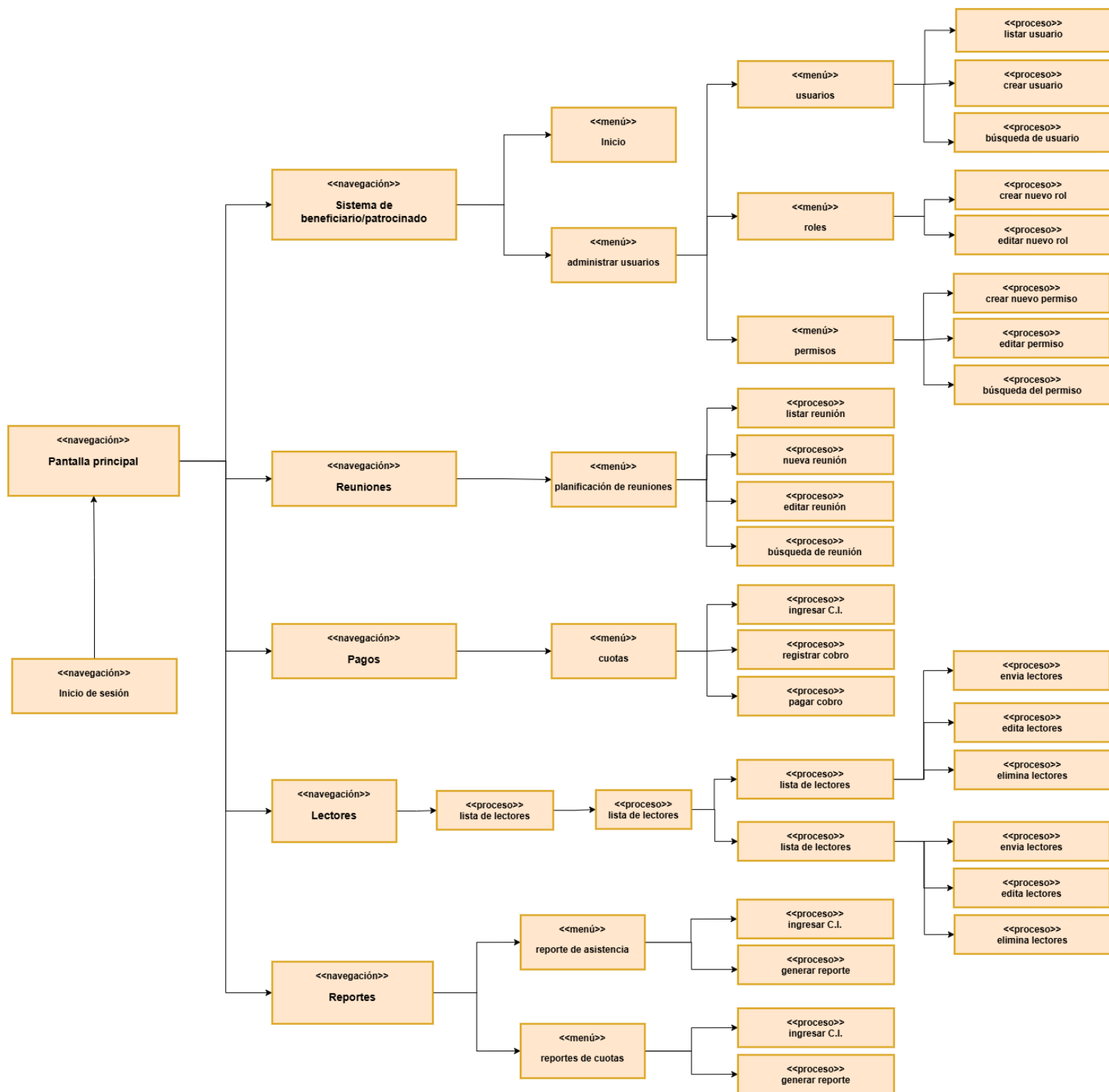
Modelo de navegación - Administrador



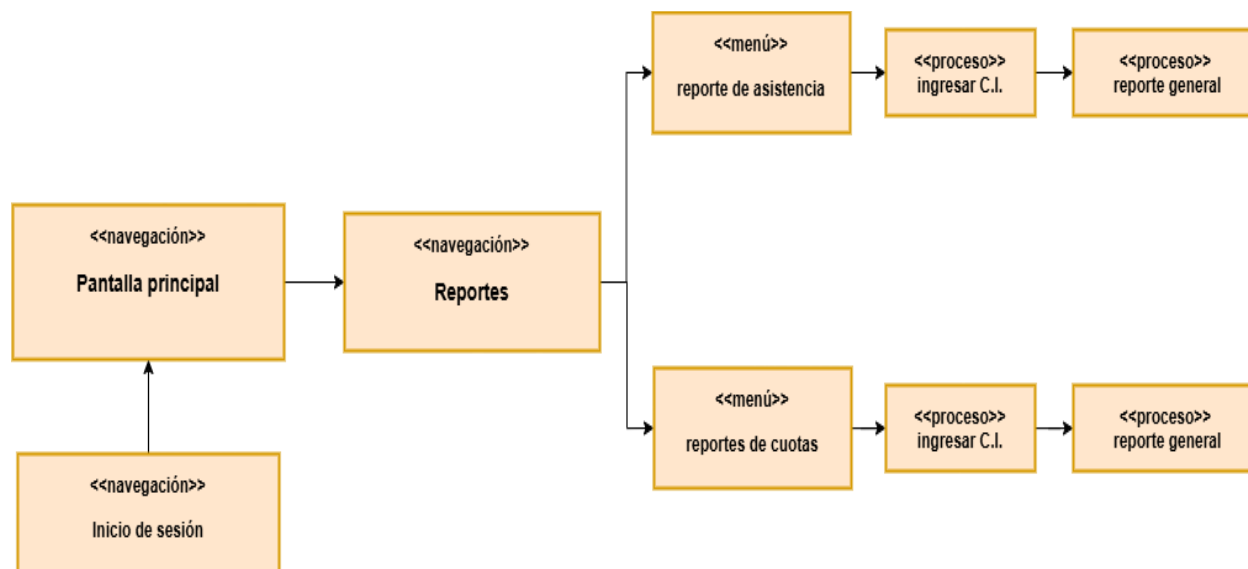
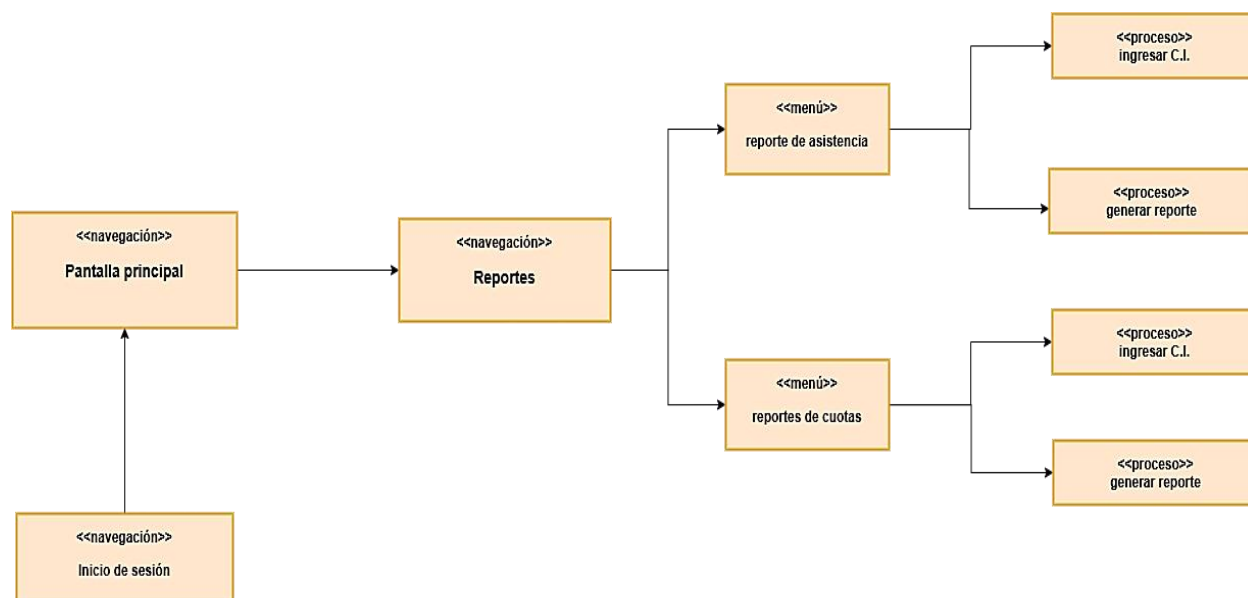
Nota. Modelo navegacional del rol administrativo

Figura 28

Modelo de navegación - Supervisor/Patrocinio



Nota. Modelo navegacional del rol Supervisor/Patrocinio

Figura 29*Modelo de navegación - Tutor**Nota. Modelos de navegación - Tutor***Figura 30***Modelo de navegación – Beneficiario/Patrocinado**Nota. Modelo navegacional de los beneficiarios/participantes*

3.2.4.6 Modelo de presentación

A continuación, se muestra los modelos de presentación para el Sistema de la Asociación Misión Luz de Vida, según UWE propone para la construcción de las páginas en forma de bosquejo. En el cual muestra como los usuarios podrán acceder al sistema mostrando los menús correspondientes según el tipo de usuario.

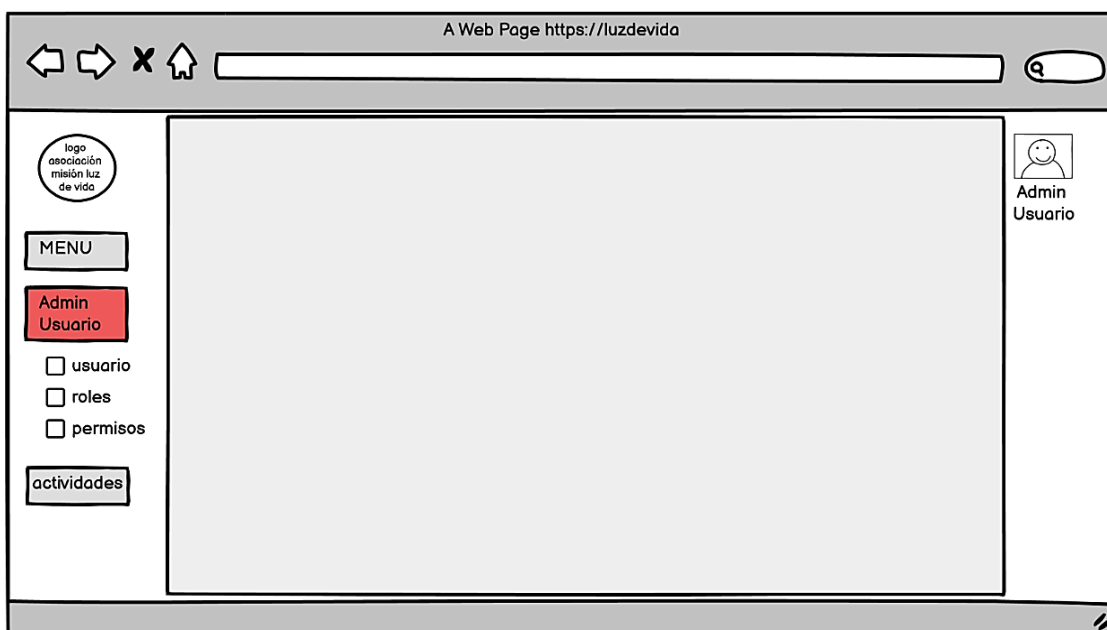
Figura 31

Modelo de presentación página inicial del Sistema – Login

El diagrama muestra la interfaz de usuario para la página de inicio de sesión. En la parte superior, hay una barra de encabezado verde con el texto "Iglesia Presbiteriana Luz de Vida" a la izquierda y un círculo verde con el texto "logo Iglesia" a la derecha. En el centro de la página, hay un recuadro blanco con el texto "logo Asociación Misión Luz de Vida". Debajo de este recuadro, hay tres campos de entrada: un campo verde con el texto "usuario", un campo verde con el texto "contraseña", y un botón rectangular con el texto "Iniciar Sesión".

Figura 32

Modelo de presentación del sistema – Inicio

**Figura 33**

Modelo de presentación – sistemas

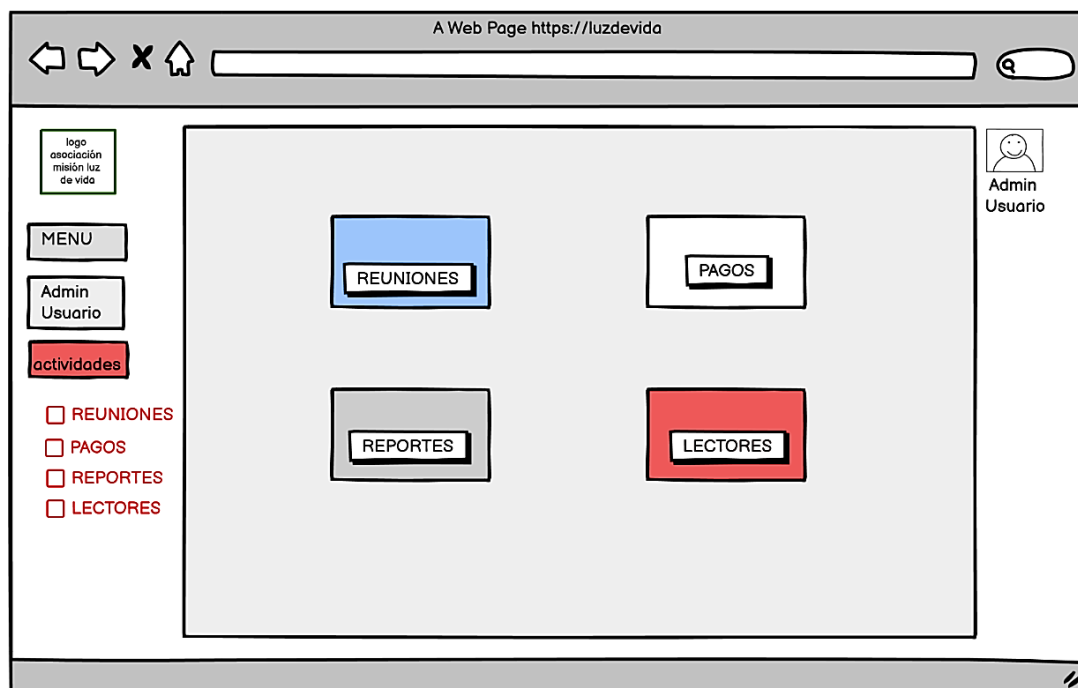


Figura 34

Modelo de presentación – Reuniones

Reuniones Luz de Vida

logo luz de vida

MENU

Admin Usuario

actividades

REUNIONES

► Planificación de reuniones

PAGOS

REPORTES

LECTORES

LISTA DE REUNIONES

nuevo

Admin Usuario

N	TITULO	DESCRIPCIÓN	ESTADO	REUNIÓN	ANTICIPO	TOLERANCIA	ACCIÓN
							termina visualiza reporte
							termina visualiza reporte
							termina visualiza reporte
							termina visualiza reporte
							termina visualiza reporte

Figura 35

Modelo de presentación del sistema - Pagos

PAGOS Luz de Vida

logo luz de vida

MENU

Admin Usuario

actividades

REUNIONES

PAGOS

► Cuotas

REPORTES

LECTORES

PAGAR CUOTAS

CI ESTUDIANTE

MESES

MONTO

Pagar Cuota

Figura 36

Modelo de presentación del sistema - Reportes Asistencias

The interface for 'Reportes Asociación Misión Luz de Vida' features a dark green header with the title. On the left, a sidebar contains a logo, a 'MENU' button, and several menu items: 'Admin Usuario', 'actividades', 'REUNIONES', 'PAGOS', 'REPORTES' (highlighted in green), and 'LECTORES'. Under 'REPORTES', there are two sub-items: 'Reporte de Asistencia' (highlighted with a green arrow) and 'Reporte de Cuotas'. The main content area is titled 'REPORTE DE ASISTENCIA' in a red box. It includes a text input field for 'CI ESTUDIANTE', two date selection fields labeled 'Fecha de inicio' and 'Fecha de fin' (each with a calendar icon), and a 'Generar reportes' button.

Figura 37

Modelo de presentación del Sistema - Reporte de Cuotas

The interface for 'Asociación Misión Luz de Vida' has a dark green header with the title. The sidebar is identical to Figure 36, with 'REPORTES' highlighted in green and 'Reporte de Cuotas' highlighted with a green arrow. The main content area is titled 'REPORTE DE CUOTAS' in a red box. It features a text input field for 'CI ESTUDIANTE' and a 'Generar reportes' button, which is highlighted with a cyan background.

Figura 38

Modelo de presentación del sistema - Lectores



3.2.4.7 Modelo de implementación

Figura 39

Login

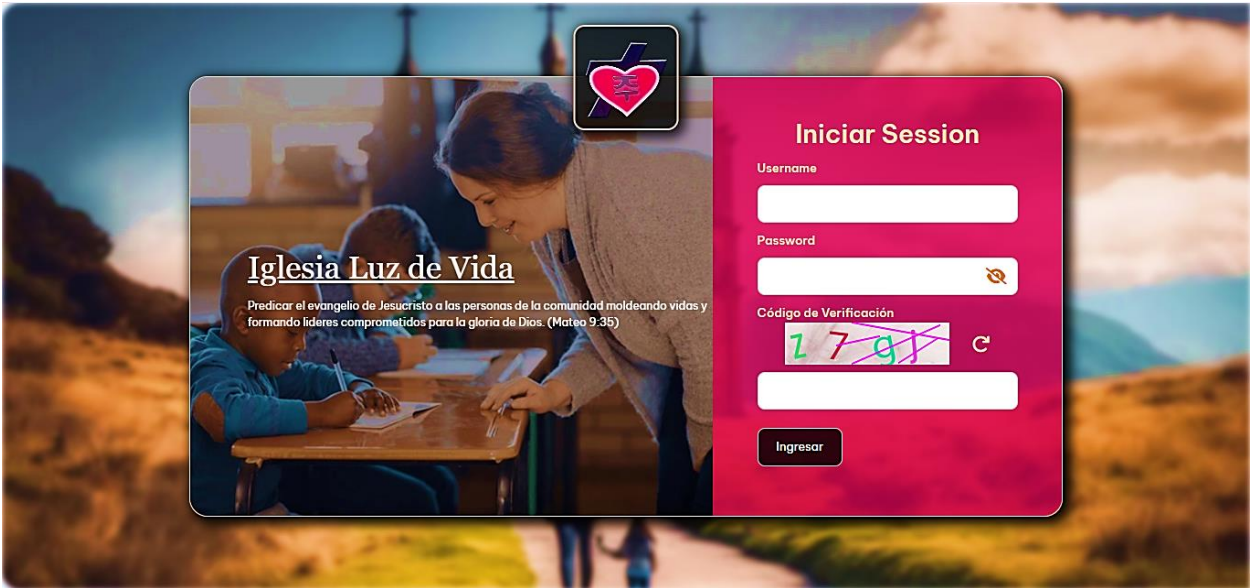


Figura 40

Menú Inicio

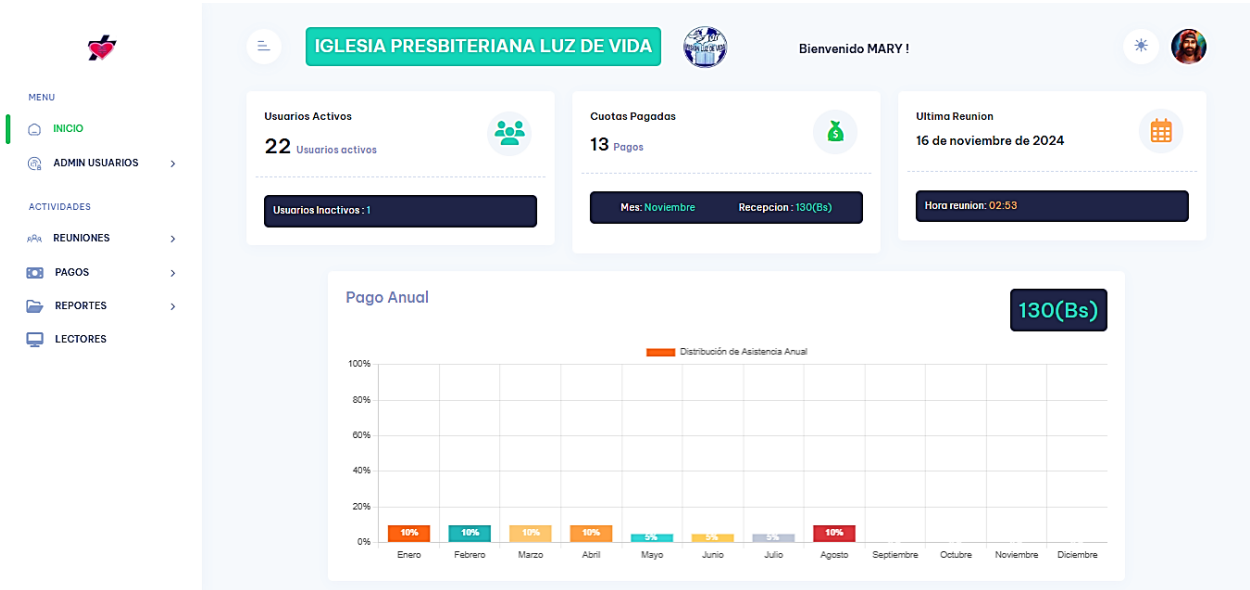


Figura 41

Administrar usuarios



Figura 42

Menú de Roles



Figura 45

Módulo de pagos o cobros de cuotas

**Figura 46**

Generar reporte de cuotas



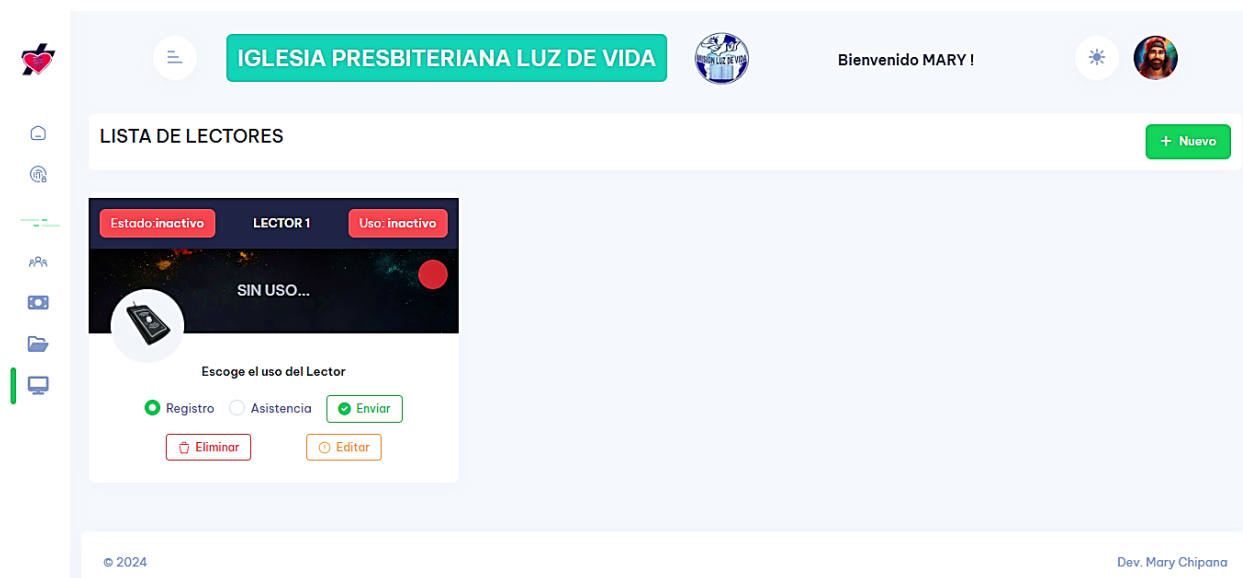
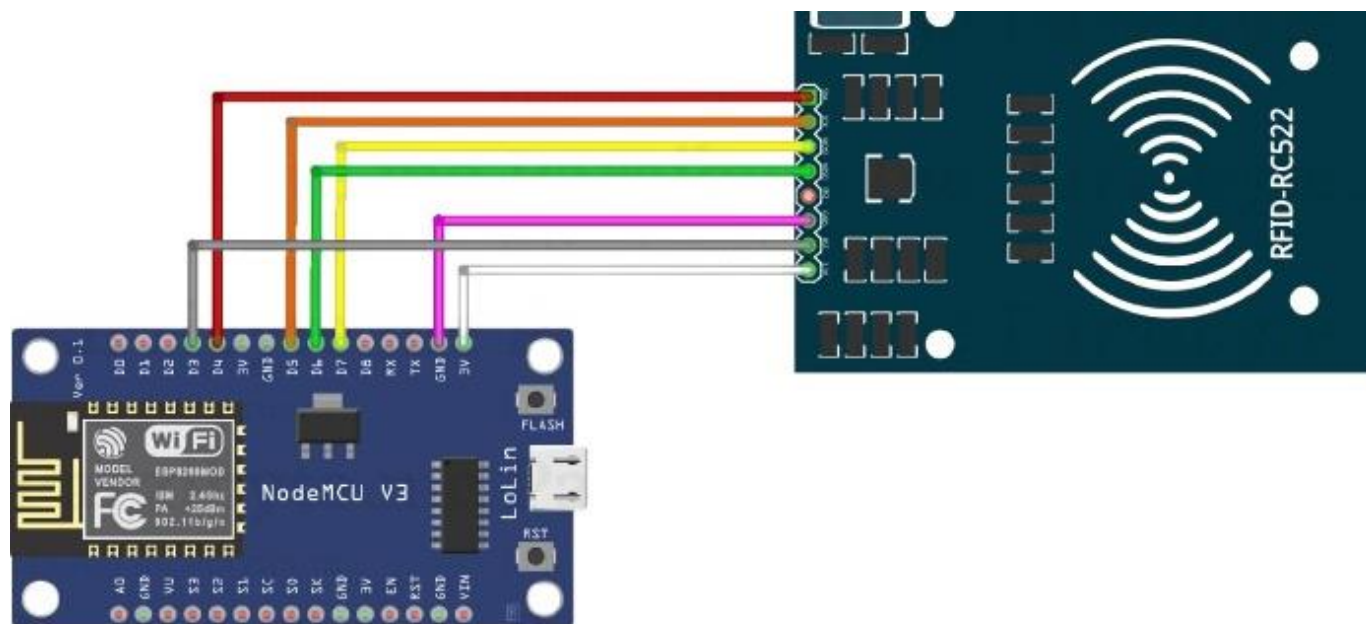
Figura 47*Administrar lector***Figura 48***Circuito del lector de radio frecuencia*

Figura 49

Codificación del modelo ESP para envío de datos al sistema



```

lectorcrdid$
//#define LED2 D2 // respuesta correcta
#define LED3 D8 // respuesta incorrecta
#define RST_PIN D3
#define SS_PIN D4

MFRC522 reader(SS_PIN, RST_PIN);
MFRC522::MIFARE_Key key;

char ssid[100] = "JAN";
String password = "hermanitosChipanas123";
String protocolo="http://";
String host="192.168.0.18";
String recurso="/control_reuniones/public/api/codigo_targeta";
String url=protocolo + host + recurso;
int port=80;
String serial = "";
byte cont = 0;
byte max_intentos = 50;
void setup()
{
    pinMode(LED1,OUTPUT);
    pinMode(LED3,OUTPUT);
    Serial.begin(9600); // Inicializa la comunicación serial y espera hasta que esté lista
    while (!Serial)
    {
        // No hay nada aquí. Solo espera a que la comunicación serial esté presente
    }

    SPI.begin(); // Inicia el bus SPI

    reader.PCD_Init(); // Inicializa el lector MFRC522
    // Solo espera unos segundos...
    delay(4);
    for (byte i = 0; i < 6; i++)
    {
        key.keyByte[i] = 0xFF; // keyByte está definido en la estructura "MIFARE_Key" en el archivo .h de la biblioteca
    }
    Serial.println("Inicio!");
}

```

Nota. Elaboración del modelo ESP

Figura 50*Codificación para el envío de datos al sistema*

```

void enviarDatos(){
  HTTPClient http;
  WiFiClient client;
  String respuesta="";
  http.begin(client,url); // Cargamos la url a ser conectada
  http.addHeader("Content-Type", "application/x-www-form-urlencoded"); // cargamos la cabecera para saber que tipo de informacion se va enviar
  String postData = "dato="+serial+"&lector=1"; // los datos que se envian
  //String postData = "dato1="+ dato1 + "&dato2="+ dato2 + "&dato3="+ dato3;
  int httpCode=http.POST(postData);// funcion para enviar los datos
  respuesta = http.getString(); // obtenemos la respuesta del servidor
  Serial.println("Código de estado HTTP: " + String(httpCode));
  Serial.println("Respuesta servidor:"+respuesta);
  if(httpCode != 200){
    tonoIncorrecto();
    digitalWrite(LED3,HIGH);
    delay(1000);
    digitalWrite(LED3,LOW);
    serial="";
    return;
  }
  if(respuesta != "correcto"){

    tonoIncorrecto();
    digitalWrite(LED3,HIGH);
    delay(1000);
    digitalWrite(LED3,LOW);
    serial="";
    return;
  }
  serial="";
  tonoCorrecto();
  // Serial.println("Respuesta servidor:"+serial);
  Serial.print("Memoria libre: ");
  Serial.println(ESP.getFreeHeap());
  client.stop(); // Cierra la conexión después de enviar datos
  http.end(); // Cerrar conexión
}

```

Figura 51*Codificación conexión Wifi*

```

void conexion() { // Conexión WIFI
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED and cont < max_intentos) { //Cuenta hasta 50 si no se puede conectar lo cancela
    cont++;
    delay(500);
    Serial.println("Conectando...");
  }
  Serial.println("\n");
  if (cont < max_intentos) { //Si se conectó
    Serial.println("*****");
    Serial.print("Conectado a la red WiFi: ");
    Serial.println(WiFi.SSID());
    Serial.print("IP: ");
    Serial.println(WiFi.localIP());
    Serial.print("macAddress: ");
    Serial.println(WiFi.macAddress());
    Serial.println("*****");
    digitalWrite(LED1, HIGH);
  }
  else { //No se conectó
    Serial.println("-----");
    Serial.println("Error de conexion");
    Serial.println("-----");
  }
}

void tonoCorrecto() {
  tone(buzzPin, 500); // Primer tono
  delay(400);
  noTone(buzzPin); // Apagar sumbador
}

void tonoIncorrecto() { // Tono para incorrecto (melodía descendente)
  tone(buzzPin, 1000); // Primer tono
  delay(300);
  tone(buzzPin, 800); // Segundo tono más grave
  delay(300);
  tone(buzzPin, 600); // Tercer tono más grave aún
  delay(400);
  noTone(buzzPin); // Apagar sumbador
}

```

3.2.5 Codificación

En esta etapa se realiza la tarea de programación, que consiste esencialmente en llevar el código fuente, todo lo diseñado en la fase anterior.

Figura 52

Controlador de inicio de sesión

```
<?php

namespace App\Http\Controllers\Usuario;

use App\Http\Controllers\Controller;
use App\Http\Requests\Login\UsuarioRequest;
use App\Models\Pago;
use App\Models\Reunion;
use Illuminate\Http\Request;
use App\Models\User;
use Carbon\Carbon;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\Hash;
use Illuminate\Support\Facades\Validator;
use Spatie\Permission\Models\Role;

class Controlador_login extends Controller
{
    private $mensajeError = "Usuario o contraseña inválidos";

    public function ingresar(Request $request)
    {
        if ($this->validarDatos($request)->fails()) {
            return $this->respuestaError('Todos los campos son requeridos');
        }

        $usuario = $this->buscarUsuario($request->usuario);

        if (!$usuario) {
            return $this->respuestaError($this->mensajeError);
        }

        if ($this->autenticarUsuario($request)) {
            return $this->respuestaExitosa('Inicio de sesión con éxito');
        }

        return $this->respuestaError($this->mensajeError);
    }
}
```

Figura 53

Codificación del controlador de usuario

```
<?php
namespace App\Http\Controllers\Usuario;
use App\Http\Controllers\Controller;
use App\Http\Requests\Perfil\UpdatePasswordRequest;
use App\Http\Requests\Usuario\Rol\UpdateRolRequest;
use App\Http\Requests\Usuario\Usuario\UsuarioRequest;
use App\Models\Departamento;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Validator;
use App\Models\registro_lector;
use App\Models\User;
use Database\Seeders\UsuarioSeeder;
use Illuminate\Support\Facades\Hash;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\Log;
use Illuminate\Support\Facades\DB;
use Spatie\Permission\Models\Role;
use Illuminate\Support\Facades\Storage;
use Exception;
use Illuminate\Support\Facades\Redis;
use PhpParser\Node\Stmt\TryCatch;

use function PHPUnit\Framework\isEmpty;

class Controlador_usuario extends Controller
{
    public $mensaje = [];
    public function index()
    {
        $usuarios = User::with('roles')->get();
        $roles = Role::select('id', 'name')->get();
        $departamentos = Departamento::select('id', 'expedido', 'departamento')->get();

        return view('administrador.usuarios.usuarios', [
            'usuarios' => $usuarios,
            'roles' => $roles,
            'departamentos' => $departamentos,
        ]);
    }
    public function show($id_usuario)
    {
        try {
            $user = User::select('id')->find($id_usuario);
            $usuario_actual = auth()->user()->id;
            $registro = registro_lector::select('id', 'cod_targeta')->where('user_id', $usuario_actual)->get();

            if ($registro->isEmpty()) {
                throw new \Exception('No existe targeta escaneada');
            }
        }
    }
}
```

Figura 54

Codificación del controlador de cuotas

```
<?php

namespace App\Http\Controllers\Pagos;

use App\Http\Controllers\Controller;
use App\Models\Mes;
use App\Models\Pago;
use App\Models\User;
use Illuminate\Http\Request;
use Carbon\Carbon;
use Exception;
use Illuminate\Support\Facades\DB;

class Controlador_pagarCuotas extends Controller
{
    public $mensaje = [];

    public function index()
    {
        $meses = Mes::select('id', 'mes')->get();
        return view('administrador.pago.pagarCuotas', [
            'meses' => $meses,
        ]);
    }

    public function create()
    {
        //
    }

    public function store(Request $request)
    {
        try {
            if ($request->meses == null) {
                throw new \Exception("seleccione un mes de pago");
            }
            $user = User::select('id', 'ci')->where('ci', $request->ci_estudiante)->first();

            if (!$user) {
                throw new \Exception(" estudiante no encontrado");
            }

            foreach ($request->meses as $key => $mes) {

                $respuesta = $this->verificarPago($user->id, $mes);

                if ($respuesta != "correcto") {
                    throw new \Exception("ya se registro ese pago");
                }
            }
        } catch (\Exception $e) {
            //
        }
    }
}
```

Figura 55

Codificación de reporte de Asistencia

```

public function reporte_asistencia(Request $request)
{
    $user = User::select('id', 'nombres', 'paterno', 'materno')->where('ci', $request->ci_estudiante)->first();
    if (!$user) {
        // Redirige de regreso con un mensaje de error en la sesión
        return redirect()->back()->withErrors(['ci_estudiante' => 'cedula de identidad no encontrada']);
    }

    $reuniones = $user->reuniones()
        ->whereDate('reuniones.entrada', '>=', $request->fecha_inicio)
        ->whereDate('reuniones.salida', '<=', $request->fecha_final)
        ->get();

    // Reuniones no asistidas por el usuario en el rango de fechas
    $reunionesNoAsistidas = Reunion::whereDate('entrada', '>=', $request->fecha_inicio)
        ->whereDate('salida', '<=', $request->fecha_final)
        ->whereDoesntHave('users', function ($query) use ($user) {
            $query->where('user_id', $user->id);
        })
        ->get();

    $entradaSalidas = DB::table('user_reunion')
        ->where('user_id', $user->id)
        ->select('salida', 'reunion_id', 'entrada') // Selecciona los campos que necesitas
        ->get();

    $pdf = Pdf::loadView('administrador/pdf/reporteControlAsistencia', compact('user', 'reuniones', 'entradaSalidas', 'reunionesNoAsistidas'));

    return $pdf->stream();
}

```

Figura 56

Codificación usuario

```
public function update(UsuarioRequest $request, $id)
{
    DB::beginTransaction();
    try {

        // Encontrar el usuario por ID
        $user = User::findOrFail($request->id_usuario);

        if ($request->estado == "activo") {
            $user->estado = "inactivo";
        }
        if ($request->estado == "inactivo") {
            $user->estado = "activo";
        }

        $user->save();
        DB::commit();

        $this->mensaje("exito", "Estado cambiado Correctamente");

        return response()->json($this->mensaje, 200);
    } catch (Exception $e) {
        // Revertir los cambios si hay algún error
        DB::rollBack();

        $this->mensaje("error", "error" . $e->getMessage());

        return response()->json($this->mensaje, 200);
    }
}
```

Figura 57*Codificación de reporte de asistencia*

```
public function reporte_asistencia($reunion_id)
{
    // Obtener todos los estudiantes
    $estudiantes = User::role('estudiante')->get();
    // $estudiantes=User::all();

    $reunion = Reunion::find($reunion_id);

    // Obtener los asistentes a la reunión
    $asistentes = Reunion::find($reunion_id)->users()->role('estudiante')->get();

    // Filtrar los estudiantes que no asistieron
    $noAsistentes = $estudiantes->diff($asistentes);

    $entradaSalidas = DB::table('user_reunion')
        ->where('reunion_id', $reunion_id)
        ->select('salida', 'user_id', 'entrada') // Selecciona los campos que necesitas
        ->get(); // Cambia first() por get()

    $pdf = Pdf::loadView('administrador/pdf/asistencia', compact('reunion', 'asistentes', 'noAsistentes', 'entradaSalidas'));
    return $pdf->stream();
}
```

CAPÍTULO IV

PRUEBAS

DE EVALUACIÓN Y

RESULTADOS

**INGENIERIA
DE SISTEMAS**
UNIVERSIDAD PUBLICA DE EL ALTO

4. PRUEBAS DE EVALUACIÓN Y RESULTADOS

4.1 MÉTRICAS DE CALIDAD ISO 25001

ISO/IEC 25001 proporciona requisitos y recomendaciones para una organización responsable de implementar y gestionar las actividades de especificación y evaluación de requisitos de calidad de productos de software y sistemas mediante el suministro de tecnología, herramientas, experiencias y habilidades de gestión. Establece un marco de trabajo común para evaluar la calidad del software tiene por objetivo la creación de un marco para definir los requisitos y evaluar la calidad del producto del Software y se desarrolla la evaluación de calidad del presente proyecto para brindar facilidad a todos los usuarios, ahorro económico y seguridad de información.

4.1.1 Funcionabilidad

Es una técnica del software, para realizar el cálculo de la métrica de funcionalidad del software se debe calcular el valor del punto de función que está dada por la siguiente forma:

- **Número de entradas de usuarios**
 - **Número de salidas de usuario**
 - **Número de peticiones de usuario**
 - **Número de archivos**
 - **Número de interfaces externas**
-
- Número de entradas de usuarios, cuenta cada entrada de usuarios que proporciona diferentes datos como ser: mensajes, reportes, alertas, etc.

Tabla 19*Número de entradas de usuarios*

Entradas de usuario	N.º
Administración de usuarios	8
Administración de supervisor/patrocinio	10
Administración de beneficiarios	5
Total	23

Nota. Se determina la cantidad de entradas de usuarios.

- Número de salidas de usuario: Se cuenta cada salida que proporciona el usuario información orientada a la aplicación. El número de salidas que proporciona el sistema como ser mensajes, notificaciones, reportes, alertas, etc.

Tabla 20*Numero de salidas de usuario*

Entradas de usuario	N.º
Administración de usuarios	3
Administración de supervisor/patrocinio	3
Administración de beneficiarios	15
Total	21

Nota. Cantidad de salidas de usuario.

- Número de peticiones de usuario, una petición se define como una estrada interactiva que produce la generación de alguna respuesta del software inmediata en forma de salida interactiva. El número de salidas que proporciona el sistema como ser: notificaciones, reportes, alertas, etc.

Tabla 21

Número de peticiones de usuario

Entradas de usuario	N.º
Administración de usuarios	4
Administración de supervisor/patrocinio	7
Administración de beneficiarios	6
Total	17

Nota. Se determina la cantidad de peticiones

- Numero de archivos, se cuenta y se toman los grupos de archivos maestros lógicos de datos o que son archivos independientes.
- Numero de interfaces externas: la suma de todas las interfaces que son leíbles por la máquina.

Tabla 22*Punto función sin ajustar*

Parámetros de medición	Cuenta
Número de entradas de usuario	23
Número de salidas de usuario	21
Número de peticiones de usuario	17
Número de archivos	10
Número de interfaz Externa	4

Nota. Se determina el punto función sin ajustar.

- Para realizar el cálculo de la cuenta total con factores de ponderación se debe determinar los valores ajustados se hace el empleo de los factores de complejidad listados a continuación.

Tabla 23*Factores de ponderación*

Parámetro de medición	Cuenta	Factor	Total
Número de entradas de Usuario	23	7	161
Número de salidas de Usuario	21	5	105
Número de peticiones de Usuario	17	6	102
Número de Archivos	10	15	150
Número de Interfaz Externa	4	9	36
Cuenta total			554

Nota. Se determina el punto de función sin ajustar.

En la tabla se puede apreciar la cuenta total de los puntos de función si ajustar, para determinar los valores ajustados se hace el empleo de los factores de complejidad listados a continuación:

Tabla 24

Factores de Complejidad

Importancia	0%	20%	40%	60%	80%	100%
	No influye	incidental	moderado	medio	significativo	esencial
Escala						
Factor	0	1	2	3	4	5
¿Requiere el sistema copias de seguridad y de recuperación fiable?						x 5
¿Se requiere comunicación de datos?					X	5
¿Existe funciones de procesos distribuidos?					X	3
¿Es crítico el rendimiento?						x 4
¿El sistema web será ejecutado el SO es actual?						x 5
¿Se requiere una entrada interactiva para el sistema?					x	5
¿Se requiere que el sistema tenga entradas a datos con múltiples ventanas?					x	5
¿Se actualiza los archivos de forma interactiva?						x 5
¿Son complejas las entradas, salidas, los					x	5

archivos o las peticiones?

¿Es complejo el procesamiento interno del sistema?

x 5

¿Se ha diseñado el código para ser reutilizado?

x 4

¿Se ha diseñado el sistema para facilitar al usuario el trabajo y ayudarlos a encontrar la información?

x 5

TOTAL

56

Nota. Se realizó los factores de complejidad eligiendo lo más que se acerque al proyecto.

Para calcular los puntos de función (PF), utilizaremos la relación siguiente

$$PF = \text{Conteo total} [0.65 + 0.01 * \sum fi]$$

Donde:

$$\sum(fi)$$

Sumatoria de los valores de los factores de ajuste, se tiene un resultado:

$$\sum(fi) = 56$$

$$PF = 554 * [0.65 + 0.01 * 56]$$

$$PF = 670.14$$

Considerando que el máximo valor es la sumatoria de los factores de ajuste $\sum(fi) = 54$, se calcula el siguiente valor:

$$PF_{max} = \text{Cuenta Total} [0.65 + (0.01 * \sum fi)]$$

$$PF_{max} = 554 * [0.65 + (0.01 * 70)]$$

$$Pf_{max} = 749.9$$

El cálculo de la funcionalidad se obtiene de la relación entre los máximos valores obtenido del punto de función ajustado:

$$Funcioalidad = \frac{PF}{PFmax} * 100$$

$$Funcioalidad = \frac{670.14}{749.9} * 100$$

$$Funcionalidad = 0.89 * 100$$

$$\mathbf{Funcioalidad} = 89.36\%$$

4.1.2 Confiabilidad

A la confiabilidad del sistema se medirá de acuerdo a la probabilidad de operación libre de fallos de un programa de computadora en un entorno determinado.

$$P(T \leq t) \text{ Posibilidad de fallas.}$$

$$P(T \leq t) = 1 - F(t) \text{ Posibilidad de trabajo sin fallas.}$$

Para calcular la confiabilidad, se toma en cuenta en un periodo de tiempo en el que se ejecuta y se obtiene muestras.

$$F(t) = f * e^{u*t}$$

$$\mu = \frac{\text{Numero de fallas de acceso}}{\text{Número de acceso al sistema}}$$

Donde:

f : funcionalidad del sistema

μ = Probabilidad de error que puede tener el sistema

t = Tiempo de duración de gestión del sistema (días)

Para lo que consideramos un periodo de 20 días como tiempo de prueba se tiene de cada 10 ejecuciones se presenta 1 falta

$$F(t) = f * e^{u*t}$$

$$F(t) = 0.89 \cdot e^{(-\frac{1}{10} \cdot 20)}$$

$$F(t) = 0.1353 * 100$$

$$F(t) = 13.53\%$$

$$\text{Reemplazando: } P(T \leq t) = 0.1353 = 13.53\%$$

$$P(T \leq t) = 1 - 0.1353$$

$$P(T \leq t) = 0.8647 * 100\%$$

$$P(T \leq t) = 86.47\%$$

4.1.3 Usabilidad

La usabilidad o facilidad de uso (FU) se realiza una evaluación para conocer si el sistema satisface los requerimientos establecidos por el usuario, entonces esto medirá de acuerdo a las encuestas planteadas a los usuarios del sistema y los resultados siguientes.

Tabla 25

Escala de calificación

ESCALA	VALOR
Muy bueno	5
Bueno	4
Regular	3
Malo	2
Pésimo	1

Nota. Elaboración de escala de calificación

En la siguiente tabla se calcula la usabilidad utilizando la escala de evaluación de la encuesta que se realizó.

Tabla 26

Evaluación de preguntas para calcula la usabilidad

No	Preguntas	Si	No	Evaluación
1	¿El sistema satisface los requerimientos de manejo de información?	4	1	0,9
2	¿Puede utilizar con facilidad el sistema?	4	1	0,9
3	¿Puede controlar las operaciones que el sistema solicite?	4	1	0,9
4	¿El sistema facilita el trabajo que realiza?	5	0	1
5	¿El sistema cuenta con una interfaz gráfica agradable a la vista?	5	0	1
6	¿Cómo considera el ingreso de datos al sistema?	3	2	0,8
7	¿El sistema permite la retroalimentación de la información?	4	1	0,9
8	¿Las funciones del sistema le parecen complicadas?	3	2	0,8
9	¿Durante el uso del sistema ocurrió errores?	3	2	0,8
TOTAL				8,8

Nota. Elaboración de preguntas para calcular la usabilidad.

Fórmula para determinar a usabilidad del sistema:

$$FU = \left[\frac{\sum xi}{n} * 100 \right]$$

Donde:

- **Xi** = Valores de evaluación de preguntas
- **n** = número de preguntas

Calculando:

$$FU = \frac{8,8}{10} * 100 \%$$

$$FU = 89\%$$

Por lo tanto, se tiene un 89% de usabilidad.

4.1.4 Eficiencia

La eficiencia es el conjunto de atributos que se relacionan con el nivel de performance del software y de la cantidad de recursos usados, bajo las condiciones establecidas en tiempo y recurso. La eficiencia mide si el sistema emplea óptimamente los recursos del sistema.

Tabla 27

Escala de valores de eficiencia

ESCALA	VALOR
Muy bueno	5
Bueno	4
Regular	3
Malo	2
Pésimo	1

Nota. Valores de eficiencia.

Tabla 28*Valoración en la eficiencia del sistema*

No	Preguntas	%
1	¿La respuesta es rápido al utilizar las funciones?	95
2	¿Responde adecuadamente cuando utilizas las funciones?	95
3	¿Puede controlar las operaciones que el sistema solicite?	95
<i>TOTAL</i>		95

Nota. Elaboración en la eficiencia del sistema.

Para la medición de dicho computo de elementos es:

$$E = \frac{\sum xi}{n} * \frac{100}{n}$$

El Sistema se considera eficiente por la óptima utilización los recursos en un **90%**.

4.1.5 Mantenibilidad

El mantenimiento se desarrolló para mejorar el sistema en respuesta a los nuevos requerimientos que la institución tenga y los reglamentos que esta rígid por la misma Por lo cual el índice de madurez del software se determina de acuerdo a la siguiente ecuación.

$$IMS = \frac{[Mt - (Fa + Fb + Fc)]}{Mt}$$

Donde:

Mt: Número de módulos en la versión actual

Fa: Número de módulos en la versión actual que se han añadido

Fb: Número de módulos en la versión actual que se han modificado

Fc: Número de módulos de la versión anterior que se han borrado en la versión.

Variable	Valor
Mt	16
Fa	0
Fb	2
Fc	0

Reemplazamos en la formula:

$$IMS = \frac{[16 - (0 + 2 + 0)]}{16} * 100\%$$

$$IMS = 0.875 * 100\%$$

$$IMS = 87\%$$

Por lo tanto, el sistema tiene un índice de estabilidad de un 87% de facilidad de mantenimiento, y el restante 13% es el margen de error correspondiente a los cambios y las modificaciones.

4.1.6 Portabilidad

La portabilidad define la facilidad de llevar el sistema de un entorno a otro sin mucho esfuerzo y sin ningún problema. El presente proyecto está diseñado en un entorno de acceso vía Web y mide su portabilidad en el lado del Servidor y en el lado del cliente, para este tipo de sistema se toma en cuenta 3 aspectos:

- Acceso a internet.
- El equipo debe contar con un servidor Apache que ejecute PHP.
- Computadora con sistema operativo Windows o Linux, con base de datos MySQL.

Para obtener la potabilidad, se tiene la siguiente formula:

numero de dias par portar el sistema

$$Portabilidad = 1 - \left(\frac{1}{\text{numero de dias para implementar el sistema}} \right)$$

Remplazando a la fórmula

$$Portabilidad = 1 - \left(\frac{1}{8} \right)$$

$$Portabilidad = 0.875 * 100$$

$$Portabilidad = 87.5\%$$

Por lo tanto, el proyecto presenta una portabilidad, ya que los requerimientos para la implementación y ejecución del software son mínimos, por ellos a criterio propio se brinda un 87,5% de portabilidad.

Resultados Final: En la siguiente tabla se muestra el resultado final de los recursos de norma ISO 25000.

Tabla 29

Resultados total de las Métricas de calidad

CARACTERÍSTICA	RESULTADOS
Funcionabilidad	89,36%
Confiabilidad	86,47%
Usabilidad	89%
Eficiencia	90%
Mantenibilidad	87%
Portabilidad	87%
Evaluación Total	88,13%

Nota. Elaboración del resultado total de las métricas de calidad.

4.2 Seguridad del Software ISO 27001

La ISO 27001 es un estándar internacional que define los requisitos para establecer, implementar, mantener y mejorar continuamente un Sistema de Gestión de Seguridad de la Información (SGSI). se considera a la información como un activo valioso para la entidad, el estándar se basado en el Sistema de Gestión de Seguridad de la información (SGSI) que permite evaluar todo tipo de riesgos y amenazas y luego tratarlos sistemáticamente.

4.2.1 Seguridad lógica

- **Copias de Seguridad:** ayuda a protegerla de ser accedida por personas no autorizadas.
- **Encriptación:** La encriptación de la información ayuda a protegerla de ser accedida por personas no autorizadas.
- **Autenticación:** uso de contraseñas encriptadas y sesiones controladas ayuda a proteger la información de accesos no autorizados. Solo las personas autorizadas tienen acceso a la información.
- **Base de datos:** Copias de seguridad regularmente.
- **Código CAPTCHA:** Se usan códigos CAPTCHA para evitar que programas automáticos puedan acceder al sistema.

4.2.2 Seguridad física

- El acceso al servidor Web es restringido y sólo lo puede realizar el administrador o personal autorizado.
- Los dominios son administrados por el personal encargado.
- Se cuenta con contraseñas de acceso al servidor.

4.3 PRUEBAS DE FUNCIONAMIENTO

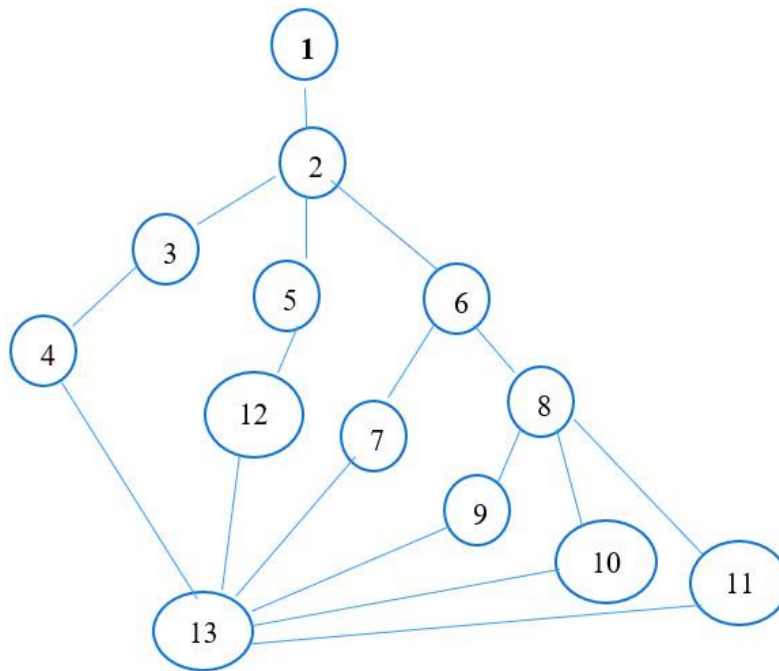
4.3.1 Prueba de caja blanca

Para esta prueba es necesario tener un conocimiento minucioso acerca de la lógica o código a evaluar, pues tomaremos como prioridad el comportamiento interno del programa, aplicando la cobertura de caminos describiremos casos de pruebas suficientes para que se ejecuten todos los caminos de un programa.

- $V(G)=A-N+2$, donde A es el número de aristas y N es el número de nodos
- $V(G)=P+1$, donde P es el número de nodos prediado.

Figura 58

Prueba caja blanca - usuario supervisor/patrocinio



Nota. Elaboración de caja blanca

Donde se tiene como resultado

- Inicio del sistema (1).
- Login usuario y contraseña (2).
- Modulo sistemas (3).
- Administración de usuarios (4).
- Asignación de usuarios (6).
- Cambiar estado (7).
- Módulo de reuniones (8).
- Crea una nueva reunión (9).
- Definir tipo de reunión (10).
- Módulo de cuotas (11).
- Modulo asistencias beneficiario (5).
- Búsqueda de personas (12).
- Fin sistema (13)

La complejidad ciclomática mide el número de cambios independientes dentro del código que es sometido a prueba. La fórmula es:

$$V(G) = A - N + 2 \text{ Donde:}$$

A = Es el número de aristas (Lados)

N = Es el número de nodos (Vértices)

Por tanto:

$$V(G) = 16 - 13 + 2$$

$$V(G) = 6$$

El código tiene una complejidad ciclomática de 6, eso quiere decir que debemos realizar 2 pruebas para asegurarnos de que cada instrucción se ejecute una vez.

4.3.2 Prueba de caja negra

Para las pruebas de caja negra de este proyecto, se deberá evaluar la funcionalidad del sistema sin necesidad de conocer su estructura interna. Aquí hay algunas pruebas que podrían realizarse:

Figura 59

Prueba de caja negra - registro de beneficiario/patrocinado

Campo	Entrada Valida	Entrada invalida
CI	Cadena de texto	Caracteres no alfanuméricos
Nombres	Cadena de texto	Tipo numérico
Apellidos	Cadena de texto	Tipo numérico
Fecha	Cadena de texto	Caracteres no alfanuméricos
hora inicio	Cadena de texto	Caracteres no alfanuméricos
hora final	Cadena de texto	Caracteres no alfanuméricos
Descripción	Cadena de texto	Tipo numérico
Id tarjeta	Tipo numérico	Cadena de texto

Nota. Elaboración prueba de caja negra

Figura 60*Prueba de caja negra - registro de beneficiario/patrocinado*

ENTRADA	C.I.	90548679
	nombres	Mary
	apellidos	Chipana Tarqui
	Fecha	29-10-2024
	hora inicio	09:00
	hora final	12:30
	descripción	Asistió
	id tarjeta	18898266001
SALIDA	Si la información ingresada es correcta el sistema permitirá el registro del pasante para guardar la información en la base de datos.	
	El sistema valida y verifica la información introducida, creando el registro	
RESULTADOS	de un beneficiario/patrocinado y procede a actualizar la tabla del listado.	

Nota. Elaboración caja negra de beneficiario/patrocinado.

La prueba de caja negra evidencia que el formulario cumple con la función programada de registro de pasantes.

4.3.3 Pruebas de estrés

El propósito de las pruebas de estrés es someter al sistema web a una carga extrema hasta alcanzar su límite de capacidad. Para llevar a cabo estas pruebas, se empleó la herramienta Apache JMeter. Los resultados obtenidos se presentan a continuación. Prueba de estrés de la página de inicio de los usuarios lectores.

CAPÍTULO V

ESTIMACIÓN

DE COSTOS

**INGENIERÍA
DE SISTEMAS**
UNIVERSIDAD PÚBLICA DE EL
ALTO **INGENIERÍA**



5. ESTIMACIÓN DE COSTOS

5.1 COCOMO II

Estimación de costos COCOMO II, estimación del sistema se ha desarrollado utilizando KLDC (kilo-líneas de código), como se detalla a continuación. Para calcular el esfuerzo requerido, es necesario determinar la variable KLDC. Este proyecto comprende

Las líneas de código en su totalidad son LCD=9580

Aplicando las conversiones se tiene:

$$KLDC = (LDC)/1000$$

$$KLDC = 9580/1000$$

$$KLDC = 9,58 \text{ KLDC}$$

Tabla 30

Aplicación del Modelo Intermedio

PROYECTO SOFTWARE	A	B	C	D
Orgánico	3,2	1,5	2,5	0,38
Semia coplado	3,0	1,12	2,5	0,35
Empotrado	2,8	1,20	2,5	0,32

A continuacion, se presentaran las ecuaciones necesarias para calcular el costo total del software.

Tabla 31*Ecuaciones de Cocomo II*

VARIABLE	ECUACIÓN	TIPO/UNIDAD
Esfuerzo requerido por el proyecto	$E = a * (KLDC)^b * FAE$	Personas/mes
Tiempo requerido por el proyecto	$T = c * (E)^n$	Meses
Número de personas requeridas para el proyecto.	$NP = E / T$	Personas

Para hallar los valores de FAE, se utilizará la tabla de atributos multiplicadores

Tabla 32

Cálculo de atributos de FAE

ATRIBUTOS	VALOR					
	Muy bajo	Bajo	Nominal	Alto	Muy Alto	Extra Alto
Atributos de Software						
Fiabilidad	0,75	0,88	0,1,00	1,15	1,40	
Tamaño de base de datos			0,1,00	1,08	1,16	
Complejidad		0,70	1,00	1,15	1,30	1,65
Atributos de Hardware						
Restricciones de tiempo de ejecución			1,00	1,11	1,30	1,66
Restricciones de memoria virtual			1,00	1,07	1,21	1,56
Volatilidad de la máquina virtual		0,87	1,00	1,15	1,30	
Tiempo de respuesta		0,87	1,00	1,07	1,15	
Atributos de Personal						
Capacidad de análisis	1,46	1,19	1,00	0,86	0,71	
Experiencia en la aplicación	1,29	1,13	1,00	0,91	0,82	
Calidad de los programadores	1,42	1,17	1,00	0,86	0,70	
Experiencia de la máquina virtual	1,21	1,10	1,00	0,90		

Experiencia en el lenguaje	1,14	1,07	1,00	0,95	
Atributos de Proyecto					
Técnicas actualizaciones de programación	1,24	1,10	1,00	0,91	0,82
Utilización de herramientas de software	1,24	1,10	software	0,91	0,83
Restricciones de tiempo de desarrollo	1,22	1,00		1,04	desarrollo
Total			0,72		

$$FAE = 1.15 * 1 * 1.15 * 1 * 1.07 * 1 * 1 * 0.71 * 0.91 * 0.86 * 1 * 0.95 * 0.91 * 0.91 * 1$$

$$= 0,619$$

Remplazando valores a la fórmula de esfuerzo, se tiene:

$$E = a^{*}(KLDC)^b * FAE$$

$$E = 3,2^{*}(9,58)^{1.05} * 0,61$$

$$E = 20.94 \text{ (Personas/Mes)}$$

Cálculo del tiempo

$$T = c * \text{Esfuerzo}$$

$$T = 2,5 * 21,038$$

$$T = 7,9 \text{ (Meses)}$$

Calculando el personal requerido

$$NP = \frac{E}{T} \text{ personas}$$

$$NP = \frac{20.94}{7.9} \text{ personas}$$

$$NP = 1.65 \cong 2 \text{ personas}$$

Esto indica que se requieren aproximadamente 3 personas para el proyecto.

Cálculo de la productividad

$$PR = LCD/E$$

$$PR = 9580/21$$

$$PR = 456.19 (\text{LCD/Personas Mes})$$

Cálculo del personal requerido

$$P = E/T$$

$$P = 20.94/7.9$$

$$P = 2.65$$

P=2.65(Personas) equivalente a 3 personas

Costo Total del proyecto

(Coste Mes) = P * Salario medio de programadores

Costo Persona Mes 350\$

$$\text{Coste Mes} = 3 \times 350 = 1050 \text{ \$us en Bs.- } 7350$$

$$\text{Costo Total} = 1050 \times 8 = 8400 \text{ \$us en Bs.- } 58800$$

En resumen, los costos que se obtiene estimado que 3 personas a un trabajo de 8 meses resultan un costo total de 8400\$us equivalente en bolivianos 58800 Bs.

Tabla 33*Resultado total de costos*

PARÁMETRO	RESULTADO	UNIDAD
Esfuerzo (E)	20,94	Personas/meses
tiempo (T)	7,9	Meses
número de personas (NP)	3	Personas
Productividad (PR)	456,19	LCD/Personas/Mes
Costo Total (USD)	8400	Dólares
Costo Total (Bs)	58.800	Bolivianos

Nota. El costo total estimado es de 58.800 Bs

CAPÍTULO VI

CONCLUSIONES Y RECOMENDACIONES

**INGENIERIA
DE SISTEMAS**
UNIVERSIDAD PUBLICA DE EL ALTO

6. CONCLUSIONES Y RECOMENDACIONES

6.1 CONCLUSIONES

Se concluyo con los objetivos planteados en el presente proyecto, desarrollando un sistema de información en la Asociación Misión Luz de Vida en el manejo de datos de manera centralizada del beneficiario y patrocinados.

6.1.1 *Respecto al proyecto*

Las conclusiones se realizan de acuerdo con los objetivos específicos que se ha puesto en el presente proyecto.

- Se diseño un módulo de registro de reuniones de control de asistencia aplicando la tecnología de radio frecuencia RFID. **(Ver p. 112)**
- Se desarrollo con éxito la Base de Datos centralizada que cumple con los requisitos de almacenamiento de datos de la institución, permitiendo una gestión eficiente y organizada de la información. **(Ver p. 102)**
- En el sistema se diseñó los módulos de registro de aportes o cuotas, asegurando un mejor manejo sobre las finanzas dentro de la Asociación Misión Luz de Vida. **(Ver p. 113)**
- Se ha implementado un módulo para generar reportes de las actividades realizadas (asistencias de reuniones, cuotas o mensualidades) para los beneficiarios o patrocinados, teniendo un acceso de información inmediata. **(Ver p. 112)**

6.1.2 *Respecto a los objetivos específicos. Respecto al objetivo general*

Se logro implementar satisfactoriamente con el SISTEMA DE INFORMACIÓN WEB PARA LA ADMINISTRACIÓN Y CONTROL DE MEMBRESÍA MEDIANTE LA TECNOLOGÍA RFID, destinado en la Asociación Misión Luz de Vida.

De esta manera se tiene un software que cumple y ayuda a todos los beneficiarios y personal que trabajan o están en la Institución.

Tabla 34

Tabla comparativa antes y después del Sistema

DESCRIPCIÓN	ANTES DEL SISTEMA	DESPUÉS DEL SISTEMA
Proceso de control de asistencia	4 a 5 min.	3 seg.
Manejo de información de los beneficiarios o patrocinados	Se tenía en registro de planillas de Excel lo que generaba perdidas de documentos.	Se almacena en una base de datos toda la información necesaria
Cobro de cuotas o mensualidades	Se generaba documentos o recibos físicos	Módulo de registro de cuotas para las finanzas
Al momento de tener una actividad se necesita la información detallada de cada beneficiario o patrocinado	Demora en la búsqueda de información lo que generaba largas filas	Se genero reportes al acceso inmediato a la información de los beneficiarios

Nota. Elaboración de tabla comparativa del antes y después del Sistema

6.2 RECOMENDACIONES

En base a las observaciones realizadas en la implementación durante la prueba del sistema se elaboraron las siguientes recomendaciones, con el fin de buscar el mejoramiento del sistema.

- Realizar copias de seguridad de la base de datos, con el fin de evitar pérdidas de información.
- Se recomienda cambiar continuamente las contraseñas para la seguridad del sistema y proteger el acceso a personas ajenas.
- Revisar el manual de usuario, si no se tiene claro como realizar una operación del sistema.
- Tener discreción en el manejo de usuarios y contraseñas ya que la información que maneja el sistema es de mucha importancia.
- Se recomienda trabajar con los lenguajes de programación compatibles con el framework Laravel, para tener las versiones actualizadas a futuro.

7. Referencias Bibliográficas

- Altube Vera, R. (2021). *Qué es Laravel: Características y ventajas*, OpenWebinars. Obtenido de <https://openwebinars.net/blog/que-es-laravel-caracteristicas-y-ventajas/>
- Aurazo. (2017). *Modelo conceptual*.
- Cabello, E. (2013). *FASES O ETAPAS DE LA METODOLOGÍA UWE - Diseño del sistema*.
- Cabello, E. (2013). *METODOLOGÍA UWE*.
- Franco Ochoa, J. C. (2010). *"METODOLOGÍA PARA TESTING DE SOFTWARE BASADO EN COMPONENTES, PROYECTO DE GRADO*. Universidad EAFIT. Obtenido de <https://repository.eafit.edu.co/server/api/core/bitstreams/10fedd82-f517-499f-8275-56aaab65fc73/content>
- Garita González, G., & Lizano Madriz, F. (2018). *Estimación de costo de software: Una propuesta de aplicación pedagógica de COCOMO*. Obtenido de <http://dx.doi.org/10.15359/ru.32-1.8>
- Germany. (1968). *Ingenieria de Software*.
- Gómez, M. R. (2013). *CURSO DE DESARROLLO WEB*:. Obtenido de HTML, CSS Y JAVASCRIPT. Obtenido de https://www.cga.es/catalogos/capitulos_promocion/MU00524701_muestra_31754603.pdf
- Guerrero. (2014).
- LÓPEZ PONCE, S. J. (s.f.). *MODELO CLIENTE SERVIDOR*. Obtenido de <https://redespomactividad.weebly.com/>
- MySQL: *Características, Ventajas y Desventajas*. (2024).
doi:<https://www.dongee.com/tutoriales/mysql-caracteristicas-ventajas-y-desventajas/>
- Pressman, R. (2010). *Ingeniería del Software Un Enfoque Práctico*. (7ma ed. ed.). University ofConnecticut. McGraw-Hill Interamericana Editores, S.A.
- Roa, P., Morales, C., & Gutiérrez, P. (2015). *Norma ISO/IEC 25000*.
- Roa, P., Morales, C., & Gutiérrez, P. N. (2015). *Norma ISO/IEC 25000 ISO/IEC 25000 Standard*. Obtenido de <https://revistas.udistrital.edu.co/index.php/tia/article/download/8373/11349/47605>
- Somerville. (2005). *COCOMO II*. Obtenido de https://www.academia.edu/19200942/COCOMO_II
- Sommervielle. (s.f.). *Ingeniería de requerimientos*.
- SQL, C. (2024). *Ventajas y Desventajas de Mysql*. Obtenido de <https://codigosql.top/ventajas-y->

desventajas-de-mysql/

TechTarget. (2024). *MySQL*. Obtenido de

<https://www.computerweekly.com/es/definicion/MySQL#:~:text=MySQL%20es%20un%20sistema%20de,incluyendo%20Linux%2C%20UNIX%20y%20Windows.>

Vallejos S, J., & Sandoval, A. (2016). *Servicios Web*. Managua Nicaragua.

Zelkovitz. (1978). *Ingenieria de Software*.

Chiavenato, I. (2004). *Introduccion a la teoria general de la adminitracion* .

Chiavenato, I. (2006). *Introducción a la Teoría General de la Administración», Séptima Edición. Séptima Edición.*

Chiavenato, I. (2007). *definición de la administración*.

Chiavenato, I. (2019). *Introducción a la teoría general de la administración*.

Clavadetscher, C. (2015). *Autorización en PostgreSQL, Documento de apoyo para cursos*.

Obtenido de https://www.artesano.ch/documents/04-publications/autorizacion_en_postgresql_script_pdfa.pdf

Eslava Muñoz, V. J. (2013). *El nuevo PHP. Conceptos avanzados*.

Estava Muñoz, V. J. (2013). *El nuevo PHP Conceptos avanzados*. España: Bubok Publishing S L.

Franco Ochoa, J. C. (2010). *“METODOLOGÍA PARA TESTING DE SOFTWARE BASADO EN COMPONENTES, PROYECTO DE GRADO*. Universidad EAFIT. Obtenido de <https://repository.eafit.edu.co/server/api/core/bitstreams/10fedd82-f517-499f-8275-56aaab65fc73/content>

Garita González, G., & Lizano Madriz, F. (2018). *Estimación de costo de software: Una propuesta de aplicación pedagógica de COCOMO*. Obtenido de <http://dx.doi.org/10.15359/ru.32-1.8>

Garita, G. (2014). *Métodos analíticos y métricas de calidad del software*. San José. Costa Rica: EUNED. Obtenido de <https://www.redalyc.org/journal/4759/475958170008/html/>

Germany. (1968). *Ingenieria de Software*.

Gigch, V. (1978). *Definición de sistema*.

Gómez, M. R. (2013). *CURSO DE DESARROLLO WEB:.* Obtenido de *HTML, CSS Y JAVASCRIPT*. Obtenido de https://www.cga.es/catalogos/capitulos_promocion/MU00524701_muestra_31754603.pdf

Gonzalez Longatt, F. (s.f.). *Introduccion a los Sistemas de Informacion: Fundamentos*. Obtenido de

- <https://www.uv.mx/personal/artulopez/files/2012/08/FundamentosSistemasInformacion.pdf>
- Guerrero. (2014).
- Guerrero, C., Pech, J., & Dominguez, V. (2014). *UWE en Sistema de Recomendación de Objetos de Aprendizaje. Aplicando Ingenieria Web. Un metodo en caso de estudio*. Obtenido de file:///C:/Users/Mary/Downloads/admin,+Gestor_a+de+la+revista,+relais-v2-n3-137-143%20(1).pdf
- Huidobro, J. M. (s.f.). *La tecnología RFID*. Obtenido de www.acta.es
- Idalia, D. (s.f.). *informatica de cabecera, XAMPP*. Obtenido de <https://dianabaca.wordpress.com/xampp/>
- KeepCoding. (2023). *El porqué usar JavaScript: ventajas y desventajas*. Obtenido de <https://keepcoding.io/blog/porque-usar-javascript/>
- koch. (2000).
- Laguna Quiroga, R. P. (2017). *CONTROL DE RUTAS PARA VEHÍCULOS PÚBLICOS MEDIANTE TECNOLOGÍA RFID [Tesis de Grado, Universidad Mayor de San Andrés]*. Repositorio Institucional. Obtenido de <http://repositorio.umsa.bo/xmlui/handle/123456789/12238>
- Leeman, J. (2013). *Qué es la membresía*.
- Loidon, K. C., & Loudon, J. P. (2012). *definicion Sistema de Informacion*.
- Lopez. (2017). *ISO/IEC 27001*.
- López Espinoza, O. (2014). *Sistema de información y seguimiento de actividades vía web para la iglesia M.N.P.P.D. y del Sistema de radio y televisión S.R.T.P.D. SISA [Proyecto de Grado, Universidad Mayor de San Andrés]*. Repositorio Institucional. Obtenido de <http://repositorio.umsa.bo/xmlui/handle/123456789/8791>
- LÓPEZ PONCE, S. J. (s.f.). *MODELO CLIENTE SERVIDOR*. Obtenido de <https://redespomactividad.weebly.com/>
- Mantilla Guerra, A. (2017). *Gestión de seguridad de la información con la norma ISO 27001:2013*. Recuperado el 29 de diciembre de 2017, de <https://www.revistaespacios.com/a18v39n18/a18v39n18p05.pdf>
- Martínez, A. (2023). *Definición de Sistema*. Obtenido de <https://conceptodefinicion.de/sistema/>.
- Münch Galindo, L. (2018). *Gestión organizacional, enfoques y proceso administrativo*.
- Münch, L. (2010). *Administración. Gestión organizacional y proceso administrativo*.
- Murgesan. (2022). *Ingenieria Web*.
- Pachari Rodriguez, R. D. (2020). *PORTAL WEB PARA LA GESTIÓN DE INFORMACIÓN*

ADMINISTRATIVA Y ACADÉMICA [Proyectos de grado, UNIVERSIDAD PÚBLICA DE EL ALTO]. Repositorio Institucional UPEA. Obtenido de <http://repositorio.upea.bo/jspui/handle/123456789/164>

Peña. (2006).

Pérez Porto, J., & Gardey, A. (2021). *Html - Qué es, definición, estructura e historia.* Obtenido de <https://definicion.de/html/>

Pérez Porto, J., & Gardey, A. (11 de mayo de 2021). *Información - Qué es, importancia, formas y clasificación.* Obtenido de <https://definicion.de/informacion/>

Pérez Porto, J., & Gardey, A. (2021). *Membresía - Qué es, beneficios, definición y concepto.* Obtenido de <https://definicion.de/membresia/>

Portillo Garcia, J. I., Bermejo Nieto, A. B., & Bermardos Barbolla, A. M. (2007). *tecnología de identificación de radiofrecuencia (RFID).*

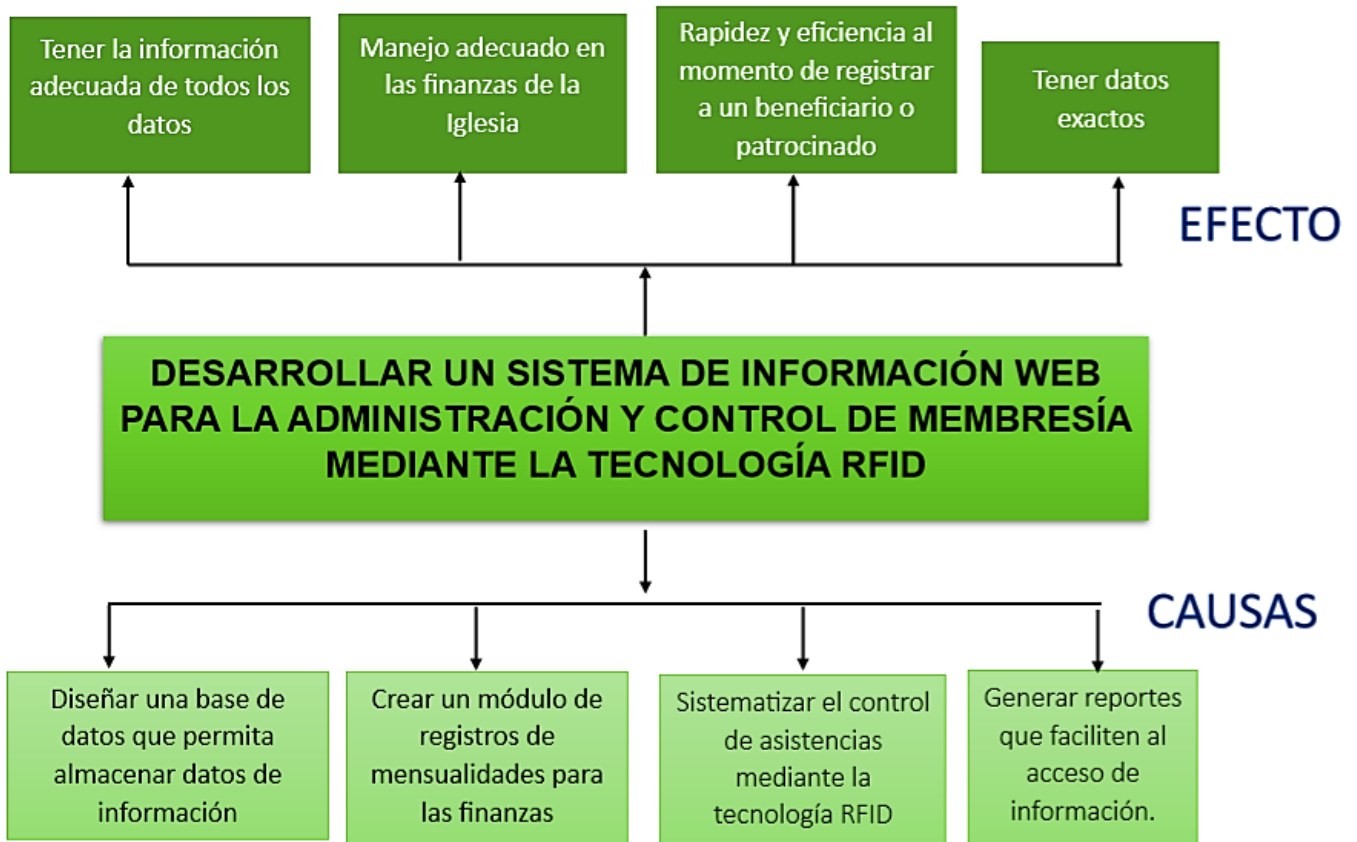
ANEXO

**INGENIERIA
DE SISTEMAS**
UNIVERSIDAD PUBLICA DE EL ALTO

ÁRBOL DE PROBLEMAS



ÁRBOL DE OBJETIVO





MANUAL DE USUARIO



**ASOCIACIÓN MISIÓN
LUZ DE VIDA**

2024

V 1.0



EL SISTEMA DE INFORMACIÓN WEB PARA LA ADMINISTRACIÓN Y CONTROL DE MEMBRESÍA MEDIANTE LA TECNOLOGÍA RFID desarrollado en la **Asociación Misión Luz de Vida**. El sistema fue desarrollado pensando en agilizar una mejor atención.

LOGIN DEL SISTEMA

1. Ingresa el nombre del usuario como se registró en el sistema
2. Ingresar la contraseña con lo que se registró en el sistema
3. ingresar el capcha de seguridad generando en el sistema.

The image shows a login interface for 'Iglesia Luz de Vida'. On the left, there is a banner with a photo of children and the text: 'Iglesia Luz de Vida' and 'Predicar el evangelio de Jesucristo a las personas de la comunidad moldeando vidas y formando líderes comprometidos para la gloria de Dios. (Mateo 9:35)'. Above the banner is a logo featuring a heart with a cross and a lightning bolt. On the right, a pink box contains the login form titled 'Iniciar Session'. The form has three input fields: 'Username', 'Password', and 'Código de Verificación'. The 'Código de Verificación' field contains a CAPTCHA image showing the letters 'r', 'u', 'd', and 't' with a refresh icon. Below the fields is an 'Ingresar' button. Three blue circles with numbers 1, 2, and 3 are placed to the right of the form, with arrows pointing to the Username, Password, and CAPTCHA fields respectively.

Iniciar Session

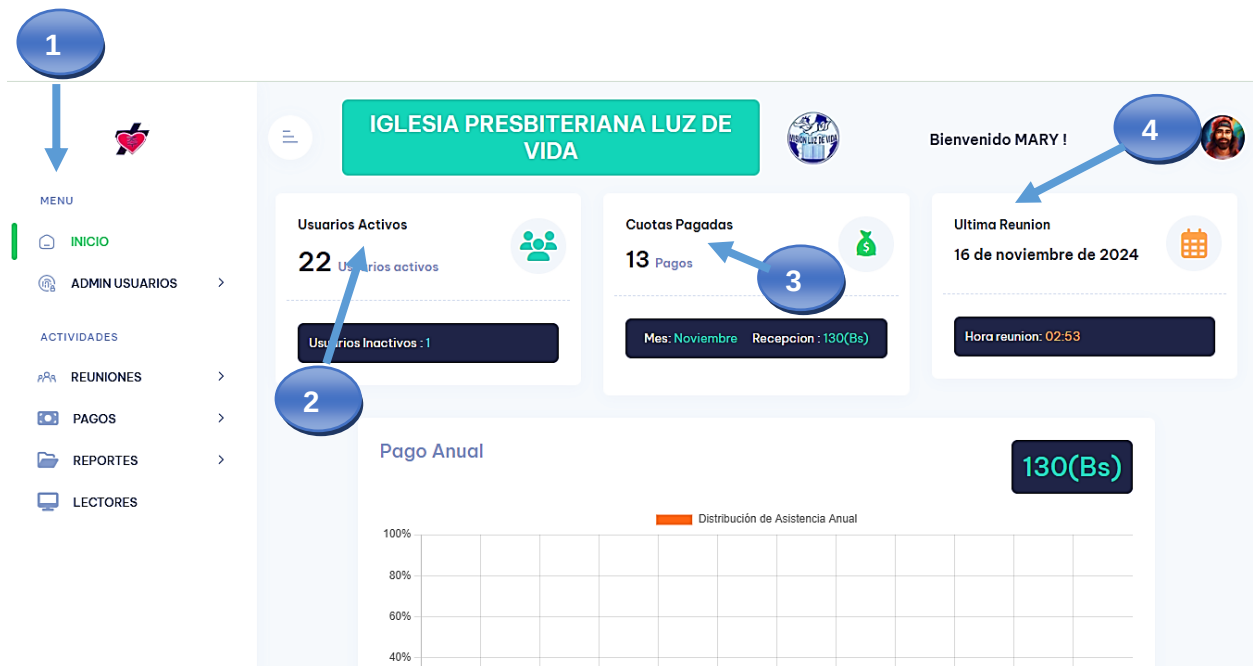
Username

Password

Código de Verificación

INICIO DEL SISTEMA

1. Menú: Se encuentran los administradores y usuarios, también muestran las actividades.
2. Usuarios activos: Muestra la cantidad de los usuarios activos.
3. Cuotas pagadas: Muestra la cantidad de las cuotas pagadas y la suma total del monto que se tiene durante la gestión.
4. Última reunión: Muestra la última reunión realizada durante el mes



Se visualiza la lista de todos los usuarios, también se puede crear nuevos usuarios y designar una tarjeta RFID.

The screenshot shows the 'LISTA DE USUARIOS' (User List) interface. On the left, a sidebar menu highlights 'ADMIN USUARIOS' > 'Usuarios'. The main area displays a table of users with columns: N°, NOMBRE, PATERNO, MATERNO, CI, ROL, ESTADO, COD.TARJETA, and ACCION. Annotations with arrows point to specific actions: 'Crear nuevos' points to a '+ Nuevo' button; 'Clic resetea' points to a reset icon in the 'ACCION' column; 'Edita rol' points to a role icon; and 'Registra nueva' points to a card icon. A search bar is located at the top right of the table area.

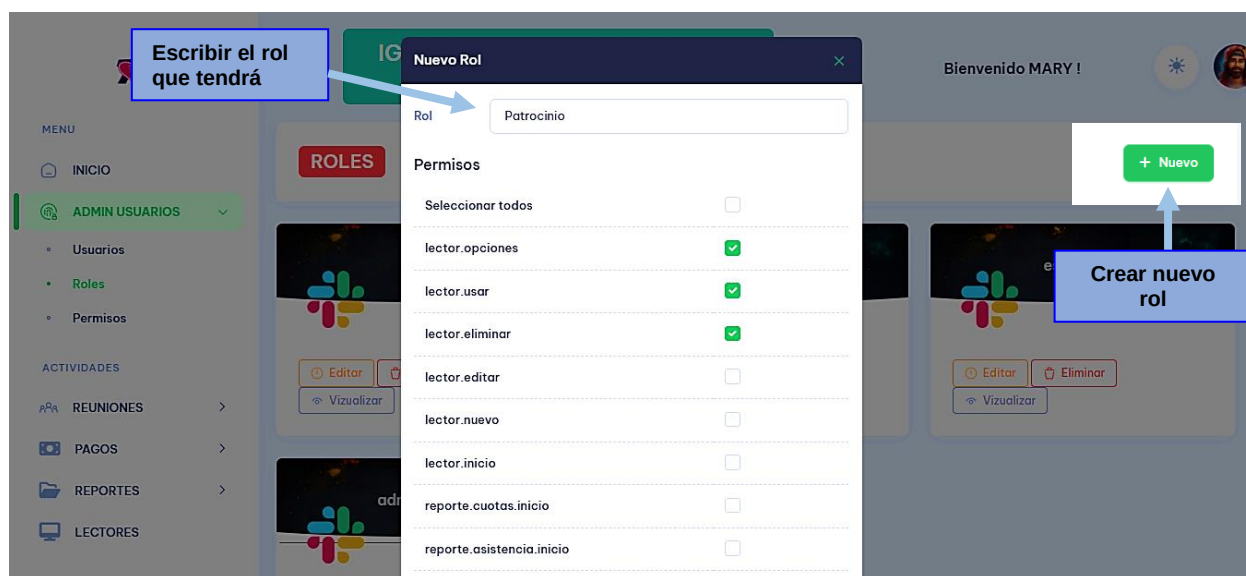
N°	NOMBRE	PATERNO	MATERNO	CI	ROL	ESTADO	COD.TARJETA	ACCION
1	MARY	Chipana	Tarqui	10028685	administrador	☒	Sin asignar	[Reset] [Edit Role] [Add Card]
1	ABIGAIL	Mamani	Apaza	70082848	estudiante	☒	962561607	[Reset] [Edit Role] [Add Card]
1	ASHLY	Mamani	Mamani	54961615	estudiante	☒	447969169	[Reset] [Edit Role] [Add Card]
1	JAIR	Quispe	Merlo	37233615	estudiante	☒	597301994	[Reset] [Edit Role] [Add Card]
1	ESPERANZA	Cadena	Mamani	83689791	estudiante	☒	317582103	[Reset] [Edit Role] [Add Card]
1	LUCIA	Quispe	Choque	36320483	estudiante	☒		[Reset] [Edit Role] [Add Card]

A continuación, se muestra como registrar un nuevo usuario, llenar

The screenshot shows the 'REGISTRAR NUEVO USUARIO' (Register New User) form. The form fields are organized into two columns: DOCUMENTO DE IDENTIDAD (CI/Pasaporte), EXPEDIDO (Expedido), APELLIDO PATERNO (SU APELLIDO PATERNO), ROL (Seleccione una opción), N° TARGETA (TARGETA), and USUARIO (usuario); and COMPLEMENTO, NOMBRE (SU NOMBRE), APELLIDO MATERNO (SU APELLIDO MATERNO), CORREO (juan123@gmail.com), and CONTRASEÑA (contraseña). Annotations with arrows point to the 'Cancelar usuario' button, the 'Guardar los campos' button, and the 'Cerrar' and 'Guardar' buttons at the bottom.

Cancelar usuario → **Cerrar** / **Guardar** (after **Guardar los campos**)

En este campo ingresamos a ADMIN USUARIOS – Roles, Seleccionamos NUEVO y luego se designará el rol que va a tener el nuevo funcionario, elegimos los roles para que pueda acceder al Sistema.



Se muestra la lista de los permisos y se crea nuevos permisos.



Muestra el módulo de Reuniones, se crean nuevas reuniones

IGLESIA PRESBITERIANA LUZ DE VIDA

Bienvenido MARY!

LISTA DE REUNIONES

10 entries per page

N°	TÍTULO	DESCRIPCION	ESTADO	REUNION	ANTICIPO	TOLERANCIA	ACCION
1	reunión	asuntos importantes	terminado	2024-11-16 - 02:53 hasta 03:53	02:38:00	03:23:00	
2	reunión	reunion general	terminado	2024-11-14 - 18:33 hasta 19:03:00	19:03:00	19:03:00	
3	reunión	cancelar mensualidades	terminado	2024-11-14 - 14:26 hasta 15:25	14:11:00	14:56:00	
4	reunión	regalos del mes de noviembre	terminado	2024-11-13 - 21:42 hasta 22:42	21:42:00	22:42:00	
5	reunión	aportes padre	terminado	2024-11-13 - 04:25 hasta 05:25	04:10:00	04:55:00	

Showing 1 to 5 of 5 entries

A continuación, se muestra cómo crear una nueva reunión, anotar el título de la reunión, la descripción o el motivo que se realizara la reunión y para terminar seleccionar la hora de entrada de la reunión y hora de finalización.

REGISTRAR NUEVA REUNION

TITULO REUNION

Ingrese titulo de la reunion

DESCRIPCION

Ingrese descripcion del lector

ENTRADA SALIDA

--:-- --:--

Cerrar Guardar

Una vez ingresado a lista de asistencia, se tiene todas las asistencias de ese día, si por algún motivo un usuario no trajo su tarjeta se puede registrar manualmente. Haciendo clic en nuevo.



En este módulo se encuentra el módulo de pagos – cuotas, también por donaciones



En este módulo se encuentra el módulo de pagos – cuotas, aquí se realizan las cuotas de 10bs que es mensualmente.

IGLESIA PRESBITERIANA LUZ DE VIDA

Bienvenido MARY !

MICHAEL Casas Quina

PAGAR CUOTAS 10 Bs

CI ESTUDIANTE 123456

Colocar C.I.

Clic del mes

Clic pagar

Pagar Cuota

Este es el módulo para generar reportes tanto como de las asistencias de reuniones de la presente gestión, ingresar el C.I. del participante.

IGLESIA PRESBITERIANA LUZ DE VIDA

Bienvenido MARY !

REPORTES DE ASISTENCIA

CI ESTUDIANTE

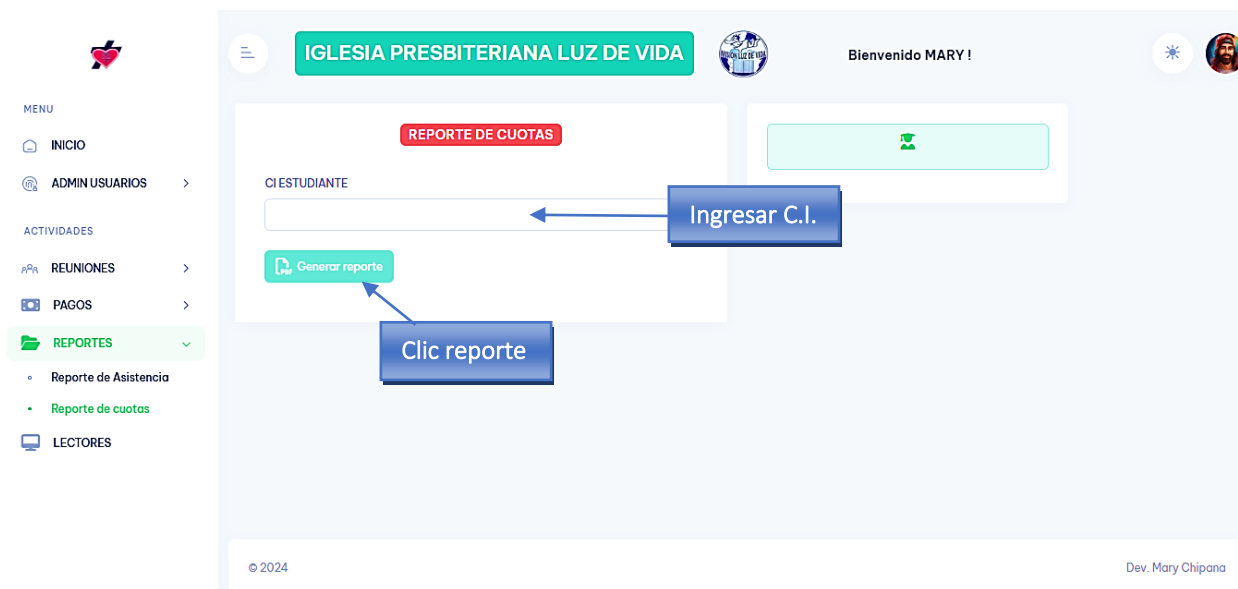
Fecha de inicio dd/mm/aaaa

Fecha de fin dd/mm/aaaa

Generar reporte

Clic reporte

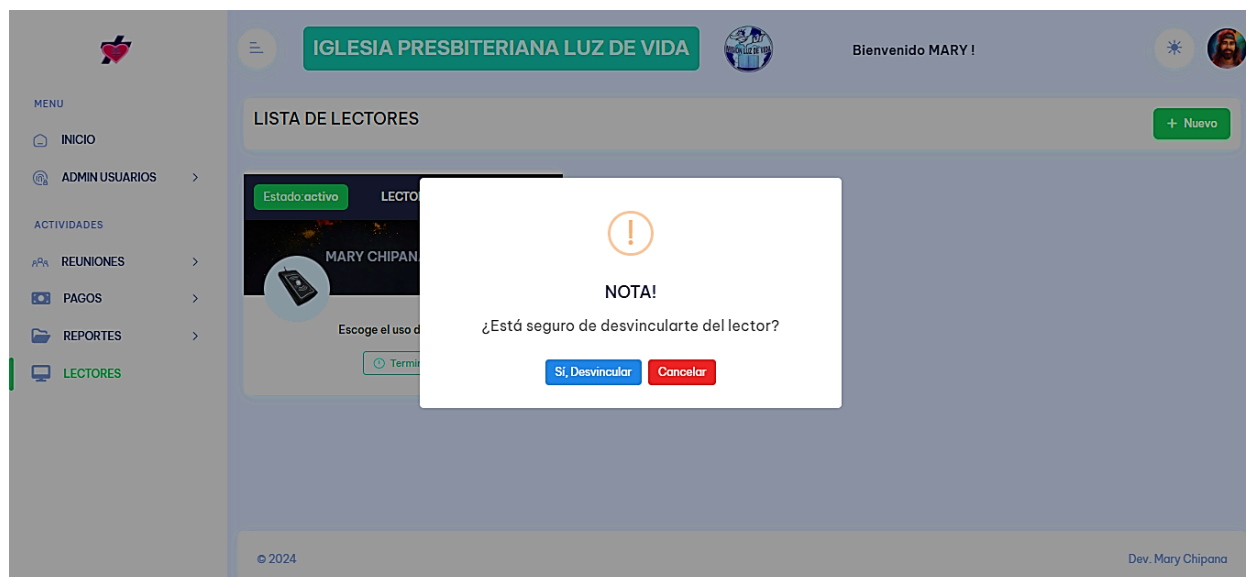
Este es el módulo para generar reportes tanto como de las cuotas o mensualidades de la presente gestión, ingresar el C.I. del participante.



Módulo de lectores para el registro de usuarios nuevo, se da la lectura de las nuevas tarjetas para los nuevos usuarios. También se puede crear nuevos lectores.



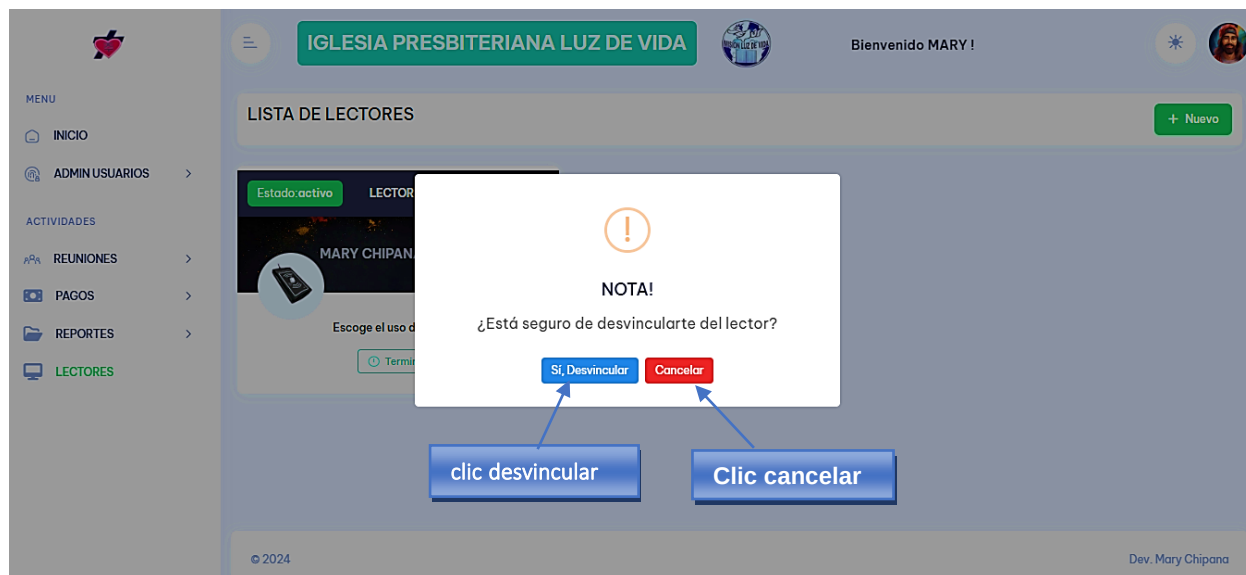
Al momento de terminar la ejecución del registro, se podrá desvincular o terminar con el registro de los nuevos usuarios.



Módulo de lectores para el registro de asistencias, se da la lectura de las tarjetas para las asistencias de los usuarios.



Cuando está en el lector para el control de asistencia, una vez finalizando o para terminar se debe de desvincular.





MANUAL TECNICO

***ASOCIACIÓN MISIÓN
LUZ DE VIDA***



2024

V 1.0



Mary Abigail
Chipana
Tarqui

EL SISTEMA DE INFORMACIÓN WEB PARA LA ADMINISTRACIÓN Y CONTROL DE MEMBRESÍA MEDIANTE LA TECNOLOGÍA RFID.

El sistema fue desarrollado pensando en la población para facilitar los registros mediante módulos de control, podrá tener un mejor seguimiento y estabilidad.

Herramientas.

- PHP versión o superior
- MySql
- Laravel en su versión 11
- Bootstrap
- JavaScript
- Visual Studio Code
- Esp8266

Instalación

- Composer

Configuración,

- GitHub

Seguridad

- Capcha
- Encriptación de URL