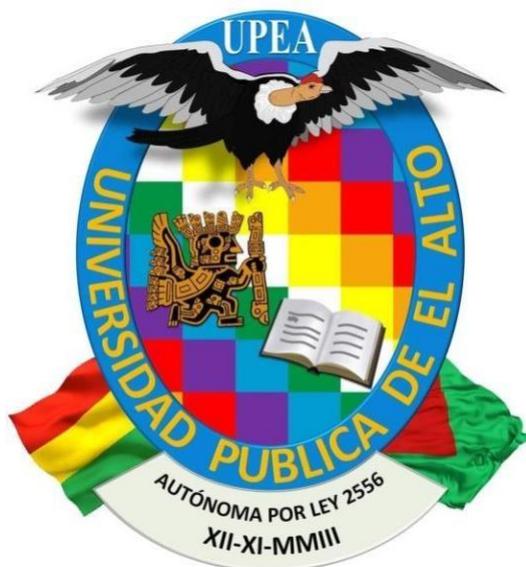


UNIVERSIDAD PÚBLICA DE EL ALTO

CARRERA INGENIERÍA DE SISTEMAS



PROYECTO DE GRADO

**“DISEÑO E IMPLEMENTACION DE UNA BIBLIOTECA
VIRTUAL Y REPOSITORIO DE PROYECTOS DE GRADO Y
TESIS”**

**CASO: ESCUELA SUPERIOR DE FORMACIÓN DE MAESTRAS Y
MAESTROS TECNOLÓGICO HUMANÍSTICO EL ALTO “ESFMTHEA”**

**Para Optar al Título de Licenciatura en Ingeniería de Sistemas
MENCIÓN: Informática y Comunicaciones**

**Postulante: Gladys Orihuela Mamani
Tutor Metodológico: M.Sc.Ing. Enrique Flores Baltazar
Tutor Especialista: Ing. Noemy Mamani Chuyma
Tutor Revisor: Ing. Pascual Yana Chejo**

**EL ALTO – BOLIVIA
2023**

DEDICATORIA

Dedico este trabajo a mi familia que con su amor y comprensión y en momentos difíciles me supieron comprender y gracias a ellos estoy donde estoy.

A mis compañeros de estudio, quienes sin su ayuda nunca hubiera podido culminar mis estudios.

AGRADECIMIENTO

A DIOS quien siempre ha permanecido a mi lado, por darme gracia e infinita sabiduría, brindarme salud, alegrías, tristezas que me tocó vivir y mucha fortaleza para seguir adelante y superar todas las adversidades para alcanzar mis metas y anhelos.

A mi padre José Orihuela Aguilar y mi madre Isabel Mamaní Huanca quienes han sido fundamentales en mi educación, por ser siempre la fortaleza de salir adelante sin importar los obstáculos, por brindarme la confianza y motivación amor y cariño durante el recorrido de toda la carrera. No hay palabras en este mundo para agradecerte, por tanto, muchas gracias papas.

A mis hermanos quienes me han brindado su amor, cariño y la confianza de que lograría culminar la carrera.

Un profundo Agradecimiento al director de carrera M.Sc.Ing. David Carlos Mamaní por siempre ser un apoyo para nosotros los estudiantes y nunca cerrarnos las puertas de la carrera.

Un profundo agradecimiento a mi tutor metodológico M.Sc.Ing. Enrique Flores Baltazar de igual manera agradecerle por su apoyo, la inmensa paciencia que me brindo por apoyarme con su conocimiento, colaboración y correcciones para la conclusión de este proyecto.

Un profundo y especial agradecimiento a mi tutora especialista Ing. Noemy Mamaní Chuyma, por todas sus enseñanzas, por el conocimiento compartido y la paciencia prestada, por toda su colaboración y tiempo brindado para el desarrollo del sistema como para la culminación de este proyecto.

También un agradecimiento a mi tutor revisor Ing. Pascual Yana Chejo por todo su apoyo, por el tiempo dedicado a todas las correcciones brindadas, en el desarrollo de este proyecto.

Gracias a mis amigos Raúl Richard Condori Chuquímia, Noemy Mamaní muchas gracias por todo el apoyo y nunca dejarme sola aun cuando el mundo se caía encima por siempre estar en los momentos buenos y malos.

A "ESFMTHEA" por acogerme en su ambiente de desarrollo al distinguido director y Lic. Mónica Angela Carpio Mendoza por ser un apoyo fundamental, por brindarme el conocimiento necesario para crecer como profesional.

Finalmente, A la carrera de Ingeniería de Sistemas por acogerme durante estos años y brindarme el conocimiento en sus instalaciones y aulas, donde he adquirido los conocimientos necesarios para poder desempeñarme con éxito en el futuro de toda mi vida, no sólo profesional sino también personal.

RESUMEN

Actualmente con el avance de la ciencia y la tecnología contar con un sistema automatizado es una necesidad en cualquier entidad, por tal razón se ha propuesto el Sistema de Diseño e Implementación de una Biblioteca Virtual y Repositorio de Proyectos de Grado y Tesis, para que brinde un mejor servicio al personal de biblioteca docentes y estudiantes como ser: préstamo de libros, descargas de libros digitalizados, reportes y estadísticas.

El avance de las comunicaciones y las nuevas tecnologías, así como el uso del internet, más concretamente la World Wide Web, han abierto una nueva vía de difusión, que induce la necesidad de desarrollar nuevos sistemas virtuales para mostrar un ambiente que simule estar al usuario dentro de una biblioteca.

En la biblioteca de la institución de Escuela Superior de Formación de Maestras y Maestros Tecnológico Humanístico el Alto "ESFMTHEA" se tiene un gran movimiento de documentos, el control de la información es realizado de forma semiautomática, asimismo los usuarios están limitados a los horarios establecidos lo que presenta un problema en el acceso a la información, por otra parte, los materiales bibliográficos se deterioran y otras veces ya no son devueltos, no se tiene un control adecuado. Es así que se plantea desarrollar e implementar un sistema de apoyo a la gestión y digitalización bibliotecaria con el uso de la metodología UWE (*Ingeniería Web basada en UML*), que es extensión a UML en el desarrollo Orientado a Objetos. Cumpliendo los objetivos de dicho sistema al administrar y controlar la información de los materiales bibliográficos de forma segura y oportuna, dando una calidad de servicio y satisfacción a los usuarios.

Viendo las características del proyecto se emplea la tecnología Web, utilizando como lenguaje de programación PHP y como gestor de base de datos MySQL. Una vez finalizado el proyecto, se realizará una evaluación de la calidad de producto de software con ISO/IEC 25000. Y para la estimación de costo de software con el modelo COCOMO II.

Palabras Clave: PHP, WWW, UML, UWE

ABSTRAC

Currently, with the advance of science and technology, having an automated system is a necessity in any entity. For this reason, the Design and Implementation System of a Virtual Library and Repository of Degree Projects and Theses has been proposed to provide a better service to library staff, teachers and students, such as: book loans, downloads of digitized books, reports and statistics.

The advance of communications and new technologies, as well as the use of the Internet, more specifically the World Wide Web, have opened a new way of diffusion, which induces the need to develop new virtual systems to show an environment that simulates the user to be inside a library.

In the library of the Humanistic Technological Teacher Training School the high "ESFMTHEA" there is a great movement of documents, the control of the information is made in a semi-automatic way, also the users are limited to the established schedules what presents a problem in the access to the information, on the other hand, the bibliographic materials deteriorate and other times they are no longer returned, there is not an adequate control. Thus, it is proposed to develop and implement a support system for library management and digitization with the use of the UWE (UML-based Web Engineering) methodology, which is an extension to UML in Object Oriented development. Fulfilling the objectives of this system to manage and control the information of bibliographic materials in a safe and timely manner, giving a quality of service and user satisfaction.

Considering the characteristics of the project, Web technology is used, using PHP as programming language and MySQL as database manager. Once the project is finished, an evaluation of the software product quality will be made with ISO/IEC 25000. And for the software cost estimation with the COCOMO II model.

Keyword: PHP, WWW, UML, UWE

ÍNDICE

1	CAPÍTULO I MARCO PRELIMINAR	1
1.1	INTRODUCCIÓN.....	1
1.2	ANTECEDENTES.....	2
1.2.1	Antecedentes Institucionales.....	2
1.3	PLANTEAMIENTO DEL PROBLEMA.....	6
1.3.1	Problema Principal	6
1.3.2	Problemas Secundarios	6
1.3.3	Formulación del Problema	7
1.4	OBJETIVOS.....	7
1.4.1	Objetivo General.....	7
1.4.2	Objetivos Específicos	7
1.5	JUSTIFICACIÓN.....	8
1.5.1	Justificación Técnica	8
1.5.2	Justificación Económica.....	8
1.5.3	Justificación Social	9
1.6	METODOLOGÍA	9
1.6.1	Metodología UWE	9
1.7	MÉTRICAS DE CALIDAD DE SOFTWARE.....	11
1.8	ESTIMACIÓN DE COSTOS	11
1.8.1	Cómo.....	11
1.9	SEGURIDAD DE SISTEMAS ISO 27000.....	12
1.10	PRUEBAS DE SOFTWARE	12
1.10.1	Pruebas de caja blanca.....	12
1.10.2	Pruebas de caja negra	13
1.10.3	Pruebas de estrés	13
1.11	HERRAMIENTAS	13
1.11.1	Visual Studio Code	13
1.11.2	Lenguaje de programación PHP	14
1.11.3	Lenguaje de Programación JavaScript	14
1.11.4	MySQL.....	14
1.11.5	HTML	14
1.11.6	CSS3.....	15
1.11.7	Frameworks Bootstrap	15
1.11.8	jQuery	16
1.11.9	WampServer.....	16
1.11.10	Patrón MVC.....	16

1.12	LÍMITES Y ALCANCES	17
1.12.1	Limites.....	17
1.12.2	Alcances	17
1.13	APORTES	18
2	CAPÍTULO II MARCO TEÓRICO	21
2.1	INTRODUCCIÓN.....	21
2.2	SISTEMA	21
2.3	BIBLIOTECOLOGÍA	21
2.3.1	Concepto de Biblioteca	21
2.3.2	Tipología de Bibliotecas	22
2.3.3	Características de la Biblioteca.....	24
2.4	IMPLEMENTACIÓN.....	25
2.4.1	Implementación de un sistema informático	25
2.5	DISEÑO	26
2.6	REPOSITORIO	26
2.6.1	Repositorio Digital	27
2.7	METADATOS.....	29
2.7.1	Tipos de Metadatos	29
2.7.2	Clasificación.....	30
2.7.3	Los Metadatos Dublin Core.....	30
2.8	INGENIERIA DE SOFTWARE	35
2.9	INGENIERIA DE SOFTWARE	38
2.9.1	Características De La Ingenieria Web	38
2.10	METODOLOGIA DE INGENIERIA WEB.....	39
2.11	METODOLOGÍA INGENIERÍA WEB BASADA EN UML (UWE).....	39
2.11.1	Características.....	40
2.11.2	Modelos de la metodología UWE.....	41
2.11.3	Faces de la metodología UWE	42
2.12	ARQUITECTURA DE SOFTWARE (MVC)	47
2.12.1	Modelo	47
2.12.2	Vista	48
2.12.3	Controlador	49
2.12.4	Características del Modelo Vista Controlador	49
2.12.5	Ciclo de vida del MVC	51
2.12.6	Ventajas y Desventajas.....	51
2.13	MÉTRICAS DE CALIDAD DE SOFTWARE.....	52
2.13.1	Calidad de Software	52

2.13.2	Importancia de las Métricas	53
2.13.3	Norma ISO/IEC 25000.....	54
2.13.4	Clasificación de las características y sub características.....	58
2.14	ARQUITECTURA DE SOFTWARE.....	64
2.15	MÉTODOS DE PRUEBA DEL SOFTWARE.....	64
2.15.1	Prueba de Caja Blanca	65
2.15.2	Prueba de Caja Negra.....	68
2.16	MÉTODOS DE ESTIMACIÓN DE COSTO DE SOFTWARE.....	69
2.16.1	Modelo COCOMO	69
2.16.2	Método de estimación de Costos COCOMO II.....	70
2.16.3	Estimación de Esfuerzo y tiempo de Desarrollo	74
2.17	SEGURIDAD DE LA INFORMACIÓN.....	76
2.17.1	Estándar ISO/IEC 27000.....	76
2.17.2	Uso de Password	76
2.18	HERRAMIENTAS	77
2.18.1	Gestor de base de datos MySQL.....	77
2.18.2	Lenguaje de Programación PHP.....	80
2.18.3	Bootstrap.....	82
2.18.4	Lenguaje de Programación JavaScript	85
2.18.5	jQuery	87
3	CAPITULO III Marco Aplicativo.....	90
3.1	INTRODUCCIÓN.....	91
3.2	ANÁLISIS DEL SISTEMA ACTUAL	91
3.2.1	Objetivo.....	91
3.2.2	Proceso actual.....	92
3.3	OBTENCIÓN DE REQUISITOS.....	92
3.3.1	Actores implicados en la biblioteca	92
3.3.2	Definición de Actores.....	93
3.3.3	Lista de Requerimientos.....	93
3.3.4	Requerimientos No Funcionales	96
3.3.5	Análisis de Requerimientos.....	96
3.3.6	Caso de Uso General de la Biblioteca	97
3.3.7	Caso de Uso – Ingresar al Sistema	99
3.3.8	Caso de Uso – Administrar Usuario.....	100
3.3.9	Caso de Uso – Administrar datos Libro	102
3.3.10	Caso de Uso – Listar Catalogo	104
3.3.11	Caso de Uso: Gestionar Carnet.....	105

3.3.12	Caso de Uso: Gestionar Préstamo	107
3.3.13	Caso de Uso: Reportes	109
3.3.14	Caso de Uso: Generar Copia de Seguridad	110
3.4	MODELO CONTENIDO.....	111
3.4.1	Diagrama de Clases	112
3.4.2	Modelo de Navegación.....	113
3.4.3	Diagrama de Navegación: Administrar Usuario.....	113
3.4.4	Diagrama de Navegación: Catalogo	114
3.4.5	Diagrama de Navegación: Préstamo	115
3.5	MODELO DE PRESENTACIÓN.....	116
3.6	IMPLEMENTACIÓN DEL SISTEMA	118
3.6.1	Ingreso al Sistema	118
3.6.2	Módulos que integran el sistema	119
4	CAPITULO IV Calidad y Seguridad	126
4.1	INTRODUCCIÓN.....	127
4.2	PRUEBAS DE SOFTWARE	127
4.2.1	Pruebas de Caja Blanca.....	127
4.2.2	Pruebas de Caja Negra.....	130
4.3	NORMA ISO 2500	136
4.3.1	Funcionalidad	136
4.3.2	Confiabilidad	141
4.3.3	Mantenibilidad.....	143
4.3.4	Usabilidad	144
4.3.5	Eficiencia.....	145
4.3.6	Portabilidad.....	147
4.3.7	Resultado.....	148
4.4	SEGURIDAD.....	149
4.4.1	Seguridad de la Base de Datos	149
4.4.2	Seguridad de autenticación	150
4.4.3	Seguridad de la aplicación	151
4.4.4	Seguridad y privacidad de la información.....	151
4.4.5	Seguridad de la Información en la Institución	153
4.4.6	Verificación del Control de Acceso	153
4.4.7	Políticas de privacidad y confidencialidad	153
4.4.8	Políticas de Disponibilidad del servicio y la información	153
4.4.9	Porcentaje de Implementación de Controles.....	153
4.5	SISTEMA DE GESTIÓN DE SEGURIDAD DE LA INFORMACIÓN ISO – 27002..	154

4.5.1	Seguridad Lógica.....	154
4.5.2	Seguridad física	155
4.5.3	Seguridad Organizada.....	155
5	CAPITULO V ANÁLISIS DE COSTOS.....	157
5.1	INTRODUCCIÓN.....	157
5.1.1	Método de Estimación COCOMO II.....	157
6	CAPITULO VI CONCLUSIONES Y RECOMENDACIONES	165
6.1	CONCLUSIONES	165
6.2	RECOMENDACIONES.....	166
	BIBLIOGRAFÍA.....	167
	ANEXO	171

Índice de Gráficos

Gráfico 1.1 Organigrama de la Biblioteca	4
Gráfico 2.1 Fases de Ingeniería de software	38
Gráfico 2.2 Ingeniería Web Basada en UML	42
Gráfico 2.3 Modelo de Caso de Uso de la Ingeniería Web	43
Gráfico 2.4 Modelo de Caso de Uso de la Ingeniería Web	43
Gráfico 2.5 Modelo Contenido de la Ingeniería Web.....	44
Gráfico 2.6 Modelo Navegación de la Ingeniería Web	45
Gráfico 2.7 Modelo de Presentación de la Ingeniería Web	46
Gráfico 2.8 Flujo de MVC	50
Gráfico 2.9 Ciclo de Vida del MVC.....	51
Gráfico 2.10 División de calidad de Software	57
Gráfico 2.11 Calidad del producto de software.....	58
Gráfico 2.12 Filosofía de las Pruebas de Caja Blanca y Caja Negra.....	65
Gráfico 2.13 Condición Múltiple	66
Gráfico 2.14 Grafo de las Estructuras Lógicas de un Programa.....	67
Gráfico 2.15 Grafo de Correspondiente a un Diagrama de Módulos.....	67
Gráfico 2.16 Estructura ISO 27000.....	77
Gráfico 2.17 Herramienta MySql.....	80
Gráfico 2.18 Herramienta php.....	81
Gráfico 2.19 Herramienta Bootstrap	85
Gráfico 2.20 Herramienta JavaScript.....	86
Gráfico 2.21 Herramienta JQuery	89
Gráfico 3.1 Esquema del Sistema.....	91
Gráfico 3.2 Proceso Actual de Préstamos de Libros.....	92
Gráfico 3.3 Diagrama de Casos de Uso General de la Biblioteca	97
Gráfico 3.4 Caso de Uso Ingresar al Sistema	99
Gráfico 3.5 Administrar Usuario	101
Gráfico 3.6 Caso de Uso Administrar datos libro.....	102
Gráfico 3.7 Caso de Uso Listar Catalogo	104
Gráfico 3.8 Caso de Uso Gestionar Carnet	105
Gráfico 3.9 Caso de Uso Gestionar Prestamos.....	107
Gráfico 3.10 Caso de Uso Reportes	109
Gráfico 3.11 Caso de Uso Copia de Seguridad.....	110
Gráfico 3.12 Diagrama de Clase.....	112
Gráfico 3.13 Diagrama de Navegación: Administrar Usuario.....	113
Gráfico 3.14 Diagrama de Navegación: Catalogo	114

Gráfico 3.15 Diagrama de Navegación: Préstamo	115
Gráfico 3.16 Diagrama de Presentación: Inicio de la sesión.....	116
Gráfico 3.17 Diagrama de Presentación: General (Panel Administrativo).....	116
Gráfico 3.18 Diagrama de Presentación: Panel Administrativo	117
Gráfico 3.19 Diagrama de Presentación: Panel Administrativo Carreras	117
Gráfico 3.20 Interfaz Principal del Sistema.....	118
Gráfico 3.21 Pantalla de Inicio Usuario Administrador y Bibliotecario	119
Gráfico 3.22 Lista de Libros Registrados.....	119
Gráfico 3.23 Lista de Libros Registrados.....	120
Gráfico 3.24 Lista de Autores Registrados	120
Gráfico 3.25 Lista de Editorial Registrados	121
Gráfico 3.26 Crear Usuario	121
Gráfico 3.27 Crear Carnet.....	122
Gráfico 3.28 Realizar Reserva	122
Gráfico 3.29 Lista de Prestamos.....	123
Gráfico 3.30 Lista de Reportes Libros	123
Gráfico 3.31 Lista de Reportes Usuarios.....	124
Gráfico 3.32 Gráficos y Estadísticas	124
Gráfico 3.33 Catalogo por Carreras.....	125
Gráfico 3.34 Respaldo Backup	125
Gráfico 4.1 Flujo de probabilidades - prestamos.....	127
Gráfico 4.2 Prueba de Caja Blanca – Inicio de Sesión.....	129
Gráfico 4.3 Pruebas de Caja Negra- Inicio de Sesión.....	131
Gráfico 4.4 Prueba de caja negra – Registro de préstamos	132
Gráfico 4.5 Software Jmeter	134
Gráfico 4.6 Ejecución del programa.....	135
Gráfico 4.7 Resultados de las pruebas.....	135
Gráfico 4.8 Modelo de base de datos entidad relación	150
Gráfico 4.9 Pantalla de usuario y contraseña.....	151

Índice de Tablas

Tabla 2.1 Tipos de metadatos	29
Tabla 2.2 Características y elementos Dublin Core.....	32
Tabla 2.3 Elementos Dublin Core recogido en DCMI Metada Terms.....	33
Tabla 2.4 Elementos refinados	35
Tabla 2.5 Ecuaciones de la Usabilidad.	60
Tabla 2.6 Valoración de las Preguntas.....	60
Tabla 2.7 Ecuaciones de la Mantenibilidad.....	63
Tabla 2.8 detalle de Coeficientes COCOMOII	70
Tabla 2.9 Ecuaciones del Método COCOMO II	71
Tabla 2.10 Atributos FAE.....	72
Tabla 3.1 Lista de Actores	93
Tabla 3.2 Lista de Requerimientos Funcionales	94
Tabla 3.3 Lista de Requerimientos No Funcionales	96
Tabla 3.4 Caso de Uso General	98
Tabla 3.5 Caso de Uso Ingresar al Sistema.....	100
Tabla 3.6 Caso de Uso – Administrar Usuario	101
Tabla 3.7 Caso de Uso Administrar Datos de Libro	103
Tabla 3.8 Caso de Uso Listar Catálogo.....	104
Tabla 3.9 Caso de Uso Gestionar Carnet	106
Tabla 3.10 Caso de Uso Listar Gestionar Préstamo.....	107
Tabla 3.11 Caso de Uso Reportes	109
Tabla 3.12 Caso de Uso Copia de Seguridad.....	111
Tabla 3.13 Evaluación de Flujo	129
Tabla 4.1 Evaluación de Flujo	130
Tabla 4.2 Valores Limites (Inicio de sesión).....	131
Tabla 4.3 Pruebas de caja negra - Inicio de sesión	131
Tabla 4.4 Valores limites (Registro de Préstamos).....	133
Tabla 4.5 Prueba de Caja Negra Registro de Préstamos.....	133
Tabla 3.6 Resultados Prueba de estrés	136
Tabla 4.7 Características de la Funcionalidad	136
Tabla 4.8 Número de Entradas de usuario	137
Tabla 4.9 Número de Salidas Usuario.....	137
Tabla 4.10 Número de Peticiones del Usuario.....	137
Tabla 4.11 Archivos Lógicos Internos	138
Tabla 4.12 Archivos Lógicos Internos	138

Tabla 4.13 Parámetros de Medición.....	138
Tabla 4.14 Factores de Ponderación Punto Función.....	139
Tabla 4.15 Valores de Ajustes de Complejidad	139
Tabla 4.16 Valores para la mantenibilidad	143
Tabla 4.17 Parámetros de medición de usabilidad	144
Tabla 4.18 Preguntas para Determinar la Usabilidad	144
Tabla 4.19 Escala de valores de eficiencia	145
Tabla 4.20 Valoración de la eficiencia del sistema	146
Tabla 4.22 Tabla Hash.....	155
Tabla 5.1 Coeficientes de a,b,c,d de COCOMO II	158
Tabla 5.2 Ecuaciones del Método COCOMO II	158
Tabla 5.3 Coeficientes del Modelo COCOMO II	158
Tabla 5.5 Cálculo de Atributos FAE.....	160
Tabla 5.6 resultado Final	163

**“DISEÑO E IMPLEMENTACIÓN DE UNA BIBLIOTECA
VIRTUAL Y REPOSITORIO DE PROYECTOS DE GRADO
Y TESIS”**

CAPÍTULO I

MARCO

2023

PRELIMINAR



1 CAPÍTULO I MARCO PRELIMINAR

1.1 INTRODUCCIÓN

Hoy en día las tecnologías de la información y la comunicación juegan un papel fundamental, no solo como mediadores instrumentales, sino como nuevos procesos de enseñanza y aprendizaje, generando cambios importantes en la enseñanza académica y la investigación, El desarrollo de la tecnología de la información afecta sin duda a las funciones tradicionales de los centros bibliotecarios, que deben afrontar nuevos retos en la sociedad de la información.

El problema que existe en la Escuela Superior de Formación de Maestras y Maestros Tecnológico Humanístico El Alto, es la administración manual de la biblioteca, teniendo que registrar en actas los préstamos y devoluciones de libros, monografías, proyectos de grado y tesis, haciendo de este un trámite burocrático. La falta de una biblioteca virtual en el campo de estudios de la “ESFMTHEA” afecta al proceso de enseñanza aprendizaje e investigación académica, científica y tecnológica, genera obstáculo en el desarrollo profesional.

Por tal razón se desarrolla el proyecto “Diseño e Implementación de una Biblioteca Virtual y Repositorio de Proyectos de Grado y Tesis” para la Escuela Superior de Formación de Maestras y Maestros Tecnológico Humanístico El Alto, la cual brindara un servicio centralizado de la toda la información bibliotecaria de la institución, y con ello reforzar el proceso de enseñanza y aprendizaje, así como en la investigación acorde a los requerimientos de los docentes de la “ESFMTHEA.

Se plantea usar la metodología UWE (UML-BASED WEB ENGINEERING) ya que cumple y cubre todo el ciclo de vida de las aplicaciones web. Las herramientas a emplear son: PHP, HTML, JavaScript, CSS3, frameworks Bootstrap y con respecto a la base de datos se usará PostgreSQL.

1.2 ANTECEDENTES

1.2.1 Antecedentes Institucionales

La actual Escuela Superior de Formación de Maestros Tecnológico Humanístico El Alto fue creada como INS, Instituto Normal Superior Tecnológico Humanístico El Alto, un 6 de marzo el año 2006, bajo Decreto Supremo N° 28625 durante la presidencia del Sr. Evo Morales Ayma mismo que encomendó el funcionamiento de dicha casa superior de estudios para la ciudad de El Alto como una recompensa por las jornadas de octubre donde la juventud alteña fue sin duda protagonista y como una sentida necesidad de una ciudad pujante, una institución educativa especializada en la formación de profesionales en el ámbito educativo.

El entonces INS Tecnológico Humanístico El Alto, inicia sus actividades académicas a partir de septiembre de 2006 con 6 carreras: Matemática, Lenguaje y Literatura, Física – Química, Artes Plásticas, Idioma Nativo y Educación Musical, en la Unidad Educativa “Kristian Larsen” de Villa Ingenio, con 459 estudiantes distribuidos en 8 paralelos y atendidos por 13 profesionales docentes.

Es así que se concretizó la apertura de una institución nueva en educación superior, buscando una mayor participación de los estamentos sociales y educativos en la gestión académica del INSTHEA, sistematizando de esta manera experiencias de la pedagogía social y productiva.

Posteriormente, mediante el Decreto Supremo N° 0156/09 de fecha 6 de junio de 2009 Los Institutos Normales Superiores se transforman en Escuelas Superiores de Formación de Maestros, por lo que el INSTHEA cambia de denominación a “ESCUELA SUPERIOR DE FORMACIÓN DE MAESTROS TECNOLÓGICO HUMANÍSTICO EL ALTO”.

En la actualidad es recordada cada 6 de marzo Aniversario la ESFMTHEA y tiene como especialidades a: Artes Plásticas y Visuales, Educación Musical, Ciencias Sociales, Educación Primaria Comunitaria Vocacional, Educación Inicial en Familia

Comunitaria, Ciencias Naturales: Biología geografía y Comunicación Lenguaje: Lengua Extranjera Ingles.

Misión

Perfilar a la Biblioteca Adela Zamudio como referencia inmediata en la búsqueda y obtención de información por parte de la comunidad de la Escuela Superior de Formación de Maestros Tecnológico Humanístico El Alto. La Biblioteca Adela Zamudio tiene como visión ser un espacio moderno, agradable y accesible, orientada al usuario (estudiante, docente, administrativo), referente informativo para la gestión y transmisión del conocimiento, donde las nuevas tecnologías estén al alcance de todos. Vinculada e integrada en las metas de calidad y objetivos de la Escuela Superior de Formación de Maestros Tecnológico Humanístico El Alto; con una gestión eficaz, dinámica y eficiente, capaz de provocar con su actividad unos usuarios mejor formados e informados; que contribuya al aprendizaje permanente y adaptada a la innovación educativa.

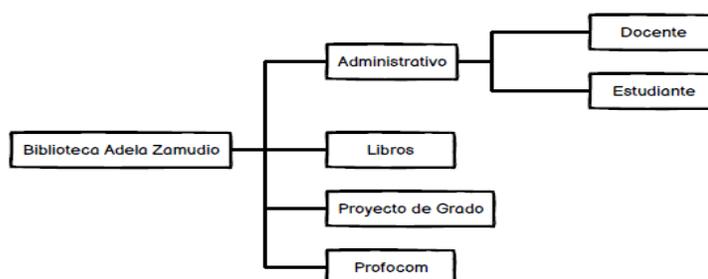
Visión

La Biblioteca Adela Zamudio tiene como misión facilitar el acceso a los recursos de información, en apoyo a estudiantes y docentes investigadores mediante la adquisición, organización y preservación de materiales documentales, conservando y difundiendo el material documental en sus diferentes formatos, brindando los servicios con personal calificado de forma oportuna y eficaz que requiere la comunidad de la Escuela Superior de Formación de Maestros Tecnológico Humanístico El Alto.

Objetivos

- Ofrecer los servicios de información que requiere la comunidad de la Escuela Superior de Formación de Maestros Tecnológico Humanístico El Alto.
- Mantenerse actualizada en las diferentes áreas.
- Salvaguardar el acervo bibliográfico del ESFMTHEA.
- Apoyar los programas de investigación y docencia mediante la adquisición, organización y preservación de materiales documentales.

Gráfico 1.1 Organigrama de la Biblioteca



Nota: Elaboración Propia

Antecedentes Academicos

Internacionales:

Haciendo referencia a trabajos realizados a nivel internacional se puede citar los siguientes:

- (Cavalleri, 2019- Argentina). “Desarrollo de una propuesta de Biblioteca Digital para bibliotecas escolares de educacion primaria de gestion estatal de la ciudad autonoma de buens aires, Argentina”. Esta investigación tiene como objetivo principal proponer el desarrollo de una biblioteca digital, para ser implementada en el contexto de las bibliotecas escolares de la Ciudad Autónoma de Buenos Aires (Argentina), con miras a optimizar la gestión de los recursos electrónicos presentes en las colecciones de estas unidades de información utiliza la metodología de investigación es de tipo proyectiva.
- (Vargas, 2020) Proyecto de una biblioteca Digital para el Centro de Documentación del Servicio Geológico Nacional (SGN) La biblioteca virtual se hace servir del ambiente virtual par su funcionalidad y no tiene necesariamente un lugar físico para su acceso. Por otro lado, Urra(como se citó en Ramírez Céspedes,2006) plantea En una biblioteca virtual es necesario que las fuentes de información estén disponibles de alguna manera y su acceso sea ubicuo, es decir, sin importar donde residan físicamente sus usuarios ni quien se encargó específicamente de su procesamiento y almacenamiento, PhpBB.

Nacionales:

Haciendo referencia a trabajos realizados a nivel nacional se puede citar los siguientes:

- (Aruquipa, 2020) Sistema Web integrado de Administración y control de Bibliotecas, El mundo de la información y la comunicación está viviendo, una época de cambios acelerados que no tiene precedentes, misma que genera multitud de discusiones sobre el futuro que nos espera y que está provocado en su mayor parte, el desarrollo tecnológico. El desarrollo de la tecnología de la información afecta sin duda a las funciones tradicionales de los centros bibliotecarios, que deben afrontar nuevos retos en la sociedad de la información metodología UWE.
- (Gutierrez, 2020) Sistema de Gestión y Digitalización Bibliotecaria, Actualmente con el avance de la ciencia y la tecnología contar con un sistema automatizado es una necesidad en cualquier entidad, por tal razón se ha propuesto el Sistema de Gestión y Digitalización Bibliotecaria, para que brinde un mejor servicio al personal de biblioteca como ser: préstamo de libros, descargas de libros digitalizados, reportes y estadísticas. El avance de las comunicaciones y las nuevas tecnologías, así como el uso del internet, más concretamente la World Wide Web, han abierto una nueva vía de difusión, que induce la necesidad de desarrollar nuevos sistemas virtuales para mostrar un ambiente que simule estar al usuario dentro de una biblioteca. Es así que se plantea desarrollar e implementar un sistema de apoyo a la gestión y digitalización bibliotecaria con el uso de la metodología UWE (Ingeniería Web basada en UML), que es extensión a UML en el desarrollo Orientado a Objetos. Cumpliendo los objetivos de dicho sistema al administrar y controlar la información de los materiales bibliográficos de forma segura y oportuna, dando una calidad de servicio y satisfacción a los usuarios. Viendo las características del proyecto se emplea la tecnología Web, utilizando como lenguaje de programación PHP y como gestor de base de datos MariaDB. Una vez finalizado el proyecto, se realizará una evaluación de la calidad de producto de software con ISO/IEC 9126. Y para la

estimación de costo de software con el modelo COCOMO II.

1.3 PLANTEAMIENTO DEL PROBLEMA

1.3.1 Problema Principal

En la actualidad la Escuela Superior de Formación de Maestros Tecnológico Humanístico El Alto (ESFMTHEA), cuenta con diferentes espacios bibliotecarios donde los recursos se encuentran dispersados de forma física, así mismo carece de la difusión del mismo, por ello se vio la necesidad de una plataforma tecnológica que permita almacenar y difundir de manera digital la producción científica académica (libros, tesis, proyecto de grado, profocom, revistas, etc.) que son utilizados y desarrollados por la comunidad docente estudiantil. La falta de un buscador adecuado demora la atención del servicio y es poco visitado por docentes y estudiantes.

1.3.2 Problemas Secundarios

- La administración y control de las bibliotecas realizadas en hojas de cálculo o manualmente de los libros, monografía, proyectos de grado y tesis generado con esto pérdidas de información.
- Ejemplares limitados y deterioro de los mismos, por lo que en muchas ocasiones solo un usuario puede acceder a la vez, o bien llega a ser extraviado a falta de respaldos.
- La ausencia de difusión de los documentos para el conocimiento del estamento docente, estudiantil.
- La búsqueda de información de registro bibliográfico en libros y otros, para su respectiva reservación de préstamo o consulta de disponibilidad, son de forma presencial por parte de la población estudiantil, afectando pérdidas en tiempo considerable.
- No existe una plataforma tecnológica de un sistema que respalde y registre los documentos intelectuales de forma digital.

1.3.3 Formulación del Problema

Analizando los problemas planteamos el siguiente interrogante.

¿De qué manera el diseño de una biblioteca virtual moderna en la Escuela Superior de Formación de Maestros Tecnológico Humanístico El Alto, y los repositorios digitales existentes en el sistema de conexión de unidades de información de la institución “ESFMTHEA” Podrían beneficiar a la población docente - estudiantil en el proceso de enseñanza aprendizaje y en la investigación?

1.4 OBJETIVOS

1.4.1 Objetivo General

Desarrollar una biblioteca virtual moderna en la Escuela Superior de Formación de Maestros Tecnológico Humanístico El Alto, para proporcionar beneficios en el acceso a la información e investigación a docentes y estudiantes, a través de los repositorios digitales existentes en el Sistema de Conexión de Unidades de Información de la Institución “ESFMTHEA”.

1.4.2 Objetivos Específicos

- Analizar los requerimientos mínimos y herramientas necesarias tanto en hardware como en el software para albergar e implementar el sistema bibliotecario repositorio institucional.
- Realizar el diseño correspondiente del sistema, desarrollando los diferentes módulos que poseerá para la administración de la información centralizada.
- Construir la plataforma web de la biblioteca y repositorio para incrementar la difusión e interacción con los usuarios.
- Implementar una biblioteca virtual, que incluya como recurso tecnológico a nuevas tecnologías de la información y comunicación, por alta demanda de los usuarios.
- Sistematizar el registro y respaldo de documentos de investigación facilitando

el alcance de los mismos a los investigadores.

- Suministro del material bibliográfico con el uso correcto de normas de catalogación (Dewey)
- Generar Reportes de libros, estudiantes.

1.5 JUSTIFICACIÓN

1.5.1 Justificación Técnica

La siguiente propuesta, busca solucionar un problema existente en la Escuela Superior de Formación de Maestras y Maestros Tecnológico Humanístico El Alto, porque al diseñar la biblioteca virtual, será un aporte a la comunidad estudiantil y de esta manera estaremos promoviendo el desarrollo de la investigación e innovación tecnológica en el proceso de enseñanza aprendizaje a través de la información actualizada, bibliografías apropiadas acceso de base de datos en línea por internet. Por tal motivo, es necesario el servicio de la plataforma virtual en la localización de la información solicitada por el usuario.

1.5.2 Justificación Económica

Hoy en día dentro de muchas instituciones que generan y almacenan grandes volúmenes de información, están buscando algún tipo de sistema que coadyuve en las tareas de almacenamiento, preservación y difusión de sus recursos minimizando sus costos, tales como la Escuela Superior de Formación de Maestros Tecnológico Humanístico El Alto, donde se administran una gran cantidad de recursos, libros, tesis, proyectos de grado. Es en ese sentido que el presente proyecto permite ahorrar costos centralizando los documentos de forma digital, ayudando a la comunidad docente estudiantil además en el ahorro de tiempo. La biblioteca y repositorio institucional en plataforma web se desarrolló utilizando software libre, es decir, no tiene costo alguno para la Escuela Superior de Formación de Maestros Tecnológico Humanístico El Alto.

1.5.3 Justificación Social

En las últimas gestiones la cantidad de libros, tesis, proyectos de grado, revistas, entre otros, han tenido un crecimiento considerable, pero con la implementación de la biblioteca y repositorio institucional en la plataforma web, se obtiene grandes beneficios para toda la comunidad docente estudiantil y sus autoridades, en lo que contempla una mejor manera de gestión de sus recursos intelectuales, libros, tesis, proyectos de grado, revistas, además de la difusión de los mismos. En cuanto a los estudiantes, se pretende beneficiar en el acceso inmediato desde medios tecnológicos para las consultas informativas y publicaciones de sus trabajos de grado, también beneficiara a los docentes, ya que les permitirá el almacenamiento de sus proyectos (profocom), la consulta de libros y documentos de investigación como referencia, finalmente beneficiara al investigador en general ya que se podrá acceder y consultar a los recursos intelectuales de una manera más sencilla y rápida desde un dispositivo con acceso a internet.

1.6 METODOLOGÍA

Desde hace unos años, la World Wide Web se ha convertido en una plataforma para la ejecución de toda clase de aplicaciones que cumplen un sinnúmero de funciones. La web ha evolucionado incorporando elementos de seguridad, optimización, concurrencia y demás requerimientos que son necesarios para crear soluciones sólidas. (Pasaje, 2005)

1.6.1 Metodología UWE

La metodología para el desarrollo del software se hará uso de la metodología UWE (UML-Based Web Engineering, en español (Ingeniería Web Basada en UML) es una metodología que permite modelar de mejor manera una aplicación Web, para el proceso de creación de aplicaciones detalla ésta, con una gran cantidad de definiciones, en el proceso de diseño lista que debe utilizarse. Procede de manera iterativa e incremental, coincidiendo con UML, incluyendo flujos de trabajo y puntos de

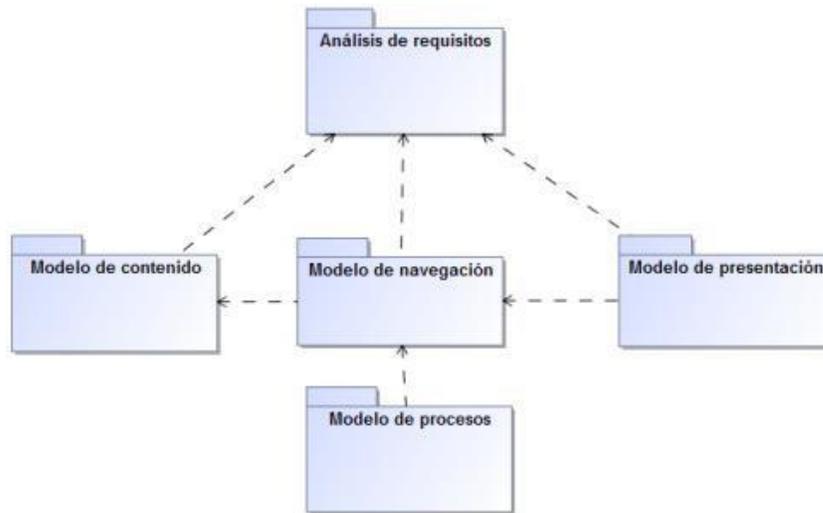
control. (Pasaje, 2005)

UWE es una metodología que permite especificar de mejor manera una aplicación Web en su proceso de creación, mantiene una notación estándar basada en el uso de UML para sus modelos y sus métodos, lo que facilita la transición (Pasaje, 2005)

Modelo de aplicación web según la metodología UWE, se aplicará con las siguientes fases:

- **Modelo de Análisis de requerimiento:** El primer paso para el desarrollo de un sistema web que se especificará con UWE, es realizar la identificación de los requerimientos y plasmarlos en un modelo de requerimientos.
- **Modelo 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.
- **Modelo Contenido:** El modelo conceptual de un sistema software constituye una abstracción externa que describe mediante diagramas y notaciones con distinto grado de formalidad el conocimiento que debe poseer una persona acerca de un sistema, conocimiento que se encuentra almacenado en la Memoria a Largo Plazo.
- **Modelo Navegación:** Especifica el entorno en la cual se realizará el aspecto de navegación de la aplicación Web.
- **Modelo de Presentación:** Representa las vistas del interfaz del usuario mediante modelos estándares de interacción UML.

Grafico 1. Modelos de UWE



Nota: (Pasaje, 2005)

1.7 MÉTRICAS DE CALIDAD DE SOFTWARE

ISO/IEC 25000 establece criterios para la especificación de **requisitos de calidad de productos software**, sus métricas y su evaluación. También incluye un modelo de calidad para unificar las definiciones de calidad de los clientes con los atributos en el proceso de desarrollo. (Rodríguez, 2012)

1.8 ESTIMACIÓN DE COSTOS

La estimación de costos es la práctica de pronosticar el costo de completar un proyecto con un alcance definido. Es el elemento principal de la gestión de costos del proyecto, un área de conocimiento que implica la planificación, el seguimiento y el control de los costos monetarios de un proyecto. (Eby, 2017)

1.8.1 Cócomo

COCOMO (Cost Constructive Model) es el Modelo Constructivo de Costes. Un modelo de dominio público, que viene siendo utilizado y evaluado ampliamente. Es un modelo matemático de base empírica que permite la estimación del coste y la duración de los Proyectos de Software: esfuerzo y tiempo. Es empírico debido a que se basa en

ecuaciones no lineales obtenidas mediante técnicas de regresión a través de un histórico de proyectos ya realizados. (Gómez, 2017)

1.9 SEGURIDAD DE SISTEMAS ISO 27000

ISO 27001 es una norma internacional que permite el aseguramiento, la confidencialidad e integridad de los datos y de la información, así como de los sistemas que la procesan.

El estándar **ISO 27001:2022** para los **Sistemas Gestión de la Seguridad de la Información** permite a las organizaciones la evaluación del riesgo y la aplicación de los controles necesarios para mitigarlos o eliminarlos. (Mendez, 2015)

1.10 PRUEBAS DE SOFTWARE

Las pruebas de software son un conjunto de procesos con los que se pretende probar un sistema o aplicación en diferentes momentos para comprobar su correcto funcionamiento. Este tipo de pruebas abarca cualquier estadio del desarrollo del sistema, desde su creación hasta su puesta en producción. Lo interesante de las pruebas es que se puedan ejecutar de manera automática, para determinar en cualquier momento si tenemos una aplicación estable o si, por el contrario, un cambio en una parte ha afectado a otras partes sin que nos demos cuenta. (MAGALLAN, 2016)

1.10.1 Pruebas de caja blanca

La prueba de caja blanca, en ocasiones llamada prueba de caja de vidrio, es una filosofía de diseño de casos de prueba que usa la estructura de control descrita como parte del diseño a nivel de componentes para derivar casos de prueba. Al usar los métodos de prueba de caja blanca, puede derivar casos de prueba que: 1) garanticen que todas las rutas independientes dentro de un módulo se revisaron al menos una vez, 2) revisen todas las decisiones lógicas en sus lados verdadero y falso, 3) ejecuten todos los bucles en sus fronteras y dentro de sus fronteras operativas y 4) revisen

estructuras de datos internas para garantizar su validez. (MAGALLAN, 2016)

1.10.2 Pruebas de caja negra

Las pruebas de caja negra, también llamadas pruebas de comportamiento, se enfocan en los requerimientos funcionales del software; es decir, las técnicas de prueba de caja negra le permiten derivar conjuntos de condiciones de entrada que revisarán por completo todos los requerimientos funcionales para un programa. (MAGALLAN, 2016)

1.10.3 Pruebas de estrés

Esta prueba se utiliza normalmente para romper la aplicación. Se va doblando el número de usuarios que se agregan a la aplicación y se ejecuta una prueba de carga hasta que se rompe. Este tipo de prueba se realiza para determinar la solidez de la aplicación en los momentos de carga extrema y ayuda a los administradores para determinar si la aplicación rendirá lo suficiente en caso de que la carga real supere a la carga esperada. (MAGALLAN, 2016)

1.11 HERRAMIENTAS

El avance científico y tecnológico impusieron varias maneras de disminuir los riesgos y eliminar los problemas de desarrollo a través de los años, tratando de que se puedan producir sistemas más rápidamente y más complejos que puedan mantenerse.

1.11.1 Visual Studio Code

Visual Studio Code es un editor de código fuente desarrollado por Microsoft para Windows, Linux y macOS. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código. También es personalizable, por lo que los usuarios pueden cambiar el tema del editor, los atajos de teclado y las preferencias. (Luca, 2015)

Es gratuito y de código abierto, aunque la descarga oficial está bajo software propietario. Se le pueden añadir extensiones que hacen más simple la legibilidad del

código, como así la escritura, por ejemplo, el cierre automático de tags, y detección automática de etiquetas, disminuyendo las posibilidades de error de escritura. (Luca, 2015)

1.11.2 Lenguaje de programación PHP

PHP es un lenguaje de programación de propósito general que se ejecuta del lado del servidor, permite la creación de páginas web dinámicas que pueden estar dentro de páginas en HTML. Es uno de los lenguajes de programación web más populares por su rapidez y la facilidad de desarrollo (Souza, 2020).

1.11.3 Lenguaje de Programación JavaScript

JavaScript es uno de los lenguajes de programación más populares para el desarrollo web, ejecutado del lado del cliente (esta expresión significa que los script o programas son ejecutados en el navegador del usuario), y funciona como una extensión del código HTML (Menéndez, 2019)

1.11.4 MySQL

MySQL es un sistema gestor de base de datos (SGBD, DBMS por sus siglas en ingles) muy conocido y ampliamente usado por su simplicidad y notables rendimiento. Aunque carece de algunas características avanzadas disponibles en otros SGBD del mercado, es una opción atractiva tanto para aplicaciones comerciales, como de entretenimiento precisamente por su facilidad de uso y tiempo reducido de puesta en marcha. Esto y su libre distribución en Internet bajo licencia GPL le otorga como beneficios adicionales (no menos importantes) contar con un alto grado de estabilidad y un rápido desarrollo. (Ortega, 2009)

1.11.5 HTML

Este lenguaje de marcación, diseñado para crear páginas web y otros documentos que sea posibles visualizar en browser.

HTML5 provee básicamente tres características: estructura, estilo y funcionalidad. Nunca fue declarado oficialmente, pero, incluso cuando algunas APIs (Interface de Programación de Aplicaciones) y la especificación de CSS3 por completo no son parte del mismo, HTML5 es considerado el producto de la combinación de HTML, CSS y Javascript. Estas tecnologías son altamente dependientes y actúan como una sola unidad organizada bajo la especificación de HTML5. HTML está a cargo de la estructura, CSS presenta esa estructura y su contenido en la pantalla y Javascript hace el resto que es extremadamente significativo. (MUÑOZ, 2021)

1.11.6 CSS3

Son hojas de Estilo en Cascada – que es un lenguaje que descubre la presentación de los documentos estructurados en hojas de estilo para mostrar un documento en pantalla.

CSS es un lenguaje que trabaja junto con HTML para proveer estilos visuales a los elementos del documento, como tamaño, color, fondo, bordes, etc. En este momento las nuevas incorporaciones de CSS3 están siendo implementadas en las últimas versiones de los navegadores más populares, pero algunas de ellas se encuentran aún en estado experimental. Por esta razón, estos nuevos estilos deberán ser precedidos por prefijos tales como –moz- o -webkit para ser efectivamente interpretados. Analizaremos este importante asunto más adelante. (Martines, 2022)

1.11.7 Frameworks Bootstrap

Es un framework, herramientas de software libre para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS.

El framework Bootstrap vio la luz en el año 2011. En un principio fue desarrollado por Twitter aunque posteriormente fue liberado bajo licencia MIT. Hoy en día continúa su desarrollo en un repositorio de GitHub. Se trata de un framework que ofrece la posibilidad de crear un sitio web totalmente responsive mediante el uso de librerías

CSS. En estas librerías, nos podemos encontrar un gran número elementos ya desarrollados y listos para ser utilizados como pueden ser botones, menús, cuadros e incluso un amplio listado de tipografías. Desde que vio la luz, Bootstrap se ha caracterizado por tratarse de una excelente herramienta para crear interfaces de usuarios limpias y totalmente adaptables a cualquier tipo de dispositivo y pantalla, independientemente de su tamaño. (A, 2023)

1.11.8 jQuery

Es una biblioteca de JavaScript rápida y concisa que simplifica el documento HTML, manejo de eventos, animación y las interacciones AJAX para el desarrollo web. jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio. (Quiroga, 2019)

1.11.9 WampServer

WAMP es el acrónimo usado para describir un sistema de infraestructura de internet que usa las siguientes herramientas: * Windows, como sistema operativo; * Apache, como servidor web; * MySQL, como gestor de bases de datos; * PHP (generalmente), Perl, o Python, como lenguajes de programación.

WampServer de Windows es un entorno de desarrollo Web. Le permite crear aplicaciones web con Apache, PHP y la base de datos MySQL. También viene con SQLiteManager PHPMyAdmin para que administre sus más fácilmente sus bases de datos. (martinez, 2013)

1.11.10 Patrón MVC

MVC es un patrón de diseño que considera dividir una aplicación en tres módulos claramente identificables y con funcionalidad bien definida: El Modelo, las Vistas y el Controlador.

Durante toda la década del setenta, SmallTalk y algunos otros lenguajes como Simula I, fueron construyendo gradualmente el paradigma de programación orientada a objetos y estableciendo conceptos tales como objetos, clases, encapsulación, herencia y polimorfismo [3]. Si bien dichos lenguajes no son usados actualmente para implementar aplicaciones comerciales, los conceptos que dejaron en el mundo del desarrollo de software están vigentes en la actualidad y son la base de lenguajes modernos como C++, Java o C#. (Pantoja, 2004)

1.12 LÍMITES Y ALCANCES

1.12.1 Límites

- El proyecto se centra especialmente en aquellos recursos intelectuales (libros, revistas) e proyectos y tesis de grado como también profocom producidos en la Escuela Superior de Formación de Maestras y Maestros Tecnológico Humanístico El Alto.
- El Sistema de biblioteca repositoria no determina la calidad de un recurso intelectual o de investigación.
- El sistema no realizará ninguna modificación al contenido de los archivos ya que se encargará de la preservación, resguardo digital y su difusión como aporte de la institución “ESFMTHEA” ante la comunidad estudiantil, docente, administrativa.

1.12.2 Alcances

En el presente proyecto de grado sistema bibliotecario repositorio “ESFMTHEA” se puede determinar los siguientes alcances:

- El proyecto tiene una difusión institucional, ya que se puede consultar los documentos intelectuales de la biblioteca repositoria.
- El sistema bibliotecario repositorio tiene un alcance sobre la comunidad institucional por la difusión de su producción intelectual compartiendo con los docentes y administrativos de la Escuela Superior de Formación de Maestras y

Maestros Tecnológico Humanístico El Alto.

Con la implementación del presente proyecto, se pretende alcanzar los siguientes módulos:

- Módulo de Ingreso: El sistema permite autenticar a un usuario registrado a través de su inicial de su primer nombre y su carnet y con su clave que es su fecha de nacimiento.
- Módulo de Administración de usuarios: el repositorio tiene la administración de los usuarios según el rol y privilegios.
- Módulo de gestión de comunidades y colecciones: el sistema permite la gestión de las comunidades, colecciones donde se alojan los documentos publicados.
- Módulo de registros: el sistema permite el registro de nuevos documentos intelectuales (libros, revistas) proyectos y tesis de grado y los respectivos metadatos del documento.

1.13 APORTES

Aporte Institucional:

- Permite a la institución contar con una mejor gestión de sus documentos intelectuales.
- Brinda una mejor manera de preservación y almacenamiento de documentos intelectuales en formato digital y centralizado.

Aporte Social:

- Los libros y proyectos de grado son difundidos mediante el sistema de biblioteca repositoria lo que permite consultar y visualizar los documentos intelectuales existentes en el sistema.

Aporte Académico

- Los estudiantes, docentes y administrativos obtienen mayor visibilidad y

difusión de sus documentos intelectuales.

- Facilita la comunicación e intercambio de información entre los miembros de la comunidad institucional.

Aporte Tecnológico

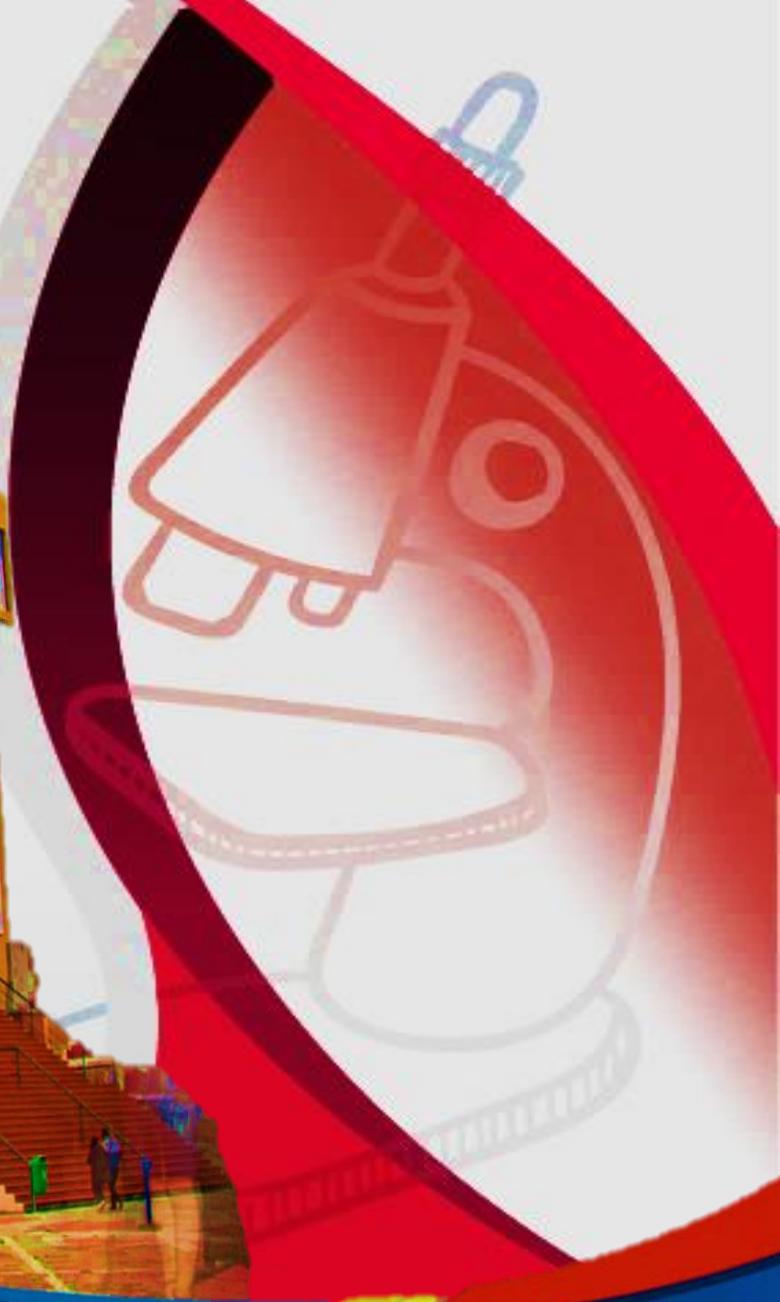
- Refleja la actividad intelectual de la institución con la difusión digital de los documentos.
- Aumenta la visibilidad de la institución al tener un sistema bibliotecario repositorio con estándares institucionales.

**“DISEÑO E IMPLEMENTACIÓN DE UNA BIBLIOTECA
VIRTUAL Y REPOSITORIO DE PROYECTOS DE
GRADO Y TESIS”**

CAPÍTULO II

**MARCO
TEÓRICO**

2023



2 CAPÍTULO II MARCO TEÓRICO

2.1 INTRODUCCIÓN

En el presente capítulo se presentará la parte teórica para la implementación del sistema de diseño e implementación de una biblioteca virtual y repositorio de proyectos de grado y tesis, describiendo las definiciones y conceptos más relevantes para una mejor comprensión y aclaración de dudas del presente trabajo de grado.

2.2 SISTEMA

Un sistema es un módulo ordenado de elementos que se encuentran interrelacionados y que interactúan entre sí. El concepto se utiliza tanto para definir a un conjunto de conceptos como a objetos reales dotados de organización, también puede mencionarse la noción del sistema informático, muy común en las sociedades modernas. Este tipo de sistemas denominan al conjunto de hardware, software y soporte humano que forman parte de una empresa u organización. Incluyen ordenadores con los programas necesarios para procesar datos y las personas encargadas en su manejo. (Lorenzon, 2020)

2.3 BIBLIOTECOLOGÍA

La bibliotecología es la ciencia que estudia las bibliotecas en todos sus aspectos, la conceptualización del término biblioteca, responde a una realidad compleja y múltiple, difícil de sistematizar en una definición globalizadora, más aún cuando se trata de una entidad de carácter dinámico y por lo tanto cambiante. (Matos, 2008)

2.3.1 Concepto de Biblioteca

No obstante, la palabra biblioteca deriva del latín biblioteca y esta a su vez lo hace del vocablo griego biblion = libro y thekes =caja, puede traducirse desde un punto de vista estrictamente etimológico como el lugar donde se guardan los libros. En la actualidad esta concepción se ha visto hace tiempo superada para pasar a referirse tanto a las colecciones bibliográficas como a las instituciones que las crean y las ponen en

servicio para satisfacer las demandas de los usuarios, la interpretación moderna del termino hace referencia a cualquier recopilación de datos almacenada en muchos otros formatos: microfilmes, revistas, grabaciones, películas, diapositivas, medios electrónicos como DVD y otros formatos que vienen imponiéndose recientemente como lo son los documentos digitales. (Yaguas, 2008)

2.3.2 Tipología de Bibliotecas

Con la finalidad de definir los distintos tipos de bibliotecas existentes han surgido distintas clasificaciones de bibliotecas según (UNESCO, 2001), la Organización de las Naciones Unidas Para la Educación, La ciencia y la Cultura en 1994 clasifica en:

- Bibliotecas nacionales
- Bibliotecas públicas
- Bibliotecas universitarias
- Biblioteca especial
- Bibliotecas escolares
- Bibliotecas especializadas

2.3.2.1 Bibliotecas Nacionales

“Bibliotecas Nacionales representan la cabecera del sistema de los estados. Están financiados con fondos públicos y cumplen una doble finalidad: proporcionar material bibliográfico de investigación para cualquier disciplina, y conservar y difundir el patrimonio cultural (referente a información registrada a lo largo del tiempo) de cada país”.

Las bibliotecas nacionales abarcan el ámbito de un país, su objetivo es reunir la producción y el patrimonio bibliográfico nacional, no siendo un factor determinante la comunidad de usuario a la que sirve, generalmente cada nación tiene su biblioteca nacional, en el caso de Bolivia esta se encuentra en Sucre bajo la denominación de Biblioteca y Archivo Nacional de Bolivia que fue creada el 6 de noviembre de 1883. (Yaguas, 2008)

2.3.2.2 Biblioteca Pública

“La Biblioteca Pública es un centro local de información que facilita a los usuarios toda clase de conocimiento e información, los servicios deben establecerse sobre la base de la igualdad para todas las personas sin tener en cuenta, su edad raza, sexo, religión, nacionalidad, idioma o condición social”. (Yaguas, 2008)

2.3.2.3 Biblioteca Universitaria

Son las bibliotecas de las facultades, escuelas y además unidades académicas de las universidades y centros de enseñanza superior difieren de las bibliotecas de investigación. Están al servicio de sus estudiantes para apoyar con los programas educativos, de investigación y de más servicios. (Yaguas, 2008)

2.3.2.4 Biblioteca Especial

La biblioteca especial se encuentra en un área geográfica especial, los usuarios tienen características especiales tanto en la forma de presentación como en su contenido, se complementa con otros de condiciones muy especiales según el usuario. (Yaguas, 2008)

2.3.2.5 Bibliotecas Escolares

Estas complementan los programas de las instituciones a las que pertenecen, aunque también disponen de libros no académicos para fomentar el hábito de la lectura. Muchas cuentan con distintos medios audiovisuales y electrónicos. Su financiación procede de las instituciones escolares en las que estas integran UNESCO. (Yaguas, 2008)

2.3.2.6 Bibliotecas Especializadas

Biblioteca establecida, mantenida y administrada por una firma comercial, una corporación privada, una asociación, un organismo estatal que tienen interés por una materia específica para atender las necesidades de información de sus miembros o

personal y alcanzar los objetivos de la organización. (Yaguas, 2008)

2.3.3 Características de la Biblioteca

Una biblioteca tiene funciones a cumplir como ser: la colección de información, organización de la información, distribución de información los cuales la describiremos a continuación:

La colección; consiste en la elección de los demás elementos que formará parte de la biblioteca. La selección se hace previo estudio a evaluación de los materiales disponibles de acuerdo con los pedidos realizados por el usuario relacionados con la difusión de conocimientos en diferentes áreas a las que pertenezca la biblioteca (Ordoñez, 2008)

Organización; consiste en la preparación de los materiales para su utilización por parte de los usuarios y además lectores que los necesiten (Ordoñez, 2008).

La organización comprende:

- Registro, sellado y rotulación.
- Clasificación del material, según el sistema de clasificación usado en la biblioteca y asignación de las asignaturas.
- Catalogación del material, esto es, consignación de los datos descriptivos de cada libro por: autor, título, notas tipográficas, números de páginas, ilustraciones y además informaciones; mantenimiento del catálogo de fichas y además ficheros de referencia necesaria.
- Hace que este material sea fácilmente accesible tanto física y lógicamente

Distribución; es la entrega en préstamo del material bibliográfico, para que ocurra esto se debe prestar un servicio de referencia, respondiendo a las consultas, preparando bibliográfico por medio del servicio de préstamo inter bibliotecario (Ordoñez, 2008).

2.3.3.1 Procedimientos Bibliotecarios

La prestación de un buen servicio de biblioteca se basa en una colección bien seleccionada y organizada. La importancia de los servicios técnicos, son un medio para que los servicios sean adecuados.

Existen una serie de servicios básicos que son denominar tradicionales entre los que destacan la lectura en sala y el préstamo domiciliario, que caso siempre han ofrecido las bibliotecas como servicio bibliotecario por excelencia en su sentido, amplio. (Ordoñez, 2008)

2.4 IMPLEMENTACIÓN

El concepto de "implementación" se refiere a todos los pasos que conlleva la puesta en marcha de un nuevo software o programa informático. Puede incluir cualquier aspecto, desde la instalación hasta la configuración, la prueba de ejecución y la edición. De modo ocasional, la palabra "implementación" también se utiliza para describir el mismo proceso. Este término también se puede utilizar cuando se habla de aplicar actualizaciones de software, parches o incorporaciones a las aplicaciones o software que ya existían. (Aduviri, 2013)

2.4.1 Implementación de un sistema informático

La implementación de sistemas informáticos es una de las etapas del desarrollo de sistemas. Es la etapa que le sigue al diseño.

La etapa de análisis nos dice qué funciones desempeñará el sistema, luego la etapa de diseño nos dice cómo organizar los distintos componentes. Ahora con la etapa de implementación es cuando realmente se comienza a escribir el código de programación.

En esta etapa se elige el entorno de desarrollo, las herramientas adecuadas y el lenguaje de programación apropiado. Estas elecciones dependerán en gran parte de la etapa de diseño.

Esta etapa permite desarrollar prototipos o sistemas de prueba que permiten ir comprobando y depurando el funcionamiento del sistema a medida que se construye.

La implementación de sistemas dentro del ciclo de vida de un sistema informático. La implementación es la cuarta etapa dentro del ciclo de desarrollo clásico de un sistema informático. (Alegsa, 2023)

2.5 DISEÑO

El diseño de sistemas informáticos es una de las etapas del desarrollo de sistemas. Es la etapa que le sigue a la factibilidad.

En la etapa de diseño donde se definen la arquitectura, componentes, módulos, interfaces y datos para el sistema que satisfacen los requerimientos especificados. El diseño del sistema puede verse como la aplicación o implementación de la teoría de sistemas para el desarrollo del producto. La división entre análisis y diseño de sistemas es más bien teórica, en ocasiones en la práctica se solapan ciertas actividades. (Espitia, 2020)

2.6 REPOSITORIO

Repositorio es un término que tiene su raíz etimológica en *repositorium*, un vocablo latino. Un repositorio es un **espacio** que se utiliza para **almacenar distintas cosas**. (Cabrera, 2015)

Lo que permite un repositorio, en definitiva, es **preservar y compartir el conocimiento**. Toda la **información** se reúne en un mismo lugar centralizado, facilitando el acceso a la misma. Si retomamos el ejemplo anterior, imaginemos lo difícil que supondría para una persona hallar todos diarios publicados durante más de cuatro décadas si dichos periódicos no estarían almacenados en un sitio único.

Según EcuRed (sf) lo define de la siguiente manera: Se considera un sistema de gestión de contenidos, que administra la producción científica en formato digital. Utilizan estándares abiertos para garantizar que sus contenidos sean accesibles y puedan ser buscados y recuperados para su uso posterior. Son un medio de

publicación científica. Ofrece otros servicios complementarios.

Un repositorio contiene permite importar, identificar, almacenar, preservar, recuperar y exportar un conjunto de objetos digitales, desde un portal web. (Cabrera, 2015)

2.6.1 Repositorio Digital

Una de las definiciones de repositorio digital indica que un repositorio digital es una plataforma centralizada creada por una Organización como un medio para crear, almacenar y gestionar información digital, facilitando las búsquedas y su posterior recuperación y acceso. El repositorio debe de estar desarrollado en base a estándares de metadatos y a protocolos de recolección e intercambio de información, implementando mecanismos para importar, exportar y también recolectar los contenidos digitales y sus metadatos. (Santiago, 2013)

Los repositorios científicos e institucionales pueden definirse como un conjunto de servicios prestados por las universidades o centros de investigación a su comunidad para recopilar, gestionar, difundir y preservar su producción científica digital a través de una colección organizada, de acceso abierto e interoperable. (Et, 2010)

2.6.1.1 Tipos de Repositorios Digitales

Existen diferentes tipos de repositorios digitales como: los institucionales, temáticos de datos, de software y gubernamentales, pero la mayoría de los autores resaltan dos grandes tipos de repositorios los cuales se refieren a los repositorios Temáticos y los Institucionales. (Et, 2010)

2.6.1.1.1 Repositorio Temático

Creados por un grupo de investigadores, una institución, etc. que reúnen documentos relacionados con un área temática particular. La temática suele ser social, de educación ciudadana o académica. (Et, 2010)

2.6.1.1.2 Repositorio Institucional

Desarrollado por organismos políticos, sociales y educativos como universidades e institutos o asociaciones, para depositar, usar y preservar la producción científica y académica que generan en formato digital y haciéndola accesible al público. De esta manera la institución ofrece un servicio acorde al movimiento de acceso abierto. (Et, 2010)

Entre otras definiciones que se han dado a repositorio institucional, se destacan los siguientes:

- define de forma muy escueta como “una colección digital que agrupa y preserva la producción intelectual de una o varias universidades”. Establece un objeto de aplicación muy amplio, el conjunto de la producción intelectual de la institución, en el que cabría más allá de los resultados de investigación y objetos de aprendizaje cualquier otro producto de carácter cultural producido por la institución. (Et, 2010)
- Un repositorio institucional es una base de datos compuesta de un grupo de servicios destinados a capturar, almacenar, ordenar, preservar y redistribuir la documentación académica de la institución en formato digital. (Et, 2010)
- considera que un repositorio debe cumplir al menos las siguientes características:
 - a) Es un servicio institucional abierto a toda la comunidad universitaria y a todo tipo de temáticas.
 - b) Su objetivo debe ser reunir, preservar y dar acceso a, entre otras cosas, la producción de los investigadores y docentes en múltiples formatos. Se excluiría cualquier repositorio que ponga límites a esta producción, por ejemplo, solo objetos de aprendizaje o solo de tesis, etcétera.

Debe recibir contenidos de forma activa, bien a través de un formulario web o simplemente a través de correo electrónico. (Et, 2010)

2.7 METADATOS

Datos altamente estructurados que describen información. Es "Información sobre información" o "datos sobre los datos", ya que proporciona información mínima necesaria para identificar un recurso e incluye información descriptiva sobre el contexto, calidad y condición o características de un dato u objeto que tiene la finalidad de facilitar la recuperación, autenticación, evaluación, preservación y/o interoperabilidad. (RODRÍGUEZ, 2002)

2.7.1 Tipos de Metadatos

Existen diversos tipos de metadatos, cada uno con su propio formato para describirlos. Según la información que proporcionen, existen metadatos sobre: el contenido, aspectos formales, derechos de autor y el contexto. (RODRÍGUEZ, 2002)

Según la función que proporcionan, se pueden clasificar en:

- **Descriptivos.** Describen e identifican recursos de información. Permite a los usuarios la búsqueda y recuperación de la información.
- **Estructurales.** Facilitan la navegación y la presentación de los recursos. Proporcionan información sobre la estructura interna de los documentos, así como la relación entre ellos.
- **Administrativos.** Facilitan la gestión de conjuntos de recursos. Incluye la gestión de derechos y sobre control de acceso y uso

Tabla 2.1 Tipos de metadatos

Tipo	Objetivo	Ejemplos
Descriptivos	Ayudan a la recuperación de información. Identifican recursos.	Dublín Core, Etiquetas HTML
Estructurales	Facilitan la presentación de recursos, su estructura interna y la relación entre ellos.	XML, RDF

Administrativos	Sirven para la gestión de documentos, de derechos y la preservación.	MOA2
------------------------	--	------

Nota. Qode (2014)

2.7.2 Clasificación

La clasificación de los metadatos definida por EcuRed es usado tres criterios siguientes:

- **Contenido.** Subdividir metadatos por su contenido es lo más común. Se puede separar los metadatos que describen el recurso mismo de los que describen el contenido del recurso. Es posible subdividir estos dos grupos más veces, por ejemplo, para separar los metadatos que describen el sentido del contenido de los que describen la estructura del contenido o los que describen el recurso mismo de los que describen el ciclo vital del recurso. (RODRÍGUEZ, 2002)
- **Variabilidad.** Según la variabilidad se puede distinguir metadatos mutables e inmutables. Los inmutables no cambian, no importa qué parte del recurso se vea, por ejemplo, el nombre de un fichero. Los mutables difieren de parte a parte, por ejemplo, el contenido de un vídeo. (RODRÍGUEZ, 2002)
- **Función.** Los datos pueden ser parte de una de las tres capas de funciones: subsimbólicos, simbólicos o lógicos. Los datos subsimbólicos no contienen información sobre su significado. Los simbólicos describen datos subsimbólicos, es decir añaden sentido. Los datos lógicos describen cómo los datos simbólicos pueden ser usados para deducir conclusiones lógicas, es decir añaden comprensión. (RODRÍGUEZ, 2002)

2.7.3 Los Metadatos Dublin Core

Los metadatos Dublin Core tratan de ubicar, dentro de Internet, los datos necesarios para describir, identificar, procesar, encontrar y recuperar un documento introducido en la red. Si este conjunto de elementos Dublin Core se lograra aceptar internacionalmente supondría que todos los procesos que **INDIZAN** documentos en **INTERNET** encontrarían, en la cabecera de los mismos, todos los datos necesarios para su indización y además estos datos serían uniformes. Si el Dublin Core lograra

estandarizar los metadatos de la cabecera de los documentos se facilitaría su indexación automática y mejoraría la efectividad de los **MOTORES DE BÚSQUEDA**. (Lapuente, 2013)

La Dublin Core Metadata Initiative (DCMI) es la responsable del desarrollo, estandarización y promoción del conjunto de los elementos de metadatos Dublin Core. Su objetivo es elaborar normas interoperables sobre metadatos y desarrollar vocabularios especializados en metadatos para la descripción de recursos que permitan sistemas de recuperación más inteligentes. (Lapuente, 2013)

Dublin Core es un **esquema de metadatos** elaborado y auspiciado por la **DCMI (Dublin Core Metadata Initiative)**, diseñado para proporcionar información descriptiva básica sobre cualquier tipo de recurso sin que importe el formato de origen (ya sean recursos físicos, recursos web u otros objetos como obras de arte, etc.). (Lapuente, 2013)

Los **4 principios que inspiran el conjunto de metadatos Dublin Core** son:

- **Simplicidad de creación y mantenimiento**, para que los mismos autores de los documentos puedan crear registros descriptivos sencillos, pero con suficiente información para poder recuperarlos en Internet.
- **Semántica universal**. Un obstáculo para el descubrimiento de información en Internet son las diferencias de terminología y prácticas en distintas áreas de conocimiento. Dublin Core puede ayudar a cualquier usuario no especializado a encontrar recursos web.
- **Cobertura internacional**. El conjunto de elementos Dublin Core se desarrolló en inglés, pero hoy está traducido a muchas otras lenguas, entre ellas el español.
- **Extensibilidad**. Dublin Core intenta mantener la simplicidad de las tareas de descripción de recursos web, junto a la necesidad de lograr una recuperación precisa. Los desarrolladores de Dublin Core reconocen la importancia de ofrecer mecanismos para que otras comunidades de expertos de metadatos

adopten los elementos Dublin Core para la información descriptiva central y puedan utilizar otros elementos más específicos en entornos locales. (Lapuente, 2013)

Desde la perspectiva de la comunidad Dublin Core, el paisaje de metadatos se encuentra caracterizado en términos de **4 niveles de interoperabilidad**:

- **Nivel 1: Definiciones compartidas**, se comparten vocabularios definidos en lenguaje natural.
- **Nivel 2: Interoperabilidad semántica formal**; se comparten vocabularios basados en una semántica formal RDF.
- **Nivel 3: Interoperabilidad sintáctica del conjunto de descripciones**, corresponde con el uso del DCMI Abstract Model en los metadatos.
- **Nivel 4: Interoperabilidad del perfil del conjunto de descripciones.**

Lamarca (2013) dijo que Podemos clasificar estos elementos en 3 grupos que indican la clase o el ámbito de la información que se guarda en ellos:

Tabla 2.2 Características y elementos Dublin Core

Contenido del Recurso	Propiedad intelectual	Instanciación del recurso
<ul style="list-style-type: none"> • Título • Materias y palabras Clave • Descripción • Fuente • Idioma • Relación • Cobertura 	<ul style="list-style-type: none"> • Creador • Editor • Colaborar • Derechos 	<ul style="list-style-type: none"> • Fecha • Tipo de Recurso • Formato • Identificador del Recurso

Nota. Lamarca (2013)

Veamos con mas detenimiento cada uno de estos 15 elementos que están recogidos En DCMI Metadadta Tems. (Lapunte, 2013)

Tabla 2.3 Elementos Dublin Core recogido en DCMI Metada Terms

Etiqueta del Elemento DC.	Descripción
DC. Title	Título: El nombre dado a un recurso. Típicamente, un título es el nombre formal por el que es conocido el recurso.
DC. Creator	Autor: La entidad primariamente responsable de la creación del contenido intelectual del recurso. Entre los ejemplos de un creador se incluyen una persona, una organización o un servicio. Típicamente, el nombre del creador podría usarse para indicar la entidad.
DC. Subject	Materias y palabras clave: El tema del contenido del recurso. Un tema será expresado como palabras clave, frases clave o códigos de clasificación que describan el tema de un recurso. Se recomienda seleccionar un valor de un vocabulario controlado o un esquema de clasificación formal.
DC. Description	Descripción: La descripción del contenido del recurso. La descripción puede incluir, pero no se limita a: un resumen, tabla de contenidos, referencia a una representación gráfica de contenido o una descripción de texto libre del contenido.
DC. Publisher	Editor: La entidad responsable de hacer que el recurso se encuentre disponible. Ejemplos de editores son una persona, una organización o un servicio. Típicamente, el nombre de un editor podría usarse para indicar la entidad.
DC. Contributor	Colaborador. La entidad responsable de hacer colaboraciones al contenido del recurso. Ejemplos de colaboradores son una persona, una organización o un servicio. Típicamente, el nombre del colaborador podría usarse para indicar la entidad.
DC. Date	Fecha: Una fecha asociada con un evento en el ciclo de vida del recurso. Típicamente, la fecha será asociada con la creación o disponibilidad del recurso. Se recomienda utilizar un valor de datos codificado definido en el documento "Date and Time Formats", que sigue la norma ISO 8601 que sigue el formato YYYY-MM-DD.
DC. Type	Tipo: la naturaleza o categoría del contenido del recurso. El tipo incluye términos que describen las categorías generales, funciones, géneros o niveles de agregación del contenido. Se recomienda seleccionar un valor de un vocabulario controlado (por ejemplo, el <i>DCMI Vocabulary</i> -DCMITYPE- Para describir la manifestación física o digital del recurso, se usa el elemento Formato.
DC. Format	Formato: la manifestación física o digital del recurso. El

	<p>formato puede incluir el tipo de media o dimensiones del recurso. Podría usarse para determinar el <i>software</i>, <i>hardware</i> u otro equipamiento necesario para ejecutar u operar con el recurso. Ejemplos de las dimensiones son el tamaño y la duración. Se recomienda seleccionar un valor de un vocabulario controlado (por ejemplo, la lista de Internet Media Types (MIME) que define los formatos de medios de ordenador).</p>
DC. Identifier	<p>Identificación: Una referencia no ambigua para el recurso dentro de un contexto dado. Se recomienda identificar el recurso por medio de una cadena de números de conformidad con un sistema de identificación formal, tal como un URI (que incluye el Uniform Resource Locator -URL, el Digital Object Identifier (DOI) y el International Standard Book Number (ISBN).</p>
DC. Source	<p>Fuente: Una referencia a un recurso del cual se deriva el recurso actual. El recurso actual puede derivarse, en todo o en parte, de un recurso fuente. Se recomienda referenciar el recurso por medio de una cadena o número de conformidad con un sistema formal de identificación.</p>
DC. Language	<p>Lengua: La lengua del contenido intelectual del recurso. Se recomienda usar RFC 3066 en conjunción con la ISO 639 [ISO639] , que define las etiquetas de dos y tres letras primarias para lenguaje, con subetiquetas opcionales. Ejemplo: "en" u "eng" para Inglés, "akk" para Acadio, y "en-GB" para inglés usado en Reino Unido.</p>
DC. Relation	<p>Relación: Una referencia a un recurso relacionado. Se recomienda referenciar el recurso por medio de una cadena de números de acuerdo con un sistema de identificación formal.</p>
DC. Coverage	<p>Cobertura: La extensión o ámbito del contenido del recurso. La cobertura incluiría la localización espacial (un nombre de lugar o coordenadas geográficas), el período temporal (una etiqueta del período, fecha o rango de datos) o jurisdicción (tal como el nombre de una entidad administrativa). Se recomienda seleccionar un valor de un vocabulario controlado (por ejemplo, del Thesaurus of Geographic Names (TGN) y que, donde sea apropiado, se usen preferentemente los nombres de lugares o períodos de tiempo antes que los identificadores numéricos tales como un conjunto de coordenadas o rangos de datos.</p>
DC. Rights	<p>Derechos: La información sobre los derechos de propiedad y sobre el recurso. Este elemento podrá contener un estamento de gestión de derechos para el recurso, o referencia a un servicio que provea tal información. La información sobre derechos a menudo corresponde a los derechos de propiedad intelectual, copyright y otros derechos de propiedad.</p>

Nota. Lamarca (2013)

A continuación, se muestra la tabla que recoge dichos elementos refinados extraídos

de DMCI Metadata Terms.

Tabla 2.4 Elementos refinados

Etiqueta del Elemento DC.	Descripción
abstract	Un resumen del contenido del recurso
accessRights	Información acerca de quién puede acceder al recurso o una indicación de su estatus de seguridad.
accrualMethod	El método por el cual los ítems se añaden a una colección.
accrualPeriodicity	La frecuencia con la que los ítems son añadidos a la colección.
alternative	Otra forma del título usada como un subtítulo o alternativa al título formar del recurso.
audience	Una clase de entidad para la que está indicado el uso del recurso.
available	Fecha (a menudo un rango) en la que el recurso comenzará o estará disponible
bibliographicCitation	Una referencia bibliográfica para el recurso.
conformsTo	Una referencia a un estándar establecido con el cual está conforme el recurso.
created	Fecha de creación del recurso.
dateAccepted	Fecha de aceptación del recurso (p. e. de la tesis por un dpto.de universidad, artículo para una revista, etc.).
dateCopyrighted	Fecha del establecimiento del copyright.
issued	Fecha de la puesta en circulación formal (p.e., publicación) de un recurso.
isFormatOf	El recurso descrito tiene el mismo contenido intelectual que el recurso referido, pero presentado en otro formato.
medium	El material o sustancia física del recurso.
modified	Fecha en que se ha cambiado el recurso
references	Las referencias de los recursos descritos, citas, u otros puntos de vista que se refieran el recurso.
tableOfContents	Una lista de sub-unidades del contenido del recurso.

Nota. Lamarca (2013)

2.8 INGENIERIA DE SOFTWARE

Con objeto de elaborar software listo para enfrentar los retos del siglo XXI, el lector debe aceptar algunas realidades sencillas:

- El software se ha incrustado profundamente en casi todos los aspectos de

nuestras vidas y, como consecuencia, el número de personas que tienen interés en las características y funciones que brinda una aplicación específica ha crecido en forma notable. Cuando ha de construirse una aplicación nueva o sistema incrustado, deben escucharse muchas opiniones. Y en ocasiones parece que cada una de ellas tiene una idea un poco distinta de cuáles características y funciones debiera tener el software. Se concluye que debe hacerse un esfuerzo concertado para entender el problema antes de desarrollar una aplicación de software.

- Los requerimientos de la tecnología de la información que demandan los individuos, negocios y gobiernos se hacen más complejos con cada año que pasa. En la actualidad, grandes equipos de personas crean programas de cómputo que antes eran elaborados por un solo individuo. El software sofisticado, que alguna vez se implementó en un ambiente de cómputo predecible y auto contenido, hoy en día se halla incrustado en el interior de todo, desde la electrónica de consumo hasta dispositivos médicos o sistemas de armamento. La complejidad de estos nuevos sistemas y productos basados en computadora demanda atención cuidadosa a las interacciones de todos los elementos del sistema. Se concluye que el diseño se ha vuelto una actividad crucial.
- Los individuos, negocios y gobiernos dependen cada vez más del software para tomar decisiones estratégicas y tácticas, así como para sus operaciones y control cotidianos. Si el software falla, las personas y empresas grandes pueden experimentar desde un inconveniente menor hasta fallas catastróficas. Se concluye que el software debe tener alta calidad.
- A medida que aumenta el valor percibido de una aplicación específica se incrementa la probabilidad de que su base de usuarios y longevidad también crezcan. Conforme se extienda su base de usuarios y el tiempo de uso, las demandas para adaptarla y mejorarla también crecerán. Se concluye que el software debe tener facilidad para recibir mantenimiento.

La ingeniería de software es una tecnología con varias capas, cualquier enfoque de

ingeniería (incluso la de software) debe basarse en un compromiso organizacional con la calidad. La administración total de la calidad, Six Sigma y otras filosofías similares alimentan la cultura de mejora continua, y es esta cultura la que lleva en última instancia al desarrollo de enfoques cada vez más eficaces de la ingeniería de software. El fundamento en el que se apoya la ingeniería de software es el compromiso con la calidad. El fundamento para la ingeniería de software es la capa proceso. El proceso de ingeniería de software es el aglutinante que une las capas de la tecnología y permite el desarrollo racional y oportuno del software de cómputo. El proceso define una estructura que debe establecerse para la obtención eficaz de tecnología de ingeniería de software. El proceso de software forma la base para el control de la administración de proyectos de software, y establece el contexto en el que se aplican métodos técnicos, se generan productos del trabajo (modelos, documentos, datos, reportes, formatos, etc.), se establecen puntos de referencia, se asegura la calidad y se administra el cambio de manera apropiada. Los métodos de la ingeniería de software proporcionan la experiencia técnica para elaborar software. Incluyen un conjunto amplio de tareas, como comunicación, análisis de los requerimientos, modelación del diseño, construcción del programa, pruebas y apoyo. Los métodos de la ingeniería de software se basan en un conjunto de principios fundamentales que gobiernan cada área de la tecnología e incluyen actividades de modelación y otras técnicas descriptivas. Las herramientas de la ingeniería de software proporcionan un apoyo automatizado o semiautomatizado para el proceso y los métodos. Cuando se integran las herramientas de modo que la información creada por una pueda ser utilizada por otra, queda establecido un sistema llamado ingeniería de software asistido por computadora que apoya el desarrollo de software. (Pressman, 2010).

Grafico 2.1 Fases de Ingeniería de software



Nota: (Pressman 2010)

2.9 INGENIERIA DE SOFTWARE

La ingeniería web es la aplicación de metodologías sistemáticas, disciplinadas y cuantificables al desarrollo eficiente, operación y evolución de aplicaciones web de alta calidad en la World Wide Web.

En este sentido la ingeniería 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 apoyan la evolución, diseño, desarrollo, implantación y evolución de dichas aplicaciones. (R., Pressman, 2000)

2.9.1 Características De La Ingeniería Web

La ingeniería web utiliza principios de la ingeniería de software, incluye nuevos enfoques, metodologías, herramientas, técnicas, guías y patrones para cubrir los requisitos únicos de Aplicaciones Web. (Pressman, 2010)

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

- Desarrollo web colaborativo.
- Modelo conceptual de aplicaciones web.
- Diseño de modelos de datos para sistemas de información WEB.
- Ingeniería web empírica.
- Ingeniería de requisitos para aplicaciones web.

- Usabilidad de aplicaciones web.
- Accesibilidad para la web.
- Metodologías de diseño web.
- Diseño de interfaces de usuario.
- Desarrollo y despliegue de servidores web.

El desarrollo de aplicaciones web, posee determinadas características que lo hacen diferente del desarrollo de aplicaciones o software tradicional y sistemas de información.

2.10 METODOLOGIA DE INGENIERIA WEB

Una metodología es un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar un nuevo software.

Una metodología de desarrollo de software se refiere al entorno que usa para estructurar, planificar y controlar el proceso de desarrollo de un sistema de información. (Pressman, 2010).

2.11 METODOLOGÍA INGENIERÍA WEB BASADA EN UML (UWE)

La Ingeniería web basada en UML es una metodología basada en el Proceso Unificado y UML (Lenguaje Unificado de Modelado) para el desarrollo de aplicaciones Web, cubre todo el ciclo de vida de las aplicaciones Web (Pasaje, 2005)

UWE es un método de ingeniería del software para el desarrollo de aplicaciones web basado en UML. Cualquier tipo de diagrama UML puede ser usado, porque UWE es una extensión de UML.

El enfoque UWE proporciona una notación específica de dominio, un proceso de desarrollo basado en modelos y soporte de herramientas para la ingeniería de aplicaciones web. La característica de UWE es el hecho de ser un enfoque basado en estándares que no se limita al uso de la UML "lingua franca", sino que también utiliza

XML como formato de intercambio de modelos, MOF1 para el metamodelado, los principios basados en modelos del enfoque MDA, el lenguaje de transformación de modelos QVT2 y XML.

Las principales razones para utilizar los mecanismos de extensión de la UML en lugar de una técnica de modelado patentada es la aceptación de la UML en el desarrollo de sistemas de software, la flexibilidad para la definición de un lenguaje de modelado específico de dominio web: un llamado perfil UML y un amplio soporte de modelado visual por parte de las herramientas existentes de UML CASE. (Pasaje, 2005)

2.11.1 Características

La metodología UWE define vistas especiales representadas gráficamente por diagramas UML, tales como el análisis de requerimientos, diseño conceptual modelo de navegación y el modelo de presentación. UWE no limita el número de diagramas posibles de una aplicación. (Pasaje, 2005)

- Es una metodología orientada a objetos, iterativa e incremental basada en UML.
- Se basa también en el proceso de desarrollo del software unificado.
- Proporciona un diseño sistemático y una generación semiautomática en las aplicaciones web a través de un Framework de publicación XML.
- UWE define su propio perfil UML en el cual se definen todos los elementos necesarios para modelar los diferentes aspectos de una aplicación web que son la presentación y la navegación entre otros.
- En esta metodología se proponen 2 tipos de diagramas para el modelado de la navegación que son: El modelo de espacio , el cual se definen todos los caminos navegacionales, es decir todas aquellas asociaciones de navegación directa entre los distintos objetos de la aplicación más bien conocida como clases de navegación, y el segundo modelo de estructura de navegación el cual se define la estructura de acceso que son utilizadas en la navegación, es decir todo aquello referente a menús, índices y demás.

2.11.2 Modelos de la metodología UWE

El método UWE consiste en la construcción de seis modelos de análisis y diseño. Dicha construcción se realiza dentro del marco de un proceso de diseño iterativo e incremental. Las actividades de modelado abarcan: el análisis de requerimientos, diseño conceptual, modelo de usuario, diseño de la navegación, de la presentación y diseño de la adaptación. (Zapata, 2019)

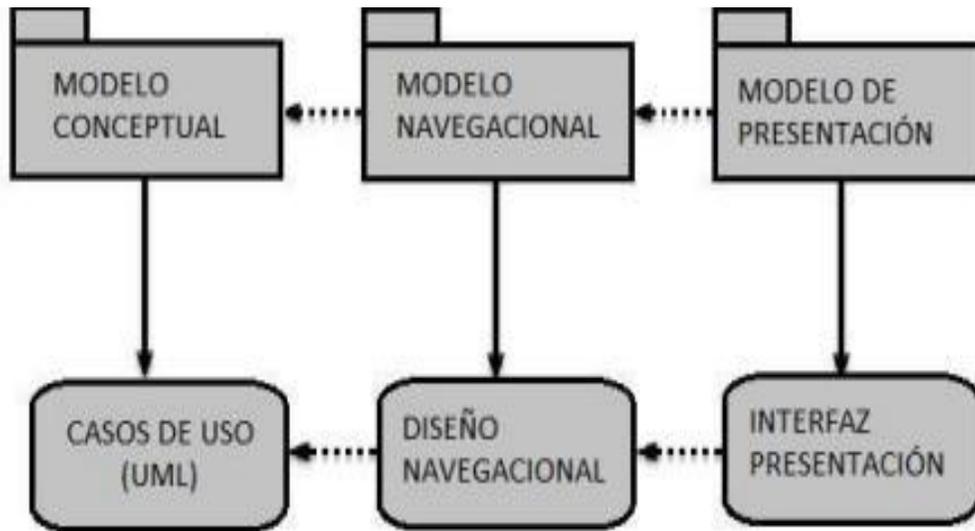
UWE proporciona guías para la construcción de modelos de forma sistemática, enfocándose en la personalización y en estudio de casos de uso.

Los principales artefactos que produce el método de diseño de UWE son los siguientes:

- Un **Modelo de Requerimientos** que captura los requerimientos del sistema.
- Un **Modelo Conceptual** para el contenido (modelo de contenido).
- Un **Modelo de Usuario**.
- Un **Modelo de Navegación** que comprende la estructura de la navegación.
- Un **Modelo de Presentación** que abarca modelos estáticos y dinámicos (modelo de estructura de la presentación, modelo del flujo de la presentación, modelo de interface abstracta de usuario, y modelo de ciclo de vida del objeto).
- Un **modelo de adaptación**.

En cuanto a los requisitos, UWE los clasifica dependiendo del carácter de cada uno. Además, distingue entre las fases de captura definición y validación de requisitos (Zapata, 2019).

Gráfico 2.2 Ingeniería Web Basada en UML



Nota: (Vargas, 2017)

2.11.3 Facetas de la metodología UWE

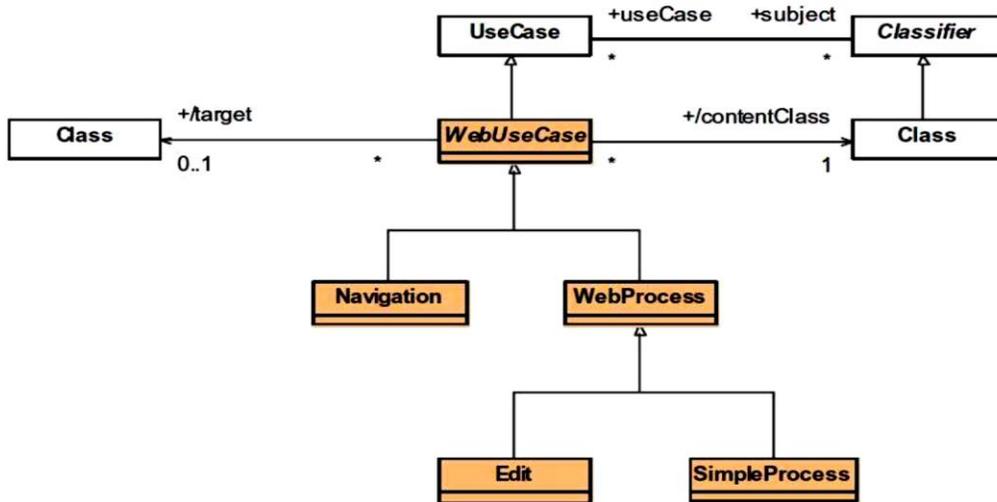
2.11.3.1 Modelo de análisis de requerimiento

El primer paso para el desarrollo de un sistema web que se especificará con UWE, es realizar la identificación de los requerimientos y plasmarlos en un modelo de requerimientos.

Los requerimientos pueden ser documentados en diferentes niveles de detalle, para este caso, UWE propone dos niveles de granularidad. En primera instancia se deben describir detalladamente las funcionalidades del sistema, las cuales son modeladas con casos de uso UML.

Como segundo paso, se debe elaborar una descripción de los casos de uso más detallada, por ejemplo, realizando diagramas de actividad UML donde se delimiten las responsabilidades y acciones de los actores involucrados.

Gráfico 2.3 Modelo de Caso de Uso de la Ingeniería Web

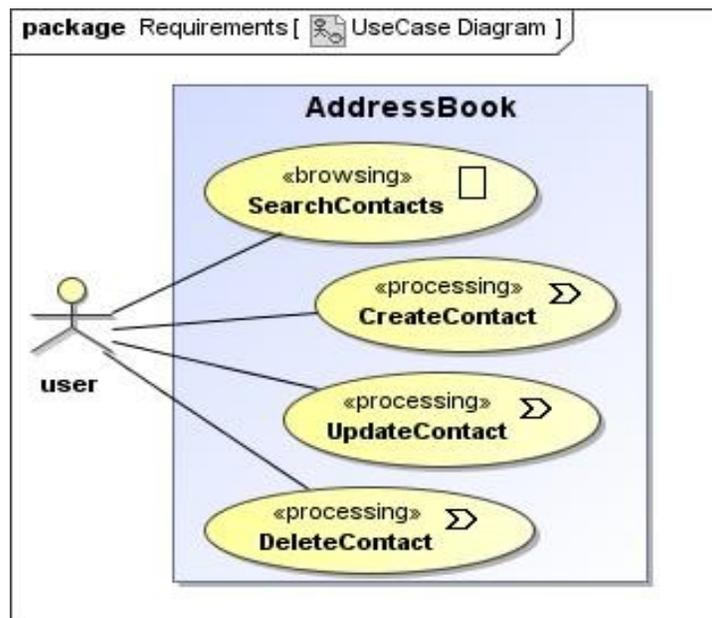


Nota. (Vargas, 2017)

2.11.3.2 Modelo 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.

Gráfico 2.4 Modelo de Caso de Uso de la Ingeniería Web

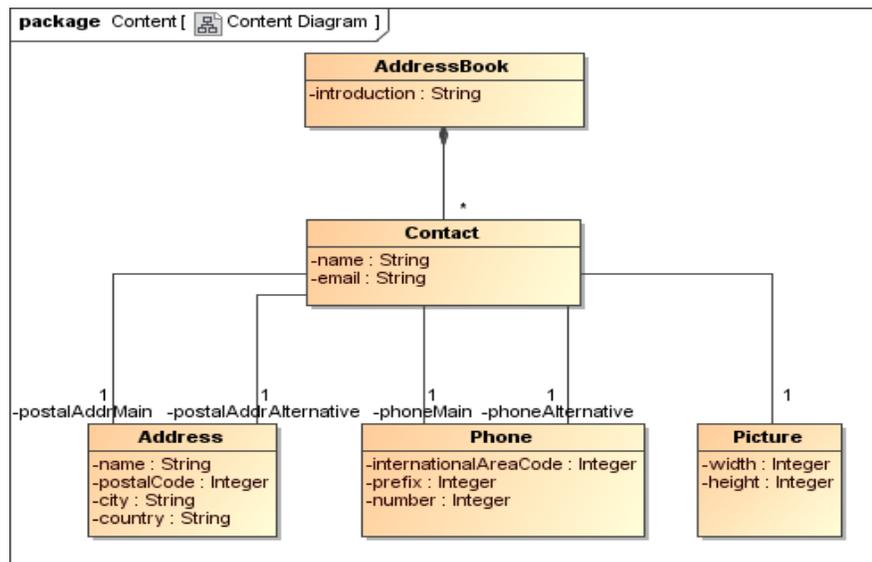


Nota. (Vargas, 2017)

2.11.3.3 Modelo Contenido

El modelo conceptual de un sistema software constituye una abstracción externa que describe mediante diagramas y notaciones con distinto grado de formalidad el conocimiento que debe poseer una persona acerca de un sistema, conocimiento que se encuentra almacenado en la Memoria a Largo Plazo.

Gráfico 2.5 Modelo Contenido de la Ingeniería Web



Nota. (Vargas, 2017)

2.11.3.4 Modelo de Navegación

El modelo de estructura de navegación define la estructura de nodos y links de una WebApp mostrando cómo se puede realizar la navegación utilizando elementos de acceso tales como índices, visitas guiadas, consultas y menús.

Los elementos de modelado son:

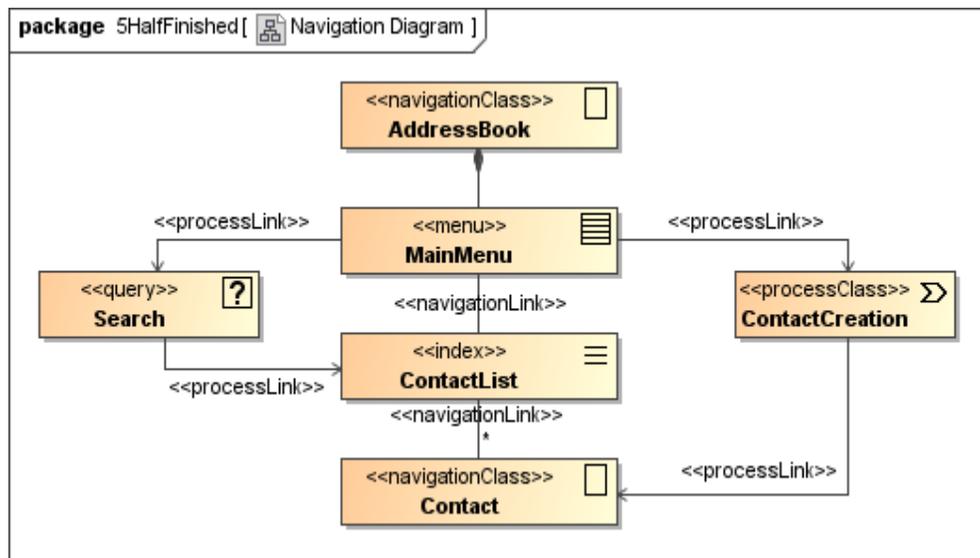
- **Clases de navegación**, que se denotan con (0), representan los nodos navegables de la estructura de hipertexto.
- **Links de navegación**, que muestran el vínculo directo entre las clases de

navegación.

- **Caminos de navegación alternativos**, los cuales son visualizados con el estereotipo <<menu>> ().
- **Permitivas de acceso**, las cuales se utilizan ya sea para llegar a múltiples instancias de una clase de navegación (<<index>> o <<guided tour>>) o para seleccionar ítems (<<query>>).
- **Clases de procesos ()**, las cuales modelan los puntos de entrada y de salida de los procesos de negocio. Cada clase de proceso está asociada a un caso de uso de proceso.
- **Links de procesos**, que representan el vínculo entre las clases de proceso y de navegación.

El modelo de estructura de navegación se representa mediante diagramas de clases UML estereotipados con las clases de navegación y procesos, menús y primitivas de acceso y así también los links de navegación y proceso.

Gráfico 2.6 Modelo Navegación de la Ingeniería Web



Nota. (Vargas, 2017)

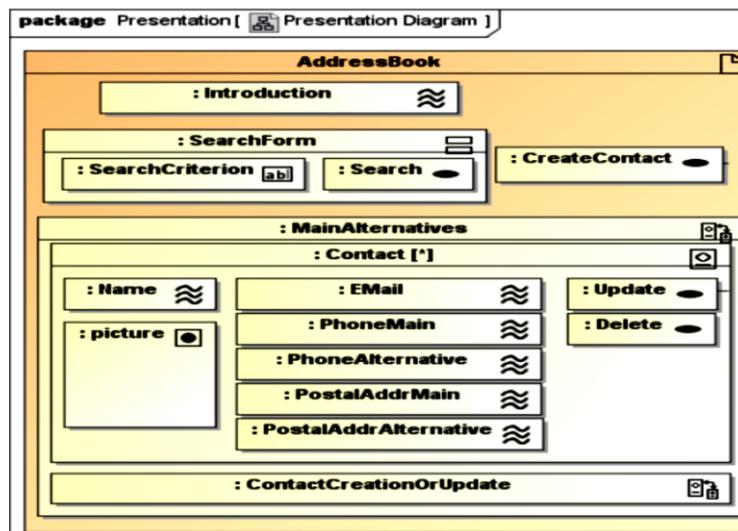
2.11.3.5 Modelo de Presentación

El modelo de presentación proporciona una vista abstracta de la interfaz de usuario (UI) de la aplicación web. Se basa en el modelo de navegación y describe qué elementos (por ejemplo, texto, elementos, links, formularios) se utilizarán para presentar los nodos de navegación.

Los elementos básicos del modelo de presentación son:

- **Clases de presentación**, las cuales se basan directamente en los nodos del modelo de navegación. Una clase de presentación () está compuesta por elementos de UI tales como, texto (<<text>>), vínculo (<<anchor>>), botón (<<button>>), imagen (<<image>>), formulario (<<form>>), y colección de vínculos (<<anchored collection>>)
- **Páginas web (<<page>>)**, que se utilizan para modelar la información proveniente de varios nodos de navegación y que se presentan en una misma página web.
- **Grupo de presentación (<<presentation group>>)**, el cual es un contenedor de clases de presentación, y a su vez de otros grupos de presentación.

Gráfico 2.7 Modelo de Presentación de la Ingeniería Web



Nota. (Vargas, 2017)

2.12 ARQUITECTURA DE SOFTWARE (MVC)

MVC es una propuesta de arquitectura del software utilizada para separar el código por sus distintas responsabilidades, manteniendo distintas capas que se encargan de hacer una tarea muy concreta, lo que ofrece beneficios diversos.

MVC se usa inicialmente en sistemas donde se requiere el uso de interfaces de usuario, aunque en la práctica el mismo patrón de arquitectura se puede utilizar para distintos tipos de aplicaciones. Surge de la necesidad de crear software más robusto con un ciclo de vida más adecuado, donde se potencie la facilidad de mantenimiento, reutilización del código y la separación de conceptos.

Su fundamento es la separación del código en tres capas diferentes, acotadas por su responsabilidad, en lo que se llaman Modelos, Vistas y Controladores, o lo que es lo mismo, Model, Views & Controllers, si lo prefieres en inglés. En este artículo estudiaremos con detalle estos conceptos, así como las ventajas de ponerlos en marcha cuando desarrollamos.

MVC es un "invento" que ya tiene varias décadas y fue presentado incluso antes de la aparición de la Web. No obstante, en los últimos años ha ganado mucha fuerza y seguidores gracias a la aparición de numerosos frameworks de desarrollo web que utilizan el patrón MVC como modelo para la arquitectura de las aplicaciones web. (Fausto, 2020)

El modelo MVC es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

2.12.1 Modelo

Es la capa donde se trabaja con los datos, por tanto, contendrá mecanismos para acceder a la información y también para actualizar su estado. Los datos los tendremos habitualmente en una base de datos, por lo que en los modelos tendremos todas las funciones que accederán a las tablas y harán los correspondientes selects, updates,

inserts, etc.

El modelo es el responsable de:

- Acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento
- Define las reglas de negocio (la funcionalidad del sistema). Un ejemplo de regla puede ser: "Si la mercancía pedida no está en el almacén, consultar el tiempo de entrega estándar del proveedor".
- Lleva un registro de las vistas y controladores del sistema.
- Si estamos ante un modelo activo, notificará a las vistas los cambios que en los datos pueda producir un agente externo (por ejemplo, un fichero por lotes que actualiza los datos, un temporizador que desencadena una inserción, etc.).

2.12.2 Vista

Las vistas, como su nombre nos hacen entender, contienen el código de nuestra aplicación que va a producir la visualización de las interfaces de usuario, o sea, el código que nos permitirá renderizar los estados de nuestra aplicación en HTML. En las vistas nada más tenemos los códigos HTML y PHP que nos permite mostrar la salida.

En la vista generalmente trabajamos con los datos, sin embargo, no se realiza un acceso directo a éstos. Las vistas requerirán los datos a los modelos y ellas se generará la salida, tal como nuestra aplicación requiera

Este presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.

Las vistas son responsables de:

- Recibir datos del modelo y los muestra al usuario.
- Tienen un registro de su controlador asociado (normalmente porque además lo instancia).

- Pueden dar el servicio de "Actualización ()", para que sea invocado por el controlador o por el modelo (cuando es un modelo activo que informa de los cambios en los datos producidos por otros agentes).

2.12.3 Controlador

Contiene el código necesario para responder a las acciones que se solicitan en la aplicación, como visualizar un elemento, realizar una compra, una búsqueda de información, etc.

En realidad, es una capa que sirve de enlace entre las vistas y los modelos, respondiendo a los mecanismos que puedan requerirse para implementar las necesidades de nuestra aplicación. Sin embargo, su responsabilidad no es manipular directamente datos, ni mostrar ningún tipo de salida, sino servir de enlace entre los modelos y las vistas para implementar las diversas necesidades del desarrollo.

Este responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista.

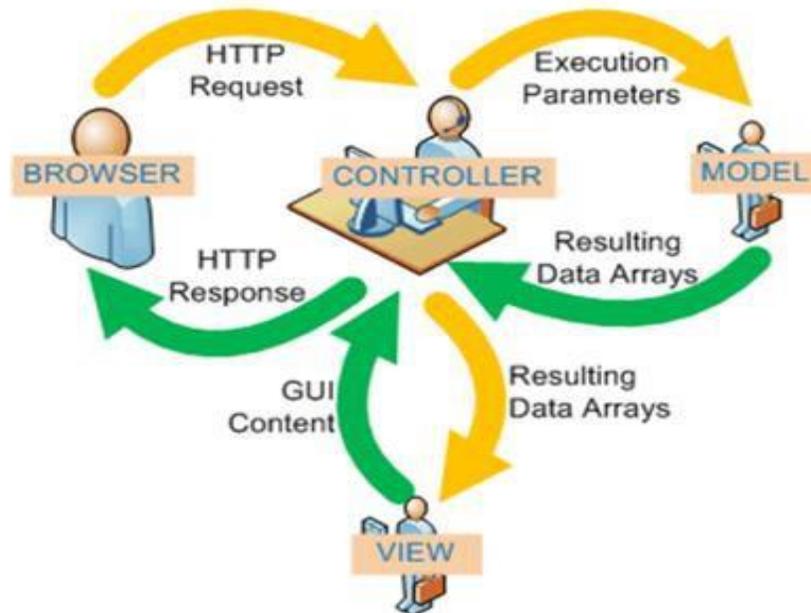
El controlador es responsable de:

- Recibe los eventos de entrada (un clic, un cambio en un campo de texto, etc.).
- Contiene reglas de gestión de eventos, del tipo "Si Evento Z, entonces Acción W". Estas acciones pueden suponer peticiones al modelo o a las vistas. Una de estas peticiones a las vistas puede ser una llamada al método "Actualizar ()". Una petición al modelo puede ser "Obtener_tiempo_de_entrega (nueva_orden_de_venta) ".

2.12.4 Características del Modelo Vista Controlador

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

Gráfico 2.8 Flujo de MVC



Nota. (Fausto, 2020)

- 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, actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada por el. 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 refleja 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 Unidad entre el modelo y la vista, permitiendo al modelo notificar a los interesados de cualquier

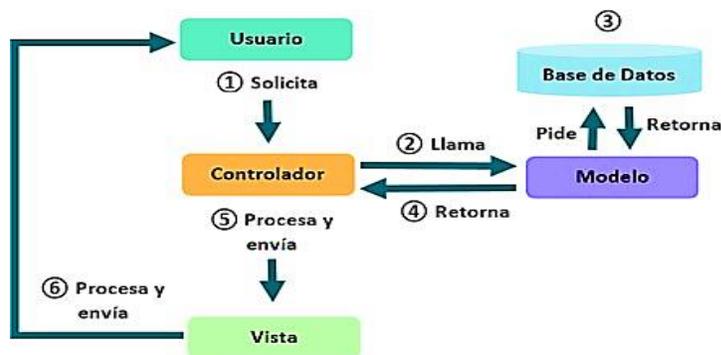
cambio.

- La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente. (Alicante, 2021)

2.12.5 Ciclo de vida del MVC

- El usuario realiza una petición.
- El controlador captura la petición del usuario.
- El controlador llama al modelo.
- El modelo interactúa con la base de datos, y retorna la información al controlador.
- El controlador recibe la información y la envía a la vista.
- La vista procesa la información recibida y la entrega de una manera visualmente.
- entendible al usuario.

Gráfico 2.9 Ciclo de Vida del MVC



Nota. (Alicante, 2021)

2.12.6 Ventajas y Desventajas

Las principales ventajas del uso del patrón MVC son:

- La separación del Modelo y la Vista, lo cual logra separar los datos, de su representación visual.
- Facilita el manejo de errores.
- Permite que el sistema sea escalable si es requerido.

- Es posible agregar múltiples representaciones de los datos.

Desventajas de MVC

Las principales desventajas del uso del patrón MVC son:

- La cantidad de archivos que se deben mantener incrementa considerablemente.
- La curva de aprendizaje es más alta que utilizando otros modelos.
- Su separación en capas, aumenta la complejidad del sistema.

2.13 MÉTRICAS DE CALIDAD DE SOFTWARE

Las Métricas de Calidad proporcionan una indicación de cómo se ajusta el software, a los requerimientos implícitos y explícitos del cliente.

El objetivo principal de la ingeniería del software es producir un producto de alta calidad. Para lograr este objetivo, los ingenieros del software deben utilizar mediciones que evalúen la calidad del análisis y los modelos de desafío, el código fuente, y los casos de prueba que se han creado al aplicar la ingeniería del software. Para lograr esta evaluación de la calidad en tiempo real, el ingeniero debe utilizar medidas técnicas que evalúan la calidad con objetividad, no con subjetividad.

El primer objetivo del equipo de proyecto es medir errores y defectos. Las métricas que provienen de estas medidas proporcionan una indicación de la efectividad de las actividades de control y de la garantía de calidad.

2.13.1 Calidad de Software

La calidad de un producto software depende del cumplimiento de un amplio listado de variables y de indicadores, que se deben tener en cuenta en las fases iniciales del ciclo de vida. Para alcanzarla es necesario darle mayor importancia a cuestiones como los requisitos no-funcionales, y a características como interoperabilidad, flexibilidad, usabilidad, confiabilidad, desempeño, escalabilidad, seguridad, etc. Entre los atributos

más relevantes se encuentran:

- **Seguridad.** Debido a iniciativas como cero papeles, mejoramiento ambiental, gestión documental, y la tendencia actual hacia la digitalización de la información de las empresas y la sociedad en general, y debido a que prácticamente todos los dispositivos electrónicos tienen conectividad, se vuelve sensible la información confidencial y protegida por marcos legales. Es por esto que este atributo toma importancia, y se hace imperante mejorar la seguridad y sus atributos de calidad, como la integridad, la privacidad y la confidencialidad, incrementado el nivel de la complejidad y la efectividad de los mecanismos software que prevengan las diferentes modalidades de ataque.
- **Disponibilidad y Confiabilidad.** La dependencia social, organizacional y de gobierno de los sistemas de información, exige el aseguramiento de una funcionalidad ininterrumpida de todas sus funciones, y cada vez son menos tolerables las interrupciones o fallas.
- **Usabilidad.** El acceso cada vez mayor a todo tipo de usuario hace que el desarrollo de software sea más exigente en el diseño de interfaz y factores relacionados con la usabilidad, y cada vez se va ir agudizando más este problema en el análisis y diseño, lo cual incrementará los costos.

2.13.2 Importancia de las Métricas

Las métricas de software se utilizan para propósitos estratégicos y son utilizadas en el proyecto para minimizar la planificación de desarrollo haciendo los ajustes 42

necesarios que eviten retrasos y reduzcan problemas y riesgos potenciales, son utilizadas también para evaluar la calidad de los productos en el momento actual y cuando sea necesario, modificando el enfoque técnico que mejore la calidad. Para establecer objetivos de mejora durante el proceso de desarrollo de software, se debe comprender el estado actual del desarrollo del software. Si no se mide, no hay una forma real de determinar si se está mejorando y si no se está mejorando, se está perdido. Disponible en: (Pressman,2008).

2.13.3 Norma ISO/IEC 25000

La calidad del producto, junto con la calidad del proceso, es uno de los aspectos más importantes actualmente en el desarrollo de Software. Relacionada con la calidad del producto, recientemente ha aparecido la familia de normas ISO/IEC 25000, que proporciona una guía para el uso de la nueva serie de estándares internacionales llamada Requisitos y Evaluación de Calidad de Productos de Software (SQuaRE - System and Software Quality Requirements and Evaluation).

La norma 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.

2.13.3.1 ISO/IEC 2500n – División de Gestión de Calidad

Las normas que forman este apartado definen todos los modelos, términos y definiciones comunes referenciados por todas las otras normas de la familia 25000. Actualmente esta división se encuentra formada por:

- **ISO/IEC 25000** - Guide to SQuaRE: contiene el modelo de la arquitectura de SQuaRE, la terminología de la familia, un resumen de las partes, los usuarios previstos y las partes asociadas, así como los modelos de referencia.
- **ISO/IEC 25001** - Planning and Management: establece los requisitos y orientaciones para gestionar la evaluación y especificación de los requisitos del producto software.

2.13.3.2 ISO/IEC 2500n – División de Modelo de Calidad

Las normas de este apartado presentan modelos de calidad detallados incluyendo características para calidad interna, externa y en uso del producto software. Actualmente esta división se encuentra formada por:

- **ISO/IEC 25010** - System and software quality models: describe el modelo de calidad para el producto software y para la calidad en uso. Esta Norma presenta las características y sub características de calidad frente a las cuales evaluar el producto software.

ISO/IEC 25012 - Data Quality model: define un modelo general para la calidad de los datos, aplicable a aquellos datos que se encuentran almacenados de manera estructurada y forman parte de un Sistema de Información.

2.13.3.3 ISO/IEC 2502 – División de Medición de Calidad

Estas normas incluyen un modelo de referencia de la medición de la calidad del producto, definiciones de medidas de calidad (interna, externa y en uso) y guías prácticas para su aplicación. Actualmente esta división se encuentra formada por:

- **ISO/IEC 25020** - Measurement reference model and guide: presenta una explicación introductoria y un modelo de referencia común a los elementos de medición de la calidad. También proporciona una guía para que los usuarios seleccionen o desarrollen y apliquen medidas propuestas por normas ISO.
- **ISO/IEC 25021** - Quality measure elements: define y especifica un conjunto recomendado de métricas base y derivadas que puedan ser usadas a lo largo de todo el ciclo de vida del desarrollo software.
- **ISO/IEC 25022** - Measurement of quality in use: define específicamente las métricas para realizar la medición de la calidad en uso del producto.
- **ISO/IEC 25023** - Measurement of system and software product quality: define específicamente las métricas para realizar la medición de la calidad de

productos y sistemas software.

- **ISO/IEC 25024** - Measurement of data quality: define específicamente las métricas para realizar la medición de la calidad de datos.

2.13.3.4 ISO/IEC 2503n – División de Requisitos de Calidad

Las normas que forman este apartado ayudan a especificar requisitos de calidad que pueden ser utilizados en el proceso de licitación de requisitos de calidad del producto software a desarrollar o como entrada del proceso de evaluación. Para ello, este apartado se compone de:

- **ISO/IEC 25030** - Quality requirements: provee de un conjunto de recomendaciones para realizar la especificación de los requisitos de calidad del producto software.

2.13.3.5 ISO/IEC 2504n – División de Evaluación de Calidad

Este apartado incluye normas que proporcionan requisitos, recomendaciones y guías para llevar a cabo el proceso de evaluación del producto software. Esta división se encuentra formada por:

- **ISO/IEC 25040** - Evaluation reference model and guide: propone un modelo de referencia general para la evaluación, que considera las entradas al proceso de evaluación, las restricciones y los recursos necesarios para obtener las correspondientes salidas.
- **ISO/IEC 25041** - Evaluation guide for developers, acquirers and independent evaluators: describe los requisitos y recomendaciones para la implementación práctica de la evaluación del producto software desde el punto de vista de los desarrolladores, de los adquirentes y de los evaluadores independientes.
- **ISO/IEC 25042** - Evaluation modules: define lo que la Norma considera un módulo de evaluación y la documentación, estructura y contenido que se debe utilizar a la hora de definir uno de estos módulos.
- **ISO/IEC 25045** - Evaluation module for recoverability: define un módulo para

la evaluación de la su característica Recuperabilidad (Recoverability).

Gráfico 2.10 División de calidad de Software



Nota. (ISO, 2021)

La calidad del producto software forma parte central como lo menciona ISO/IEC 25000, relacionando directamente las variables de funcionalidad, fiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad.

Según Monsalve (2010) “La calidad que pueden alcanzar los productos software, y en general cualquier producto, está sometida a como se desarrolla cada una de las etapas de la vida del producto, partiendo por la definición de la idea del producto hasta la entrega y mantenimiento del mismo”.

ISO/IEC 25000 define tres vistas diferenciadas en el estudio de la calidad de un producto:

- **Vista interna:** Se ocupa de las propiedades del sistema como: el tamaño, la complejidad o la conformidad con las normas de orientación a objetos. Puede utilizarse desde las primeras fases del desarrollo, permitiendo detectar

deficiencias en el sistema en etapas tempranas del ciclo de vida.

- **Vista externa:** Analiza el comportamiento del sistema en producción y estudia sus atributos, por ejemplo: el rendimiento de un software en una máquina determinada, el uso de memoria de un programa o el tiempo de funcionamiento entre fallos. Esta vista necesita que el sistema esté completo y en la fase de producción del producto, siendo muy dependiente de la máquina donde se ejecute.
- **Vista en uso:** Mide la productividad y efectividad del usuario final al utilizar el sistema.

Gráfico 2.11 Calidad del producto de software



Nota. (ISO, 2021)

2.13.4 Clasificación de las características y sub características

La Calidad del Software se clasifica en un conjunto estructurado de características y sub características de la siguiente manera:

2.13.4.1 Adecuación funcional

Representa la capacidad del producto software para proporcionar funciones que satisfacen las necesidades declaradas e implícitas, cuando el producto se usa en las condiciones específicas. Esta característica se subdivide a su vez en las siguientes sub características:

- **Compleitud funcional:** Grado en el cual el conjunto de funcionalidades cubre todas las tareas y los objetivos especificados por el usuario.
- **Corrección funcional:** Capacidad del producto o sistema para proveer resultados correctos con el nivel de precisión requerido.
- **Adecuación funcional:** Capacidad del producto software para proporcionar un conjunto apropiado de funciones para tareas y objetivos de usuario especificados.

2.13.4.2 Eficiencia de Desempeño

Esta característica representa el desempeño relativo a la cantidad de recursos utilizados bajo determinadas condiciones. Esta característica se subdivide a su vez en las siguientes sub características:

- **Comportamiento temporal:** Los tiempos de respuesta y procesamiento y las ratios de rendimiento de un sistema cuando lleva a cabo sus funciones bajo condiciones determinadas en relación con un banco de pruebas (benchmark) establecido.
- **Utilización de recursos:** Las cantidades y tipos de recursos utilizados cuando el software lleva a cabo su función bajo condiciones determinadas

2.13.4.3 Compatibilidad

- **Coexistencia:** Capacidad del producto para coexistir con otro software independiente, en un entorno común, compartiendo recursos comunes sin detrimento.
- **Interoperabilidad:** Capacidad de dos o más sistemas o componentes para intercambiar información y utilizar la información intercambiada.

2.13.4.4 Usabilidad

Capacidad del producto software para ser entendido, aprendido, usado y resultar

atractivo para el usuario, cuando se usa bajo determinadas condiciones. Esta característica se subdivide a su vez en las siguientes sub características:

- Capacidad para reconocer su adecuación: Capacidad del producto que permite al usuario entender si el software es adecuado para sus necesidades.
- **Capacidad de aprendizaje técnico:** Capacidad del producto que permite al usuario aprender su aplicación.
- **Capacidad para ser usado:** Capacidad del producto que permite al usuario operarlo y controlarlo con facilidad.
- **Protección contra errores de usuario:** Capacidad del sistema para proteger a los usuarios de hacer errores.
- **Estética de la interfaz de usuario:** Capacidad de la interfaz de usuario de agradar y satisfacer la interacción con el usuario.
- Accesibilidad técnica: Capacidad del producto que permite que sea utilizado por usuarios con determinadas discapacidades.

Tabla 2.5 Ecuaciones de la Usabilidad.

Variable	Ecuación
Usabilidad	$FU = [(\sum Xi/n) * 100]$ <p>Dónde: Xi: es la sumatoria de valores. n: es el número de preguntas.</p>

Nota. (Elaboración Propia)

Tabla 2.6 Valoración de las Preguntas

Escala	Valor
Muy bueno	5
Bueno	4
Regular	3
Malo	2

Fuente: (Elaboración Propia)

- **Entendimiento:** La capacidad que tiene el software para permitir al usuario entender si es adecuado, y de una manera fácil como ser utilizado para las tareas y las condiciones particulares de la aplicación. En este criterio se debe tener en cuenta la documentación y de las ayudas que el software entrega.
- **Aprendizaje:** La forma como el software permite al usuario aprender su uso. También es importante considerar la documentación.
- **Operabilidad:** La manera como el software permite al usuario operarlo y controlarlo.
- **Atracción:** La presentación del software debe ser atractiva al usuario. Esto se refiere a las cualidades del software para hacer más agradable al usuario, ejemplo, el diseño gráfico.
- **Conformidad de uso:** La capacidad del software de cumplir los estándares o normas relacionadas a su usabilidad.

2.13.4.5 Fiabilidad

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

- **Madurez:** Capacidad del sistema para satisfacer las necesidades de fiabilidad en condiciones normales.
- **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 restablecer el estado deseado del sistema en caso de interrupción o fallo.

2.13.4.6 Seguridad

Capacidad de protección de la información y los datos de manera que personas o sistemas no autorizados no puedan leerlos o modificarlos. Esta característica se subdivide a su vez en las siguientes sub características:

- **Confidencialidad:** Capacidad de protección contra el acceso de datos e información no autorizados, ya sea accidental o deliberadamente.
- **Integridad:** Capacidad del sistema o componente para prevenir accesos o modificaciones no autorizados a datos o programas de ordenador.
- **No repudio:** Capacidad de demostrar las acciones o eventos que han tenido lugar, de manera que dichas acciones o eventos no puedan ser repudiados posteriormente.
- **Responsabilidad:** Capacidad de rastrear de forma inequívoca las acciones de una entidad.
- **Autenticidad:** Capacidad de demostrar la identidad de un sujeto o un recurso.

2.13.4.7 Mantenibilidad

Esta característica representa la capacidad del producto software para ser modificado efectiva y eficientemente, debido a necesidades evolutivas, correctivas o perfectivas.

Esta característica se subdivide a su vez en las siguientes sub características:

- **Modularidad:** Capacidad de un sistema o programa de ordenador (compuesto de componentes discretos) que permite que un cambio en un componente tenga un impacto mínimo en los demás.
- **Reusabilidad:** Capacidad de un activo que permite que sea utilizado en más de un sistema software o en la construcción de otros activos.
- **Analizabilidad:** Facilidad con la que se puede evaluar el impacto de un determinado cambio sobre el resto del software, diagnosticar las deficiencias o

causas de fallos en el software, o identificar las partes a modificar.

- **Capacidad para ser modificado:** Capacidad del producto que permite que sea modificado de forma efectiva y eficiente sin introducir defectos o degradar el desempeño.
- Capacidad para ser probado: Facilidad con la que se pueden establecer criterios de prueba para un sistema o componente y con la que se pueden llevar a cabo las pruebas para determinar si se cumplen dichos criterios.

Tabla 2.7 Ecuaciones de la Mantenibilidad.

Variable	Ecuación
Mantenibilidad	$IMS = \frac{[Mt - (Fa + Fc + Fd)]}{Mt}$ <p>Dónde:</p> <p>Mt: número de módulos de la versión actual.</p> <p>Fc: número de módulos en la versión actual que se han modificado.</p> <p>Fa: número de módulos en la versión actual que se han añadido.</p> <p>Fd: número de módulos de la anterior versión que se han borrado en la versión actual.</p>

Fuente: (Elaboración propia)

2.13.4.8 Portabilidad

Capacidad del producto o componente de ser transferido de forma efectiva y eficiente de un entorno hardware, software, operacional o de utilización a otro. Esta característica se subdivide a su vez en las siguientes sub características:

- **Adaptabilidad:** Capacidad del producto que le permite ser adaptado de forma efectiva y eficiente a diferentes entornos determinados de hardware, software, operacionales o de uso.
- **Capacidad para ser instalado:** Facilidad con la que el producto se puede

instalar y/o desinstalar de forma exitosa en un determinado entorno.

- **Capacidad para ser reemplazado:** Capacidad del producto para ser utilizado en lugar de otro producto software determinado con el mismo propósito y en el mismo entorno.

2.14 ARQUITECTURA DE SOFTWARE

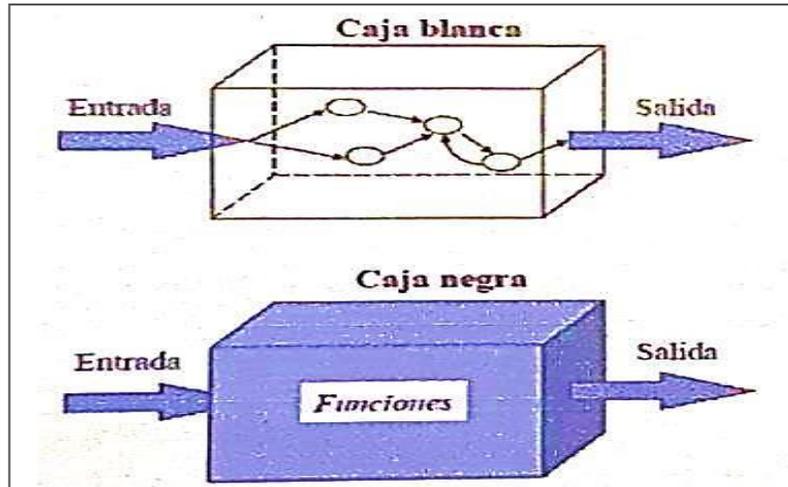
La arquitectura de software es un conjunto de patrones que proporcionan un marco de referencia necesario para guiar la construcción de un software, permitiendo a los programadores, analizar y todo el conjunto de desarrolladores de software compartir una misma línea de trabajo y cubrir todos los objetivos y restricciones de la aplicación. Es considerado el nivel más alto en el diseño de la arquitectura de un sistema puesto que establecen la estructura, funcionamiento e interacción entre las partes del software.

2.15 MÉTODOS DE PRUEBA DEL SOFTWARE

Las pruebas del software son un elemento crítico para la garantía de calidad del software. Las pruebas permiten validar y verificar el software, entendiendo como validación del software al proceso, externo al equipo del desarrollo, que determina si el software satisface los requerimientos, y verificación como el proceso interno que determina si el producto de una fase satisface las condiciones de dicha fase (Pressman, 2023)

Los objetivos de las pruebas se deben diseñar que tengan la mayor probabilidad de encontrar el mayor número de errores con la mínima cantidad de esfuerzo y tiempo posible (Pressman, 2023)

Gráfico 2.12 Filosofía de las Pruebas de Caja Blanca y Caja Negra



Nota. (Pressman, 2002)

Según (Pressman, 2023) las pruebas de **caja negra** se llevan a cabo sobre la interfaz del software. Se trata de demostrar que las funciones del software son operativas, que las entradas se manejan de forma adecuada y que se produce el resultado. Las pruebas de **caja blanca** se centran en la estructura lógica interna del software. Se basa en un examen detallado de los procedimientos caminos lógicos del sistema.

2.15.1 Prueba de Caja Blanca

Caja Blanca estructural basadas en información sobre cómo el software ha sido diseñado o codificado. Se comprueban los caminos lógicos del software proponiendo casos de prueba que ejerciten conjuntos específicos de condiciones y/o bucles. (Pressman, 2023)

2.15.1.1 Pruebas del camino Básico

La prueba del camino básico es una técnica de prueba de caja blanca propuesta inicialmente por Tom McCabe[MCC76]. El método del camino básico permite al diseñador de casos de prueba obtener una medida de la complejidad lógica de un diseño procedimental (Pressman, 2023)

Los pasos a realizar para aplicar esta técnica son:

- Representar el programa en un grafo de flujo.
- Calcular la complejidad ciclomática.
- Determinar el conjunto básico de caminos independientes.
- Derivar los casos de prueba que fuerzan la ejecución de cada camino.

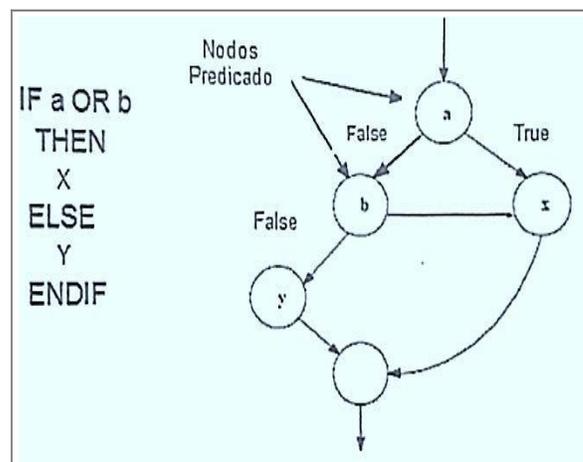
2.15.1.2 Notación de Grafo de Flujo

El grafo de flujo se utiliza para representar flujo de control lógico de un programa. Para ello se utilizan los tres elementos siguientes:

Nodos: representan cero, una o varias sentencias en secuencia. Cada nodo comprende como máximo una sentencia de decisión (bifurcación). (Pressman, 2023)

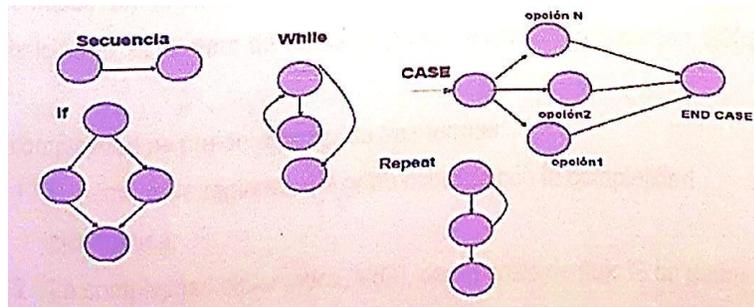
- *Aristas*: líneas que unen dos nodos.
- *Regiones*: áreas delimitadas por aristas y nodos. Cuando se contabilizan las regiones de un programa debe incluirse el área externa como una región más.
- *Nodos predicados*: cuando en una condición aparecen uno o más operadores lógicos (AND, OR, XOR, ...) se crea un nodo distinto por cada una de las condiciones simples. Cada nodo generado de esta forma se denomina nodo predicado.

Gráfico 2.13 Condición Múltiple



Nota: (Pressman, 2002)

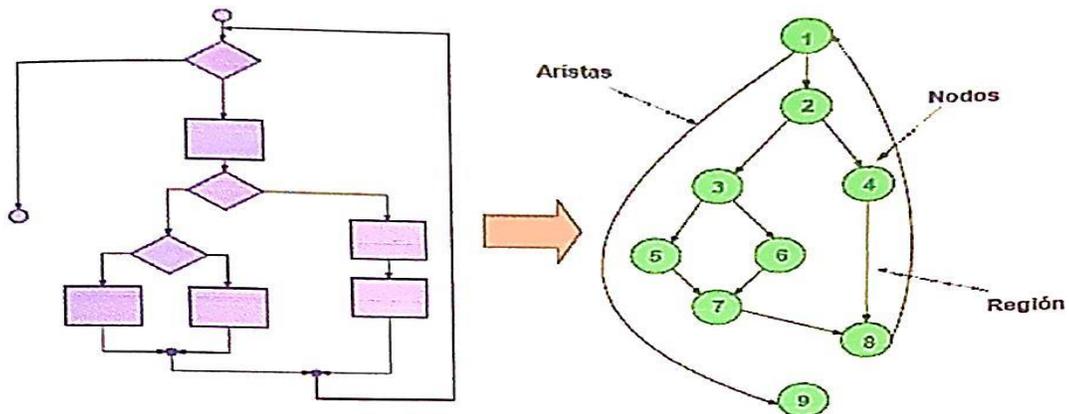
Gráfico 2.14 Grafo de las Estructuras Lógicas de un Programa



Nota: (Pressman, 2002)

Muestra un grafo de flujo del diagrama de módulos correspondientes. Nótese como la estructura principal correspondiente a un while, y dentro del bucle se encuentran anidados dos constructores *if*.

Gráfico 2.15 Grafo de Correspondiente a un Diagrama de Módulos



Nota: (Pressman, 2002)

2.15.1.3 Complejidad ciclomática

La complejidad ciclomática es una métrica del software que proporciona una medida cuantitativa de la complejidad lógica de un programa. En el contexto del método de prueba del camino básico, el valor de la complejidad ciclomática define el número de caminos independientes de dicho programa, y por tanto, el número de casos de prueba a realizar. (Pressman, 2002).

La complejidad se puede calcular de tres formas:

- El número de regiones de grafo coincide con la complejidad ciclomática
- La complejidad ciclomática, $V(G)$, de un grafo de flujo G se define como **$V(G) = \text{Aristas} - \text{Nodos} + 2$**
- La complejidad ciclomática, $V(G)$, de un grafo de flujo G se define como **$V(G) = \text{Nodos Predicado} + 1$**

Donde P es el número de nodos predicado contenido en el grafo de flujo.

2.15.1.4 Rutas de Programas independientes

Una ruta independiente es cualquiera que introduce al menos un nuevo conjunto de anuncios de procesamiento o una nueva condición en el programa. Cuando se establece como un gráfico de flujo, una ruta independiente debe moverse a lo largo de al menos de una arista que no se haya recorrido antes de definir la ruta. (Pressman, 2023)

2.15.2 Prueba de Caja Negra

Las pruebas de caja negra, también denominada prueba de comportamiento, se centran en los requisitos funcionales del software. La prueba de caja negra no es una alternativa a las técnicas de prueba de caja blanca. Más bien se trata de un enfoque complementario que intenta descubrir diferentes tipos de errores que los métodos de caja blanca. (Pressman, 2023)

Si una condición de entrada especifica un *rango de valores*, identificar una clase de equivalencia válida y dos clases no válidas. Por ejemplo, si un contador puede ir de 1 a 999, la clase válida sería “ $1 \leq \text{contador} \leq 999$ ”. Mientras que las clases no válidas serían “ $\text{contador} < 1$ ” y “ $\text{contador} > 999$ ”. (Pressman, 2002).

Si una condición de entrada especifica un valor o número de valores, identificar una clase válida y dos clases no válidas. Si una condición de entrada especifica un conjunto

de valores de entrada, identificar una clase de equivalencia válida y una no válida. (Pressman, 2023)

2.16 MÉTODOS DE ESTIMACIÓN DE COSTO DE SOFTWARE

Todo proyecto de ingeniería de software debe partir con un buen plan, pero lamentablemente, la planificación es una tarea nada trivial. Uno de los aspectos que dificultan la labor de administradores y jefes de proyecto en torno a la planificación es la difícil tarea de realizar una estimación de costos y plazos realista.

La estimación es más arte que ciencia; también es parte de la etapa de la planificación y algunas actividades de la ingeniería. La diferencia en la estimación de costos entre ingeniería de software y otras disciplinas es que en ingeniería de software lo principal para las personas es el costo; y en otras disciplinas el costo de las cosas materiales depende de la actividad.

Existen técnicas para la estimación de costos, pero para ello se requiere experiencia, acceso a una buena información histórica y coraje para confiar en medidas cuantitativas cuando todo lo que existe son datos cualitativos. El manejador de costo principal para un proyecto de desarrollo de software es sin duda el tamaño del producto. La medida del tamaño debe ser tal que esté en relación directa con el esfuerzo de desarrollo, por lo que las métricas de tamaño tratan de considerar todos los aspectos que influyen en el costo, como tecnología, tipos de recursos y complejidad. (Garza, 2016).

2.16.1 Modelo COCOMO

El modelo constructivo de costes COCOMO (Constructive Cost Model) desarrollado por Barry W. Boehm, se engloba en el grupo de los modelos algorítmicos que tratan de establecer una relación matemática la cual permite estimar el esfuerzo y el tiempo requerido para desarrollar un producto. (Pressman, 2010).

2.16.2 Método de estimación de Costos COCOMO II

COCOMO II permite realizar estimaciones en función del tamaño del software y de un conjunto de factores de coste y de escala.

- En los factores de coste se incluyen aspectos relacionados con la naturaleza del sistema, equipo y características propias de proyecto.
- Los factores de escala incluyen la parte de escala producida a medida que un proyecto de software incrementa su tamaño.

Por un lado, COCOMO II define tres modos de desarrollo de software o tipos de proyectos las cuales son las siguientes.

- **Orgánico:** proyecto relativamente sencillo, menores de 50 KDLC líneas de código, en los cuales se tiene experiencia de proyectos similares y se encuentran en entorno estable.
- **Semi-acoplado:** proyectos intermedios en complejidad y tamaño (menores de 300 KDLC), donde la experiencia en este tipo de proyecto es variable, y las restricciones intermedias.
- **Empotrado:** proyectos bastantes complejos, en los que apenas se tiene experiencia y se engloban en un entorno de gran innovación técnica. Además, se trabaja con unos requisitos muy restrictivos y de gran volatilidad.

Tabla 2.8 detalle de Coeficientes COCOMOII

Proyecto de Software	a	b	c	d
Orgánico	2,4	1,05	2,5	0,38
Semi- acoplado	3,0	1,12	2,5	0,35
Empotrado	3,6	1,20	2,5	0,32

Nota: (Pressman, 2010)

Y por otro lado existen diferentes modelos que define COCOMO:

- **Modelo básico:** Se basa exclusivamente en el tamaño expresado en LDC.
- **Modelo intermedio:** Además del tamaño del programa incluye un conjunto de medidas subjetivas llamadas conductores de costes.
- **Modelo avanzado:** Incluye todo lo del modelo intermedio además del impacto de cada conductor de coste en las distintas fases de desarrollo.

Vamos a utilizar la técnica COCOMO para una estimación del esfuerzo necesario para la realización del proyecto. para la realización del COCOMO previamente necesitamos conocer el número de líneas de código, posteriormente para poder realizar los cálculos del método de estimación usamos las siguientes ecuaciones:

Tabla 2.9 Ecuaciones del Método COCOMO II

Variable	Ecuación	Tipo/Unidad
Esfuerzo requerido por el proyecto	$E = a \times (KLDC)^b \times FAE$	Personas / Mes
Tiempo requerido por el proyecto	$T = c \times (E)^d$	Meses
Número de personas requeridas para el proyecto	$NP = ET$	Personas
Costo Total	$CT = Sueldo\ Mes \times NP \times T$	\$us.

Nota. (Prentice – Hall, 1981)

Y por otro lado también hemos de hallar la variable FAE, la cual se obtiene mediante la multiplicación de los valores evaluados en los diferentes 15 conductores de coste que se observan en la siguiente tabla:

Tabla 2.10 Atributos FAE

Atributos que afectan al coste	Valor					
	Muy Bajo	Bajo	Nominal	Alto	Muy Alto	Extra Alto
Atributos del Software						
Fiabilidad del software	0,75	0,88	1,00	1,15	1,40	
Tamaño base de datos		0,94	1,00	1,08	1,16	
Complejidad del producto	0,7	0,85	1,00	1,15	1,30	1,65
Atributos del 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	
Capacidad de programadores	1,42	1,17	1,00	0,86	0,70	
Experiencia en S.O. Usado	1,21	1,10	1,00	0,90		
Experiencia en el lenguaje	1,14	1,07	1,00	0,95		
Atributos del Hardware						
Restricciones de Tiempo de ejecución			1,00	1,11	1,30	1,66
Restricciones de memoria			1,00	1,06	1,21	1,56
Volatilidad de máquina virtual		0,87	1,00	1,15	1,30	
Tiempo de respuesta		0,87	1,00	1,07	1,15	
Atributos del Proyecto						
Uso de técnicas actuales de programación	1,24	1,10	1,00	0,91	0,82	
Uso de herramientas de software	1,24	1,10	1,00	0,91	0,83	
Restricciones de tiempo de desarrollo	1,23	1,08	1,00	1,04	1,10	

Nota: (Pressman, 2010)

Descripción de Valores:

Atributos de software

- **RELY:** garantía de funcionamiento requerida al software. Indica las posibilidades consecuencias para el usuario en el caso que existan defectos en el producto.
- **DATA:** tamaño de la base de datos en relación con el tamaño del programa datos en bytes y K es el tamaño del programa en cantidad de líneas de código.
- **CPLX:** representa la complejidad del producto.

Atributos de hardware

- **TIME:** limitaciones en el porcentaje del uso de la CPU.
- **STOR:** limitaciones en el porcentaje del uso de la memoria.
- **VIRT:** volatilidad de la máquina virtual.
- **TURN:** tiempo de respuesta requerido.

Atributos del personal

- **ACAP:** calificación de los analistas.
- **AEXP:** experiencia del personal en aplicaciones similares.
- **PCAP:** calificación de los programadores.
- **VEXP:** experiencia del personal en la máquina virtual.
- **LEXP:** experiencia en el lenguaje de programación a usar.

Atributos del proyecto

- **MODP:** uso de prácticas modernas de programación.
- **TOOL:** uso de herramientas de desarrollo de software.
- **SCED:** limitaciones en el cumplimiento de la planificación

Disponible en: (COCOMO, 2016)

2.16.3 Estimación de Esfuerzo y tiempo de Desarrollo

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

Esfuerzo Total (E) necesario para el desarrollo de un proyecto:

- Modo Orgánico: $E=2'4$ (KLDC) 1'05 Hombres/Mes
- Modo Semilibre: $E=3'0$ (KLDC) 1'12 Hombres/Mes
- Modo F. Restringido: $E=3'6$ (KLDC) 1'20 Hombres/Mes

Productividad:

- $PR = LDC/MM$ Líneas/Hombre-mes

Tiempo de Desarrollo:

- Modo Orgánico: $T=2'5$ (E) 0'38 Meses
- Modo Semilibre: $T=2'5$ (E) 0'35 Meses
- Modo F. Restringido: $T=2'5$ (E) 0'32 Meses

2.16.3.1 Modelo Composición de Aplicación

La fórmula propuesta en este modelo es la siguiente:

$$PM = NOP / PROD$$

Donde:

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

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

- OP (Puntos Objeto): Tamaño del software a desarrollar expresado en Puntos Objeto %reuso: Porcentaje de reuso que se espera lograr en el proyecto.
- **PROD**: Es la productividad promedio determinada a partir del análisis de datos de proyectos en (Banker 1994).

2.16.3.2 El Modelo de Diseño anticipativo y post-arquitectura

Los modelos de Diseño Anticipado y Post-Arquitectura se basan en la misma filosofía a la hora de proporcionar una estimación. Como hemos indicado ya su principal diferencia se produce en la cantidad y detalle de la información que se utiliza para obtener la estimación en cada uno de ellos. La fórmula básica para obtener una estimación de esfuerzo con ambos modelos es:

$$MMNOMINAL=A*(Size)B$$

Esta ecuación calcula el esfuerzo nominal para un proyecto de un tamaño dado expresado en Meses-persona (MM).

Las entradas son la medida del desarrollo del software, una constante, A, y un factor de escala, B. La medida está en unidades de líneas de código fuente (KSLOC). Esto se deriva de la medida de módulos software que constituirán el programa de aplicación, puede estimarse también a partir de Puntos de Función sin ajustar convirtiendo a SLOC y luego dividiendo por 1000.

El factor de escala (exponencial B), explica el ahorro o gasto relativo de escala encontrado en proyectos software de distintos tamaños. La constante A, se usa para cortar los efectos multiplicativos de esfuerzo en proyectos de tamaño incremental.

Diseño Anticipado y se describen sus componentes.

- Constante A

Como ya hemos visto anteriormente la constante A, se usa para capturar los efectos multiplicativos de esfuerzo en proyectos de tamaño incremental. Provisionalmente se le ha estimado un valor de 2.45.

- Variable Size

$$\text{Donde: } Size = Size * [1 + (BRAK/100)]$$

2.17 SEGURIDAD DE LA INFORMACIÓN

2.17.1 Estándar ISO/IEC 27000

Es un estándar para la seguridad de la información fue aprobada y publicada como estándar internacional en octubre de 2005 por ISO. Especifica los requisitos necesarios para establecer, implantar, mantener y mejorar un sistema de gestión de la seguridad de la información (SGSI).

SGSI (sistema de gestión de la seguridad de la información), es el conjunto de políticas y procedimientos que normalizan la gestión de la seguridad de la información, de toda una organización o de uno o varios procesos de negocio, debe garantizar la disponibilidad, confidencialidad e integridad de la información.

2.17.1.1 La ISO 27002

La ISO 27002 consiste en una guía de buenas prácticas que permiten a las organizaciones mejorar la seguridad de su información, define una serie de objetivos de control y gestión que deberían ser perseguidos por las organizaciones. (Maidana, 2004).

2.17.2 Uso de Password

Una contraseña o clave (en inglés password) es una forma de autenticación que utiliza información secreta para controlar el acceso hacia algún recurso. La contraseña debe mantenerse en secreto ante aquellos a quienes no se les permite el acceso. (Maidana, 2004).

Gráfico 2.16 Estructura ISO 27000



Nota. (Maidana, 2014).

2.18 HERRAMIENTAS

2.18.1 Gestor de base de datos MySQL

Es el sistema de administración de bases de datos (Database Management System, DBMS) más popular, desarrollado y proporcionado por MySQL AB. Es un sistema de gestión de base de datos relacional, multihilo y multiusuario. MySQL fue escrito en C y C++ y destaca por su gran adaptación a diferentes entornos de desarrollo, permitiendo su interacción con los lenguajes de programación más utilizados como PHP, Perl y Java y su integración en distintos sistemas operativos.

MySQL es un sistema de base de datos disponible gratuitamente. Sin embargo, hay varias ediciones de pago también disponibles con las que puede utilizar la funcionalidad avanzada.

MySQL es fácil de usar en comparación con otros programas de bases de datos como Microsoft SQL Server y Oracle database etc. Puede ser usado con cualquier lenguaje de programación, pero es ampliamente usado con PHP.

MySQL puede ejecutarse en múltiples plataformas como Linux, Windows, Unix y un esquema de información para definir y administrar sus metadatos. Puede instalarlo

en su sistema local o incluso en el servidor. (MySQL revista).

Características clave de MySQL

MySQL presenta algunas ventajas que lo hacen muy interesante para los desarrolladores. La más evidente es que trabaja con bases de datos relacionales, es decir, utiliza tablas múltiples que se interconectan entre sí para almacenar la información y organizarla correctamente.

Al ser basada en código abierto es fácilmente accesible y la inmensa mayoría de programadores que trabajan en desarrollo web han pasado usar MySQL en alguno de sus proyectos porque al estar ampliamente extendido cuenta además con una ingente comunidad que ofrece soporte a otros usuarios. Pero estas no son las únicas características como veremos a continuación:

- **Facilidad de gestión:** Es bastante fácil de descargar y utilizar el software.
- **Alto rendimiento:** Le proporciona utilidades de carga rápida con diferente caché de memoria.
- **Escalable:** Con MySQL, puede escalar en cualquier momento. Es realmente fácil crear almacenes de datos que incluyen una enorme cantidad de datos.
- **Compatibilidad:** MySQL es compatible con todas las plataformas modernas como Windows, Linux, Unix.
- **Rendimiento:** MySQL le ofrece resultados de alto rendimiento sin perder la funcionalidad esencial.
- **Seguridad completa de los datos:** Sólo los usuarios autorizados pueden acceder a la base de datos. Completa seguridad de los datos.
- **Bajo Costo:** Es gratis de usar.
- **Eficiencia de memoria:** MySQL tiene baja pérdida de memoria.
- **Arquitectura Cliente y Servidor:** MySQL basa su funcionamiento en un modelo cliente y servidor. Es decir, clientes y servidores se comunican entre sí de manera diferenciada para un mejor rendimiento. Cada cliente puede hacer consultas a través del sistema de registro para obtener datos, modificarlos,

guardar estos cambios o establecer nuevas tablas de registros, por ejemplo.

- **Compatibilidad con SQL:** SQL es un lenguaje generalizado dentro de la industria. Al ser un estándar MySQL ofrece plena compatibilidad por lo que si has trabajado en otro motor de bases de datos no tendrás problemas en migrar a MySQL.
- **Vistas:** Desde la versión 5.0 de MySQL se ofrece compatibilidad para poder configurar vistas personalizadas del mismo modo que podemos hacerlo en otras bases de datos SQL. En bases de datos de gran tamaño las vistas se hacen un recurso imprescindible.
- **Procedimientos almacenados.** MySQL posee la característica de no procesar las tablas directamente, sino que a través de procedimientos almacenados es posible incrementar la eficacia de nuestra implementación.
- **Desencadenantes.** MySQL permite además poder automatizar ciertas tareas dentro de nuestra base de datos. En el momento que se produce un evento otro es lanzado para actualizar registros o optimizar su funcionalidad.
- **Transacciones.** Una transacción representa la actuación de diversas operaciones en la base de datos como un dispositivo. El sistema de base de registros avala que todos los procedimientos se establezcan correctamente o ninguna de ellas. En caso por ejemplo de una falla de energía, cuando el monitor falla u ocurre algún otro inconveniente, el sistema opta por preservar la integridad de la base de datos resguardando la información.

Ventajas de usar MySQL

Descritas las principales características de MySQL es fácil ver sus ventajas. MySQL es una opción razonable para ser usado en ámbito empresarial. Al estar basado en código abierto permite a pequeñas empresas y desarrolladores disponer de una solución fiable y estandarizada para sus aplicaciones. Por ejemplo, si se cuenta con un listado de clientes, una tienda online con un catálogo de productos o incluso una gran selección de contenidos multimedia disponible, MySQL ayuda a gestionarlo todo debida y ordenadamente.

Gráfico 2.17 Herramienta MySql



Nota. (MySql, 2021)

2.18.2 Lenguaje de Programación PHP

PHP es un lenguaje de programación de código abierto muy popular, adecuado para desarrollo web y que puede ser incrustado en HTML. Es popular porque un gran número de páginas y portales web están creadas en PHP (Gonzales Gutierrez, 2020).

Fue uno de los primeros lenguajes de programación de lado del servidor que se podían incorporar directamente en un documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera el texto plano en formato UTF-8 ampliamente reconocido por el estándar HTML, dando como resultado, en los exploradores, una salida al usuario perfectamente entendible.

PHP es un lenguaje de secuencia de comandos de servidor diseñado específicamente para la Web. Dentro de una página Web puede incrustar código PHP que se ejecutara cada vez que se visite una página. El código PHP es interpretado en el servidor Web y genera código HTML y otro contenido que el visitante verá.

PHP fue concebido en 1994 y es fruto del trabajo de un hombre, Rasmus Lerdorf. Ha sido adoptado por otras personas de talent0 y ha experimentado tres transformaciones importantes hasta convertirse en el producto actual. En octubre de

2002, era utilizado por más de nueve millones de dominios de todo el mundo y su número crece rápidamente. Si desea conocer el número actual de sitios que utilizan

este lenguaje, visite el sitio <http://www.php.net/usage.php>. PHP es un producto de código abierto, lo que quiere decir que puede acceder a su código. Puede utilizarlo, modificarlo y redistribuirlo sin coste alguno.

Entre los competidores principales de PHP se puede citar a Perl, Microsoft Active Server Pages (ASP), Java Server Pages (JSP) y Allaire ColdFusion. En comparación con estos productos, PHP cuenta con muchas ventajas, entre las que se encuentran las siguientes: (Welling & Thompson, 2004)

- Alto rendimiento
- Interfaces para una gran cantidad de sistemas de base de datos diferentes.
- Bibliotecas incorporadas para muchas tareas Web habituales.
- Bajo coste.
- Facilidad de aprendizaje y uso.
- Portabilidad.
- Acceso al código abierto

El lenguaje PHP es un lenguaje de programación de estilo clásico, es decir, es un lenguaje de programación con variables, sentencias condicionales, ciclos (bucles), funciones.... No es un lenguaje de marcado como podría ser HTML, XML o WML. Está más cercano a JavaScript o a C, para aquellos que conocen estos lenguajes. Al ser PHP un lenguaje que se ejecuta en el servidor no es necesario que su navegador lo soporte, es independiente del browser, pero sin embargo para que las páginas PHP funcionen, el servidor donde están alojadas debe soportar PHP. (Estrada, 2021).

Gráfico 2.18 Herramienta php



Nota. (Estrada, 2021)

2.18.3 Bootstrap

Bootstrap es un framework CSS desarrollado por Twitter en 2010, para estandarizar las herramientas de la compañía.

Inicialmente, se llamó Twitter Blueprint y, un poco más tarde, en 2011, se transformó en código abierto y su nombre cambió para Bootstrap. Desde entonces fue actualizado varias veces y ya se encuentra en la versión 4.4.

El framework combina CSS y JavaScript para estilizar los elementos de una página HTML. Permite mucho más que, simplemente, cambiar el color de los botones y los enlaces.

Esta es una herramienta que proporciona interactividad en la página, por lo que ofrece una serie de componentes que facilitan la comunicación con el usuario, como menús de navegación, controles de página, barras de progreso y más.

Además de todas las características que ofrece el framework, su principal objetivo es permitir la construcción de sitios web responsive para dispositivos móviles. Desde que vio la luz, Bootstrap se ha caracterizado por tratarse de una excelente herramienta para crear interfaces de usuarios limpias y totalmente adaptables a cualquier tipo de dispositivo y pantalla, independientemente de su tamaño. Ya hemos dicho que Bootstrap se ha convertido en una de las herramientas más utilizadas hoy en día a la hora de realizar cualquier diseño web.

Esto es debido a algunas de sus principales características, que lo han hecho muy atractivo a los ojos de los desarrolladores. (acensTechnologies, 2016)

- Fácil e intuitivo
- Compatibles con todos los navegadores.
- Optimizado para dispositivos móviles.
- Amplia comunidad de desarrolladores tras el proyecto.

Funcionalidad de Bootstrap

Bootstrap ofrece una serie de características que se pueden implementar en un sitio web. A continuación, hablaremos de las posibilidades de esta herramienta.

- **Diseño responsive:** Una de las características principales de Bootstrap es permitir que la adaptación de la página se realice según el tipo de dispositivo utilizado. Para garantizar la responsividad, el framework funciona con: Estilización del elemento `<div>`; Uso del class container.

En la práctica, el elemento `<div>`, funciona para crear una serie de notas, similar a una tabla, capaz de estructurar la página de forma adaptable.

Ya hubo un intento de utilizar tablas para crear diseños responsivos, sin embargo, existían limitaciones para definir la longitud de las columnas lo que hizo imposible su uso en dispositivos más pequeños, como smartphones.

El elemento `<div>` es más flexible, ya que permite definir y cambiar el tamaño de la longitud fácilmente. Bootstrap le ha asignado al elemento `<div>` una característica de class container, que funciona para determinar las dimensiones apropiadas para los elementos insertados en ese espacio.

Básicamente, el framework funciona con tres tipos de containers:

- **Container:** como un conjunto con una propiedad de ancho máximo, que determina qué tamaño de tela es ideal para crear el diseño de página.
- **Container-fluid:** considera la longitud total de la tela del dispositivo para definir el diseño. Para esto, se considera la propiedad `width 100%` en todos los límites de tamaño de tela.
- **Container { breakpoint}:** considera `width 100%` hasta alcanzar un cierto tamaño.

Biblioteca de componentes: Otra interesante función de este framework es la cantidad de componentes que pueden ser usados para proporcionar una mejor

interacción y perfeccionar la comunicación con el usuario.

Enseguida mencionaremos los principales:

- **Alertas:** Bootstrap permite una configuración simple y rápida de diferentes tipos de alertas, con colores específicos, según la situación.
- **Carrusel:** Un componente ampliamente utilizado en Bootstrap es el Carrusel, una presentación de diapositivas, es decir, una herramienta que permite la visualización de imágenes de manera receptiva.
- **Barra de navegación:** Otro poderoso componente de la estructura es la NavBar (barra de navegación), que permite la construcción de un sistema de navegación sensible. Es posible configurar diferentes formas de presentar el menú, elegir entre posicionamiento lateral o superior y, también, definir una forma de visualización que se pueda extender o contraer. También es posible determinar cómo mostrar los enlaces del menú, que pueden ser en forma de botón, enlace, menú suspenso, entre otras configuraciones, para facilitar la implementación de la navegación del sitio. (acensTechnologies, 2016)

Ventajas de Bootstrap

Entre las ventajas de utilizar bootstrap podemos enumerar las siguientes:

- Es de código abierto, y todo su código actualizado se encuentra en un repositorio de Github.
- Está mantenido y actualizado por Twitter.
- Es compatible con la mayoría de navegadores (Chrome, Safari, Mozilla...).
- Dispone de gran cantidad de documentación, tanto en su portal como en páginas web especializadas.
- Utiliza componentes vitales para los desarrolladores (HTML5, CSS3, jQuery o GitHub, entre otros).
- Incluye Grid system para maquetar por columnas.
- Sus plantillas son de sencilla adaptación responsive.

- Dispone de un conjunto de elementos web personalizables.
- Se integra con librerías JavaScript.
- Usa Less: un lenguaje de las hojas de estilo CSS preparado para enriquecer los estilos de la web. (Lozada, 2020)

Gráfico 2.19 Herramienta Bootstrap



Nota. (Lozada, 2020)

2.18.4 Lenguaje de Programación JavaScript

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas de web dinámicas (Eguiluz, 2020).

El principal uso de JavaScript, está disponible principalmente del lado del Frontend5 , agregando mayor interactividad a la web, también puede usar bibliotecas y framework como: JQuery, Angular y otros. que ayudan a crear una mejor experiencia al usuario en los sitios web (Giampiere Grados Caballero, 2020).

JavaScript es un lenguaje de programación diseñado en un principio para añadir interactividad a las páginas webs y crear aplicaciones web. A pesar de la similitud en el nombre, no está relacionado con Java. Se emplea en el desarrollo de páginas web para tareas como cambiar automáticamente la fecha de una página, hacer que una página aparezca en una ventana emergente al hacer clic en un enlace o que un texto o imagen cambien al pasar el ratón por encima. También suele emplearse para hacer encuestas y formularios. Se ejecuta en el ordenador del visitante a la web, por lo que no requiere descargas constantes desde el sitio web. (Armetrics, 2021)

Uso de JavaScript

JavaScript también se utiliza en entornos que no se basan en web, tales como documentos PDF, navegadores específicos del sitio y widgets de escritorio. Máquinas virtuales de JavaScript (VMs) mucho más nuevas y rápidas, así como plataformas construidas sobre este lenguaje, han aumentado la popularidad de JavaScript para aplicaciones web. Por parte del cliente, JavaScript se ha aplicado tradicionalmente como un lenguaje interpretado, pero los navegadores más recientes pueden realizar la compilación al momento.

También se utiliza en el desarrollo de videojuegos, en la creación de aplicaciones de escritorio y móviles y en la programación de servidores con entornos de ejecución como Node.js. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del Document Object Model (DOM).

Normalmente se utilizaba JavaScript en las páginas web HTML para realizar operaciones y solo en el marco de la aplicación cliente, sin acceso a funciones del servidor. En la actualidad se emplea para enviar y recibir información del servidor junto con ayuda de otras tecnologías como AJAX. JavaScript es interpretado en el agente de usuario al mismo tiempo que las sentencias van descargándose junto con el código HTML. (McFarland, 2014).

Gráfico 2.20 Herramienta JavaScript



Nota. (Armetrics, 2021)

2.18.5 jQuery

jQuery es una librería o biblioteca de JavaScript que facilita la programación en este lenguaje, al ofrecer una serie de funciones y métodos con los cuales los usuarios pueden producir páginas web o sitios web más rápido y fácil.

JavaScript tiene un pequeño secreto vergonzoso: escribirlo puede ser difícil. Si bien es más simple que muchos otros lenguajes de programación, JavaScript sigue siendo un lenguaje de programación. Y muchas personas, incluidos los diseñadores web, encuentran difícil programarlo.

Para complicar aún más las cosas, los diferentes navegadores web entienden JavaScript de manera diferente, por lo que un programa que funciona en, digamos, Chrome puede no responder por completo en Internet Explorer 9. Esta situación común puede costar muchas horas de prueba en diferentes máquinas y diferentes navegadores para asegurarse de que un programa funcione correctamente para toda la audiencia de su sitio.

Aquí es donde entra jQuery. jQuery es una librería de JavaScript destinada a hacer la programación JavaScript más fácil y divertida. Una librería de JavaScript es un complejo conjunto de código JavaScript que simplifica las tareas difíciles y resuelve los problemas de compatibilidad en navegadores. En otras palabras, jQuery resuelve los dos mayores problemas de JavaScript: complejidad y la naturaleza meticulosa de los diferentes navegadores web.

jQuery es el arma secreta de un diseñador web en la batalla de la programación JavaScript. Con jQuery, puede realizar tareas en una sola línea de código que podrían requerir cientos de líneas de programación y muchas horas de pruebas en el navegador para lograrlo con su propio código JavaScript. (Porto & Gardey, Definición, 2019)

Características de jQuery

A continuación, se presentan las principales características de JQuery:

- Es un software libre, por lo que puede ser empleado por cualquier usuario de manera gratuita.
- Su librería permite actualizaciones constantes y rápidas.
- Posee un código abierto y compatible con diferentes navegadores.
- Es fácil de usar, por lo que permite ahorrar tiempo y esfuerzo.
- Su desempeño se integra muy bien con AJAX, una técnica de desarrollo web.
- Permite realizar animaciones, efectos y personalizaciones.
- Es compatible con diferentes buscadores como Google Chrome, Microsoft Edge, Firefox, IE, Safari, Android e IOS, cuyas páginas web se deben programar en formas diferentes. (Parada, 2019)

Ventajas y desventajas de jQuery

- JQuery es ampliamente utilizado por los desarrolladores de sitios web porque ofrece una serie de ventajas que facilitan su trabajo y ahorran tiempo y esfuerzo que pueden destinar a otros proyectos.
- Asimismo, JQuery es de uso práctico, es compatible con diversos buscadores, los bugs o errores de software son resueltos con rapidez, entre otros. Cabe mencionar que también se puede hacer uso de JQuery en los dispositivos móviles.
- Incluye, listos para usar, elementos como botones, barras de herramientas, formularios, listas y tipografías especialmente optimizadas y estilizadas para aplicaciones móviles. Además, cuenta con un sistema de bloques ideal para construir plantillas.
- Nos permite construir aplicaciones aptas para todo tipo de plataformas casi sin esfuerzo, evitándonos preocupaciones por problemas de compatibilidad.
- Es muy fácil de aprender y utilizar. Si se tienen adecuados conocimientos de

HTML y CSS, se pueden construir aplicaciones robustas en muy poco tiempo. El sitio web de jQuery cuenta con abundante documentación para aprender.

- Tiene soporte para Ajax, una tecnología para crear aplicaciones dinámicas que se ejecuten del lado cliente.
- No requiere del uso de programas especiales, a diferencia de otras tecnologías, como Adobe Flash, que exigen una aplicación para desarrollar sobre ellas

Desventajas

- Las funciones que ofrece son muchas, pero resultan difíciles de personalizar.
- Su aspecto visual es estandarizado y no se integra con el de la plataforma.
- Como es necesario invocar a un archivo para utilizar sus funciones, ralentiza levemente la carga de la página.
- Su manejo de CSS suele resultar innecesariamente complejo. A veces cuesta saber qué clases utilizar.
- No existen muchas plantillas prediseñadas sobre las cuales empezar a construir nuestra aplicación. (Parada, 2019)

Gráfico 2.21 Herramienta JQuery



Nota. (Parada, 2019)

**“DISEÑO E IMPLEMENTACIÓN DE UNA BIBLIOTECA
VIRTUAL Y REPOSITORIO DE PROYECTOS DE
GRADO Y TESIS”**

CAPÍTULO III

MARCO

2023

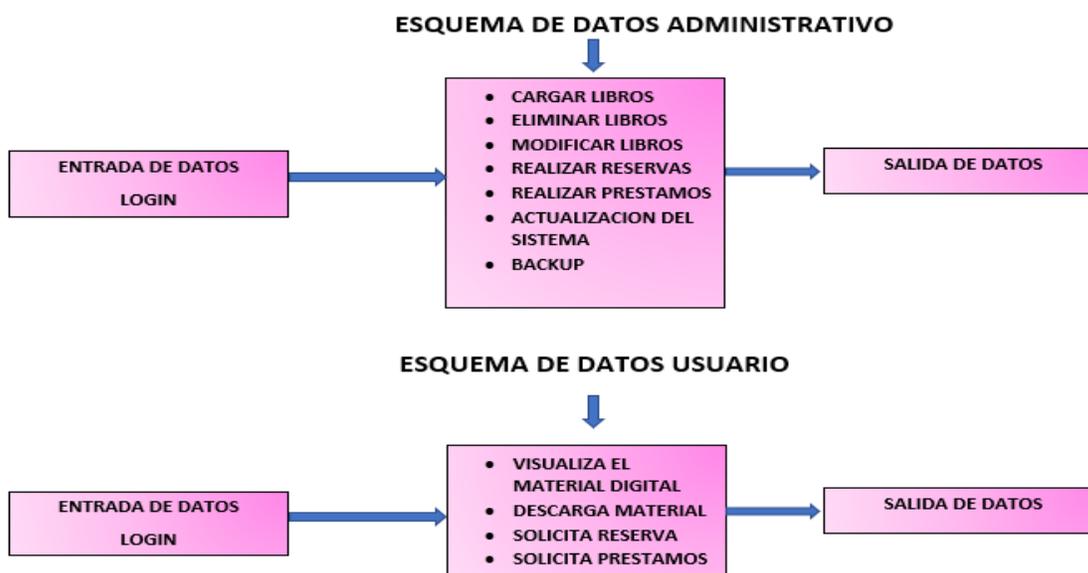
APLICATIVO



3.1 INTRODUCCIÓN

En esta etapa se realizará el análisis y diseño, siguiendo las fases de la metodología UWE (UML Based *Web Engineering – Ingeniería Web Basado en UML*). Las fases de esta metodología son: análisis de requisitos, diseño conceptual, diseño navegacional y diseño de presentación; nos presentan diversos esquemas en un proceso iterativo e incremental dando apoyo al modelado de la aplicación.

Gráfico 3.1 Esquema del Sistema



Nota. Elaboración Propia

3.2 ANÁLISIS DEL SISTEMA ACTUAL

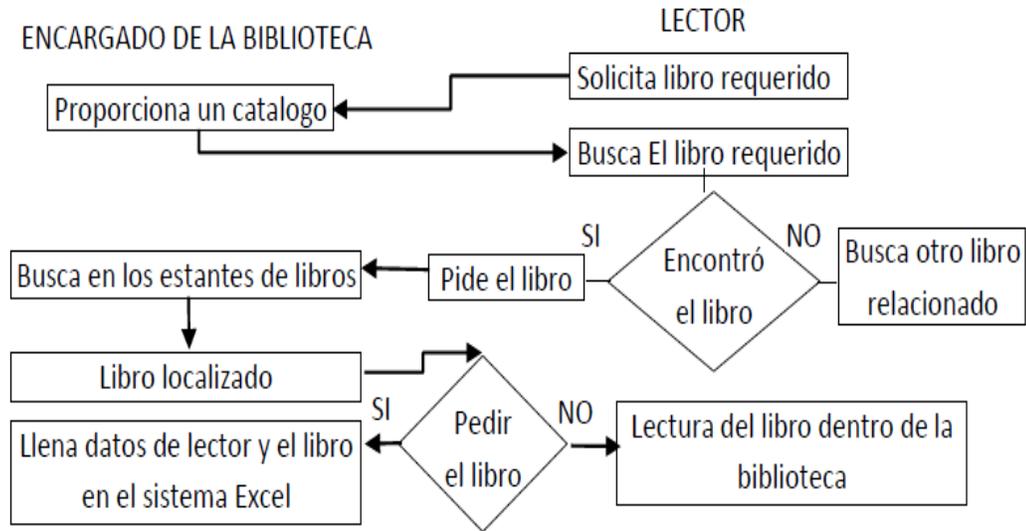
Para que el sistema pueda cubrir las diferentes necesidades en los procesos que se realizan en las diferentes bibliotecas, es necesario identificar a los principales participantes que interactúan con el sistema.

3.2.1 Objetivo

El objetivo principal de la biblioteca de Escuela Superior de Formación de Maestros Tecnológico Humanístico El Alto es proporcionar el apoyo bibliográfico a estudiantes y docentes, apoyando al proceso de formación de los futuros profesionales en procura de la excelencia académica.

3.2.2 Proceso actual

Gráfico 3.2 Proceso Actual de Préstamos de Libros



Nota. (Elaboración propia)

3.3 OBTENCIÓN DE REQUISITOS

La tarea de la ingeniería de requisitos es fundamental para que un sistema sea exitoso, utilizando las herramientas de ingeniería de requerimientos, para lo cual se realizó cuestionarios y entrevistas al personal encargado de la Biblioteca.

3.3.1 Actores implicados en la biblioteca

A) Administración	1. Responsable de la Biblioteca (responsable o administrador). 2. Técnico Bibliotecario (Administrador de apoyo). 3. Docente (Apoyo).
B) Usuario	1. Lector (Estudiantes, docentes y otros).

3.3.2 Definición de Actores

Para la integración y centralización de la información de las diferentes unidades bibliotecarias con las que cuenta la Institución ESCUELA SUPERIOR DE FORMACIÓN DE MAESTRAS Y MAESTROS TECNOLÓGICO HUMANÍSTICO EL ALTO ESFMTHEA se consideró la adición de un nuevo participante denominado ADMINISTRADOR, a continuación, mencionamos algunas tareas que realiza.

Tabla 3.1 Lista de Actores

DESCRIPCION	RESPONSABILIDADES
BIBLIOTECARIO (RESPONSABLE)	<ul style="list-style-type: none">• Dirige y administra un espacio bibliotecario.• Realiza el control y registros de los documentos bibliográficos.• Encargado de generar los respectivos códigos de catalogación (Signatura) para los nuevos registros bibliográficos.• Realiza los diferentes procesos de registros acerca de préstamos y sanciones.
USUARIO	<ul style="list-style-type: none">• Realiza las diferentes peticiones de préstamo de un documento bibliográfico a una determinada biblioteca.
ADMINISTRADOR	<ul style="list-style-type: none">• Administra los diferentes espacios bibliotecarios creados en el sistema.• Administra los diferentes usuarios asignados a las diferentes bibliotecas.• Realiza cambios administrativos, es decir realiza el cambio de un usuario de una biblioteca a otra.

Nota. (Elaboración propia)

3.3.3 Lista de Requerimientos

Un requerimiento se visualiza como una declaración abstracta de alto nivel de un servicio que debe proveer el sistema o como una restricción de este.

Requerimientos Funcionales

Los requerimientos funcionales muestran las características mínimas que necesita el sistema para cubrir los diferentes procesos, la cual debe cubrir la mayoría de las necesidades que un usuario requiere.

Tabla 3.2 Lista de Requerimientos Funcionales

COD	FUNCION	DESCRIPCION
R1	Administración de información de unidades bibliotecarias.	Debe permitir registrar, modificar y listar los datos de una unidad bibliotecaria perteneciente a una carrera o área de la ESCUELA SUPERIOR DE FORMACIÓN DE MAESTRAS Y MAESTROS TECNOLÓGICO HUMANÍSTICO EL ALTO ESFMTHEA.
R2	Administración de usuarios.	Debe permitir registrar, modificar y listar los datos de los usuarios asignados a una determinada biblioteca independientemente, para luego asignarles los respectivos roles y permisos de acceso al sistema.
R3	Registro de tipos (GRUPOS) de documentos bibliográficos: Libros, Proyectos de Grado, Tesis de Grado, Monografías y otros.	Debe permitir el registro de los diversos tipos de documentos bibliográficos indicando o seleccionando sus respectivos parámetros de contenido: Título, Autor, Código, Edición, Año y otros.
R4	Registro de documentos bibliográficos.	Debe permitir el registro y la modificación de los documentos bibliográficos indicando a su vez el Tipo de Documento al que pertenece.
R5	Vincular documentos digitales (PDF).	El sistema debe permitir vincular (Subir) un documento digital (PDF) a uno o varios ejemplares, para que un lector posea otra alternativa para la consulta de un documento bibliográfico de su

		interés.
R6	Registro y control de préstamos de documentos bibliográficos.	El sistema debe presentar un módulo que permita la administración de los procesos de préstamos, módulo que deberá registrar un préstamo y los datos personales del lector para luego generar un ticket de préstamo, de forma opcional.
R7	Administración de solicitudes de préstamos.	El sistema debe presentar el espacio de recepción de solicitudes de préstamos de documentos bibliográficos realizados por los diferentes lectores, para su posterior confirmación de préstamo.
R8	Registro de observaciones o préstamos de documentos bibliográficos inconclusos.	El sistema debe presentar un formulario donde se pueda registrar las observaciones de un préstamo no concluido.
R9	Notificaciones de solicitud de préstamos de documentos bibliotecarios.	El sistema debe emitir una notificación por una nueva solicitud de préstamo que ingrese al sistema.
R10	Interfaz de solicitud de préstamos para los diferentes lectores.	El sistema debe ofrecer una interfaz dedicada para el lector, donde pueda realizar búsquedas minuciosas acerca de un documento bibliográfico de su interés para su posterior envío de solicitud de préstamo.
R11	Reportes.	Es importante documentar los diferentes procesos de préstamos y registros existentes de los diferentes documentos bibliográficos por medio de reportes. El sistema deberá generar reportes acerca de los módulos de préstamos, observaciones y documentos bibliográficos existentes o activos para préstamos así también como los ya descartados.

Nota. (Elaboración Propia)

3.3.4 Requerimientos No Funcionales

Tabla 3.3 Lista de Requerimientos No Funcionales

COD	DESCRIPCION
R1	El sistema debe presentar una interfaz responsiva para poder adaptarse a los diferentes dispositivos que hoy en día poseemos.
R2	El sistema al poseer varios módulos, debe contar con un Control de Acceso, para los diferentes tipos de usuarios, con la finalidad de proteger algunos módulos de uso administrativo exclusivo.
R3	El sistema debe presentar un registro de historial de préstamos independiente para cada ejemplar.
R4	El soporte y mantenimiento periódico del sistema asegurará un buen rendimiento al sistema.

Nota. (Elaboración propia)

3.3.5 Analisis de Requerimientos

Con la ayuda de los diagramas de casos de uso, se irán describiendo el comportamiento del sistema frente a las acciones de los diferentes actores mencionados anteriormente.

En esta etapa se identificarán los diferentes actores que trabajan e interactúan con el sistema, las cuales permitirán definir los diferentes límites que posee el sistema.

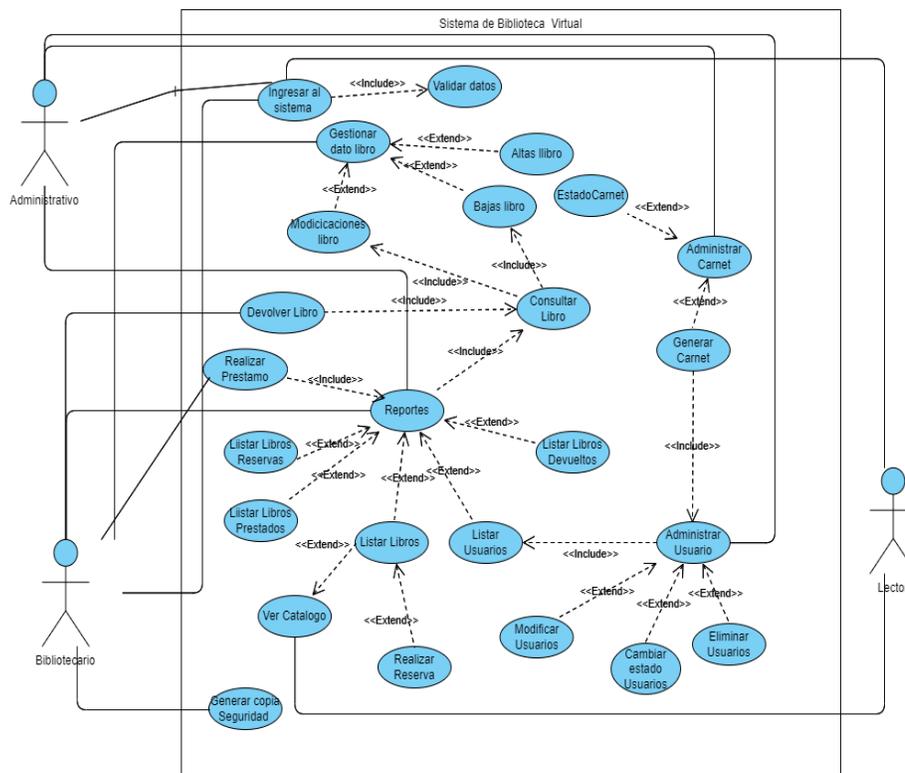
- Administrador, es el usuario principal de toda la estructura del sistema, este se encarga de crear y registrar los diferentes tipos de bibliotecas, las cuales pueden pertenecer a Carreras o Áreas. Realiza la configuración mínima para que una determinada biblioteca pueda tener acceso al uso del sistema. Cabe mencionar que los permisos de acceso solo se limitan a la configuración y administración de usuarios y permisos de las diferentes bibliotecas.

- Bibliotecario, es el segundo usuario principal, pero a nivel de una determinada biblioteca, goza de todos los permisos necesarios para poder administrar los diferentes procesos o módulos que el sistema presenta: préstamos, registro de libros, creación de nuevos usuarios y otros.
- Lector, representa a la población universitaria, las actividades que realizan son él envió de solicitudes de préstamos de uno o varios libros u otro tipo de documento bibliográfico con el fin de extraer información para incrementar su conocimiento.

3.3.6 Caso de Uso General de la Biblioteca

Aplicando los diagramas de casos de uso, se representarán e identificarán los diferentes procesos y actividades que realizan los diferentes espacios bibliotecarios y como los actores interactúan con el sistema.

Gráfico 3.3 Diagrama de Casos de Uso General de la Biblioteca



Nota. (Elaboración propia)

Tabla 3.4 Caso de Uso General

Nombre		Sistema de Biblioteca Virtual	
Autor	Gladys Orihuela Mamani		
Fecha	04/2023		
Descripción: Muestra de manera general los procesos que realiza el sistema, administración de usuarios, genera carnet, reportes, generar gráficos estadísticos, administra datos del libro, así también realiza préstamos, devoluciones y reservas.			
Actores: Administrativos, bibliotecario y lector.			
Precondiciones: El Usuario debe contar con usuario y clave asignados El usuario debe loguearse correctamente en el sistema web.			
Flujo Normal:			
Usuario	Sistema		
Carga la página del sistema	El sistema despliega ventana de ingreso		
Ingresar usuario y clave	Valida los datos. Despliega Bondades Asignadas al usuario.		
Ingresar a la opción Administrar Usuarios. Ingresar los datos.	Despliega las opciones con los campos para introducir datos del usuario. Registra, modifica, elimina y activa registros.		
Ingresar a la opción Roles Selecciona el registro de usuario.	Despliega lista actualizada. El sistema despliega lista de usuarios. Permite asignar rol, permiso. Permite activar.		
Ingresar a la opción Generar Carnet.	Despliega la lista de usuarios a generar carnet.		
Ingresar a la opción Nuevo, modificar, eliminar o ver carnet.	El sistema despliega la pantalla según la opción seleccionada y realiza las actualizaciones y registros en la base de datos.		
Ingresar a la opción administrar datos del libro. Selecciona la opción	El sistema despliega las opciones registro, modificación y eliminación. El sistema registra y actualiza la base de datos		
Ingresar a la opción Generar reportes	El sistema despliega las opciones usuarios, libros, préstamos y devoluciones e imágenes estadísticas.		
Ingresar a la opción realizar préstamo de libro.	El sistema permite seleccionar el libro, genera la fecha de préstamo y permite		

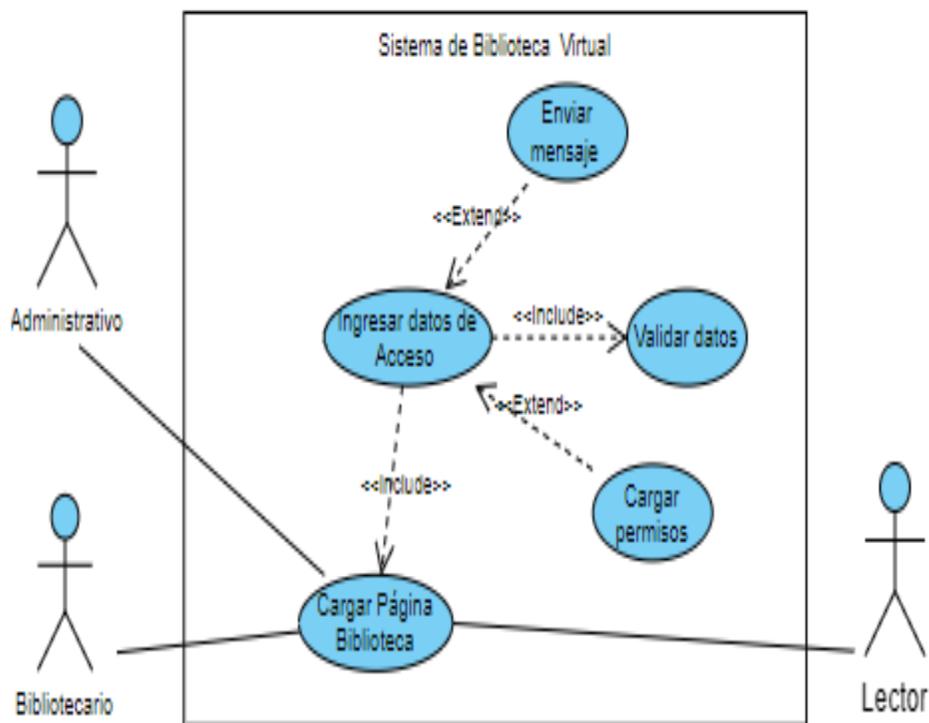
Ingresar a la opción realizar devolución.	introducir la fecha de devolución. Actualiza la base de datos El sistema verifica la fecha de devolución y libera al lector. Actualiza la base de datos.
Ingresar a hacer la reserva del libro	Verifica disponibilidad del libro. Asigna el libro en calidad de reserva, registra la fecha de solicitud de reserva y la fecha en que caduca.
Ver catálogo de libros	Despliega opciones por categoría de libros.

Nota. Elaboración Propia

3.3.7 Caso de Uso – Ingresar al Sistema

En el caso de uso del módulo de inicio de sesión se puede observar el siguiente diagrama, misma que detalla los procedimientos que el sistema debe seguir para verificar las credenciales de un determinado usuario.

Gráfico 3.4 Caso de Uso Ingresar al Sistema



Nota. Elaboración Propia

Tabla 3.5 Caso de Uso Ingresar al Sistema

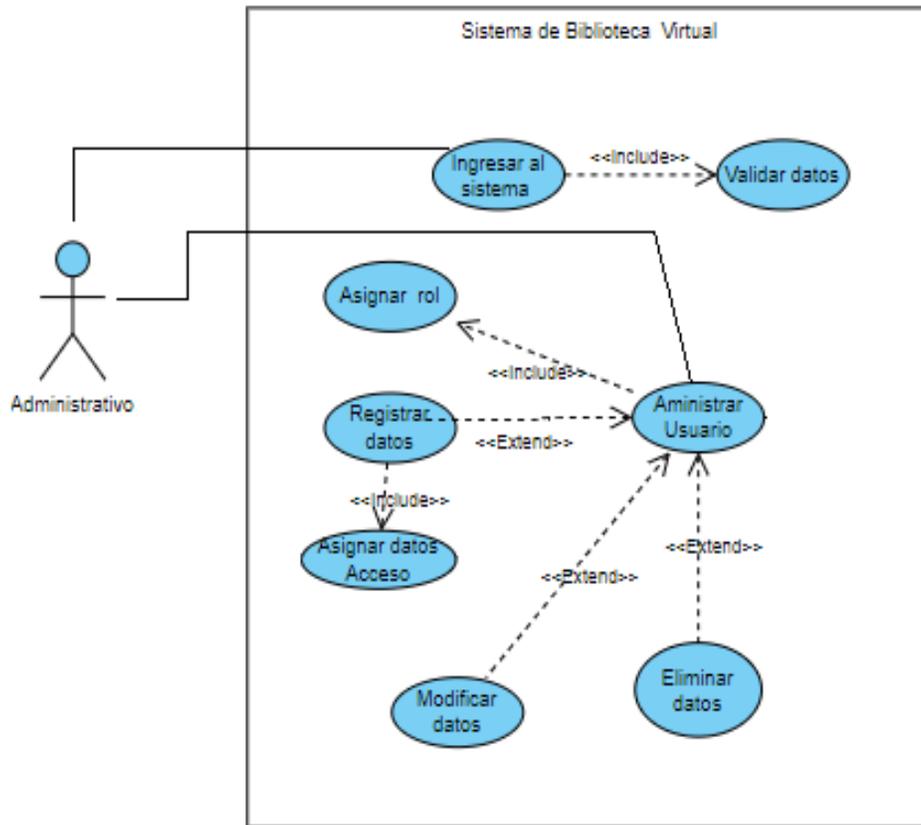
Nombre	Ingresar al Sistema
Autor	Gladys Orihuela Mamani
Fecha	04/2023
Descripción: Permite validar el acceso al sistema.	
Actores: Usuarios, base de datos	
Precondiciones: El Usuario debe contar con usuario y clave asignados El usuario debe loguearse correctamente en el sistema web	
Flujo Normal:	
Usuario	Sistema
Carga la página del sistema	El sistema despliega ventana de ingreso
Ingresar usuario y clave	Valida los datos. Despliega rol y funciones asignados al usuario.
Ingresar a la opción modificar clave	Despliega los campos para introducir clave actual y nueva clave. Verifica datos y actualiza el registro.
Ingresar la nueva clave	
Post condición	Si los datos de ingreso son incorrectos, el sistema permitirá al usuario introducir nuevamente los datos hasta que sean los correctos, mientras no podrá navegar en el sistema.

Nota. Elaboración Propia

3.3.8 Caso de Uso – Administrar Usuario

La administración de los diferentes usuarios por parte de un Bibliotecario presenta las siguientes acciones.

Gráfico 3.5 Administrar Usuario



Nota. Elaboración Propia

Tabla 3.6 Caso de Uso – Administrar Usuario

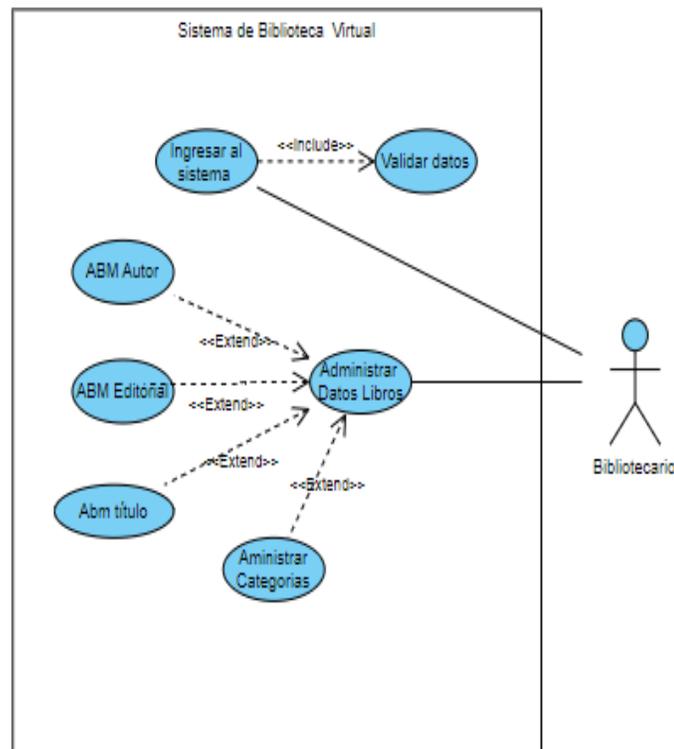
Nombre	Administrar Usuario
Autor	Gladys Orihuela Mamani
Fecha	04/2023
Descripción:	Permite administrar datos de los diferentes usuarios, ya sea crear un nuevo usuario, modificar sus datos, activar o desactivar al usuario para brindarle bondades dentro del sistema y eliminar datos.
Actores:	Administrativos, lector y bibliotecario
Precondiciones:	El usuario debe estar debidamente logueado en el sistema. Debe tener permisos para modificar registros y crear.
Flujo Normal:	
Usuario	Sistema
Carga la página del sistema	El sistema despliega ventana de ingreso

Ingresar usuario y clave	Valida los datos. Despliega Bondades Asignadas al usuario.
Ingresar a la opción administrar usuario. Presiona sobre la opción Nuevo Usuario. Asigna e rol al usuario Introduce los datos y presiona guardar. Selecciona la opción editar. Introduce los nuevos datos y presiona guardar. Selecciona activar	Lista registros de usuarios previamente creados. Despliega campos a ser llenados: carnet de identidad, nombre, apellidos, sexo. Asigna fecha de creación. El sistema registra y actualiza la base de datos. Despliega los datos del usuario. Actualiza la base de datos. Cambia de estado al usuario.

Nota. Elaboración Propia

3.3.9 Caso de Uso – Administrar datos Libro

Gráfico 3.6 Caso de Uso Administrar datos libro



Nota. Elaboración Propia

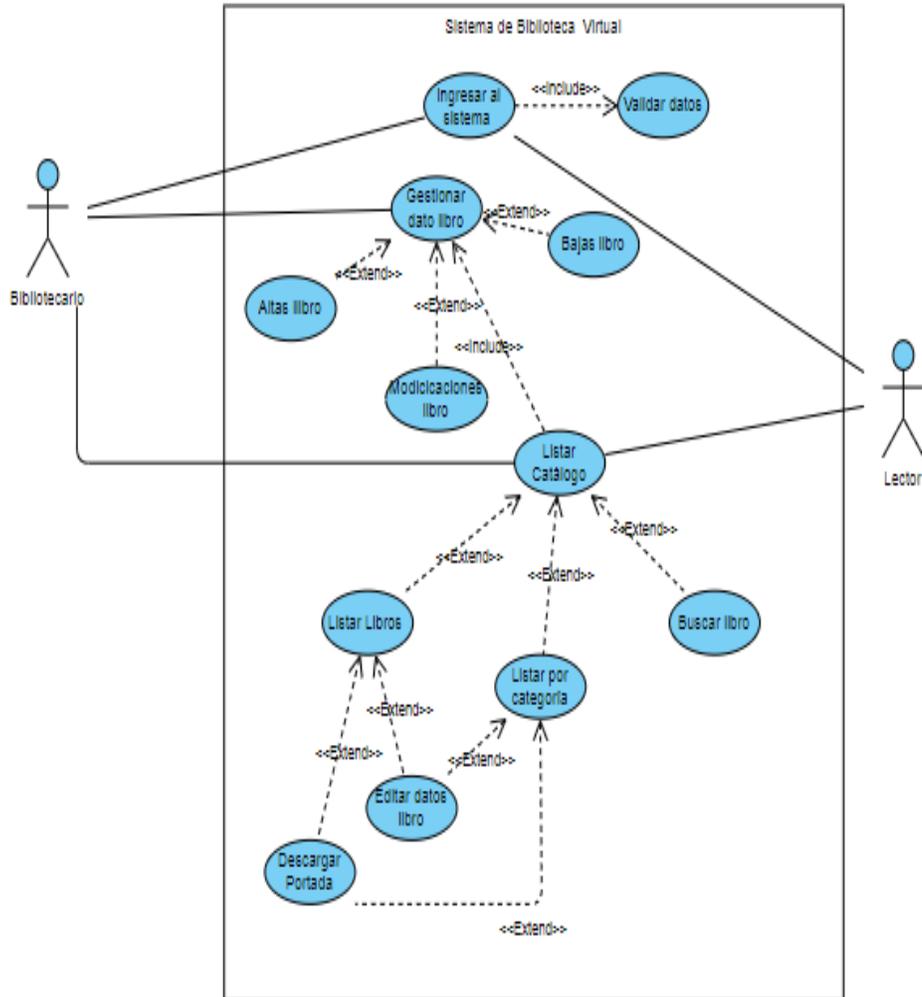
Tabla 3.7 Caso de Uso Administrar Datos de Libro

Nombre		Administrar Datos libro	
Autor	Gladys Orihuela Mamani		
Fecha	04/2023		
Descripción: Realiza un ABM (Altas, bajas y modificaciones) en el registro del libro según la categoría a la que corresponde.			
Actores: Bibliotecario			
Precondiciones: El usuario debe estar debidamente logueado en el sistema. Debe tener permisos para modificar registros y crear.			
Flujo Normal:			
Usuario	Sistema		
Carga la página del sistema	El sistema despliega ventana de ingreso		
Ingresa usuario y clave	Valida los datos. Despliega Bondades Asignadas al usuario.		
Ingresa a la opción administrar datos, autor, editorial o libro. Selecciona y presiona sobre la opción Nuevo (autor, editorial o libro). Introduce los datos. Asigna la categoría	Lista registros previamente creados. Despliega campos a ser llenados. Selecciona la categoría a la cual pertenece el libro. El sistema registra y actualiza la base de datos. Despliega la lista de registros, permite buscar. Actualiza la base de datos.		
Selecciona la opción editar. Introduce los nuevos datos y presiona guardar.	El sistema registra y actualiza la base de datos. Despliega la lista de registros, permite buscar.		
Selecciona la opción eliminar Selecciona el registro a eliminar	Actualiza la base de datos.		

Nota. Elaboración Propia

3.3.10 Caso de Uso – Listar Catalogo

Gráfico 3.7 Caso de Uso Listar Catalogo



Nota. Elaboración Propia

Tabla 3.8 Caso de Uso Listar Catálogo

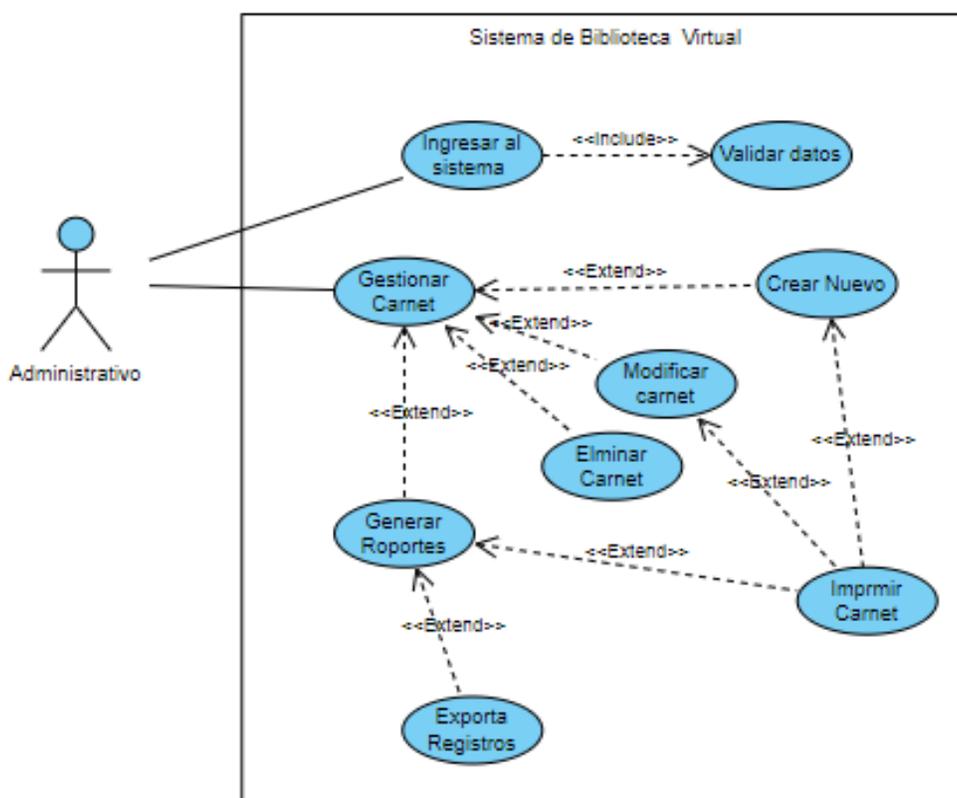
Nombre	Listar catálogo.
Autor	Gladys Orihuela Mamani
Fecha	04/2023
Descripción:	Despliega el catálogo de libros que se encuentran en la base de datos.
Actores:	Bibliotecario y el lector.
Precondiciones:	El Usuario debe contar con usuario y clave asignados El usuario debe loguearse correctamente en el sistema web.

Deben existir registros de libros.	
Flujo Normal:	
Usuario	Sistema
Carga la página del sistema	El sistema despliega ventana de ingreso
Ingresar usuario y clave	Valida los datos. Despliega rol y funciones del usuario.
Ingresar a la opción catálogo	Despliega las categorías registradas de los libros.
Ingresar a la categoría	Despliega la lista de libros de esta categoría. Habilita las opciones de buscar el libro por título, autor, editorial, imprimir portada, modificar y ver datos del libro.
Selecciona la acción.	Ejecuta la solicitud. Actualiza datos en la base de datos.

Nota. Elaboración Propia

3.3.11 Caso de Uso: Gestionar Carnet

Gráfico 3.8 Caso de Usso Gestionar Carnet



Nota. Elaboración Propia

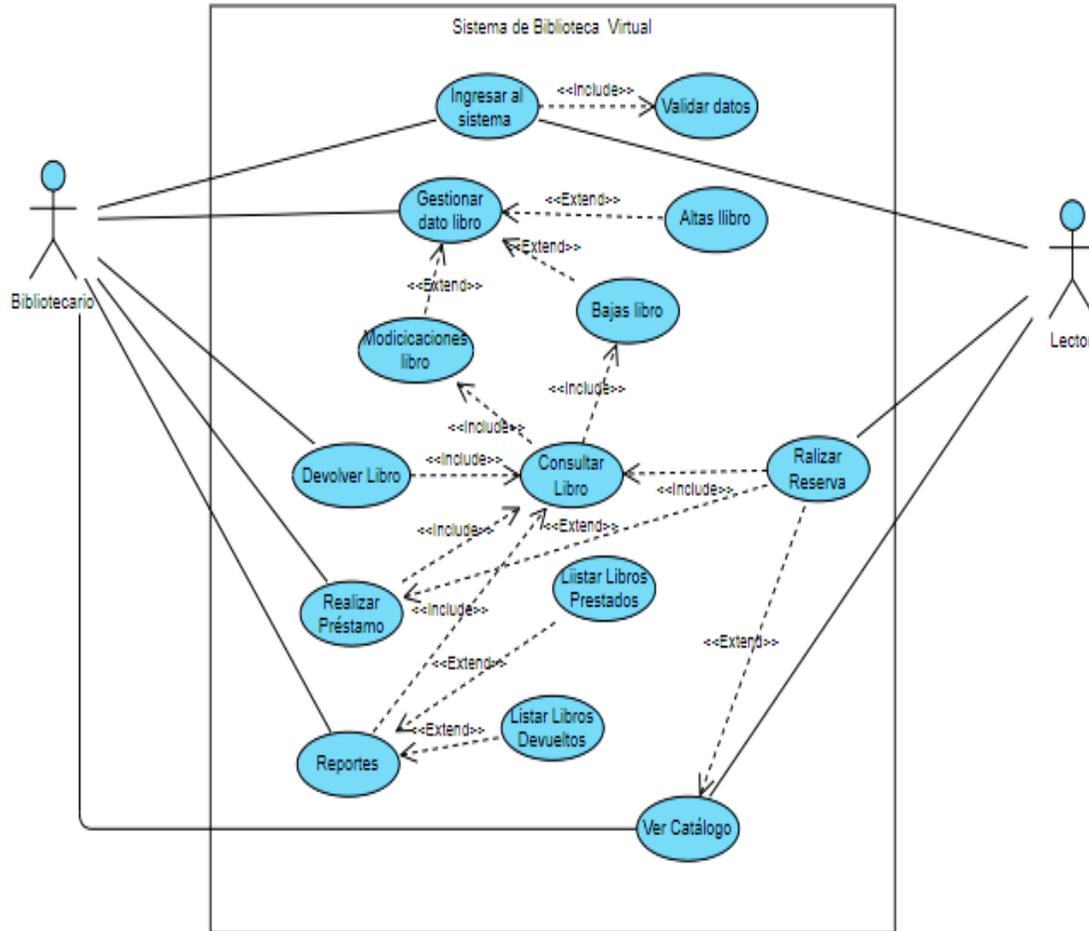
Tabla 3.9 Caso de Uso Gestionar Carnet

Nombre	Gestionar Carnet	
Autor	Gladys Orihuela Mamani	
Fecha	04/2023	
Descripción: Realiza la administración de los datos del carnet para el lector.		
Actores: Administrativo.		
Precondiciones: El Usuario debe contar con usuario y clave asignados El usuario debe loguearse correctamente en el sistema web.		
Flujo Normal:		
Usuario	Sistema	
Carga la página del sistema	El sistema despliega ventana de ingreso	
Ingresa usuario y clave	Valida los datos. Despliega rol y funciones del usuario.	
Ingresa a la carnet Selecciona la opción nuevo carnet. Ingresa los datos a ser ingresados y asignados. Presiona guardar.	Despliega la lista de estudiantes registrados. Muestra la opción de ver datos, editar datos, imprimir la lista, exportar a Excel o pdf. Se despliega la pantalla con los campos a ser llenados. Registra los datos. Actualiza la Base de datos. Permite Imprimir.	

Nota. Elaboración Propia

3.3.12 Caso de Uso: Gestionar Préstamo

Gráfico 3.9 Caso de Uso Gestionar Prestamos



Nota. Elaboración Propia

Tabla 3.10 Caso de Uso Listar Gestionar Préstamo

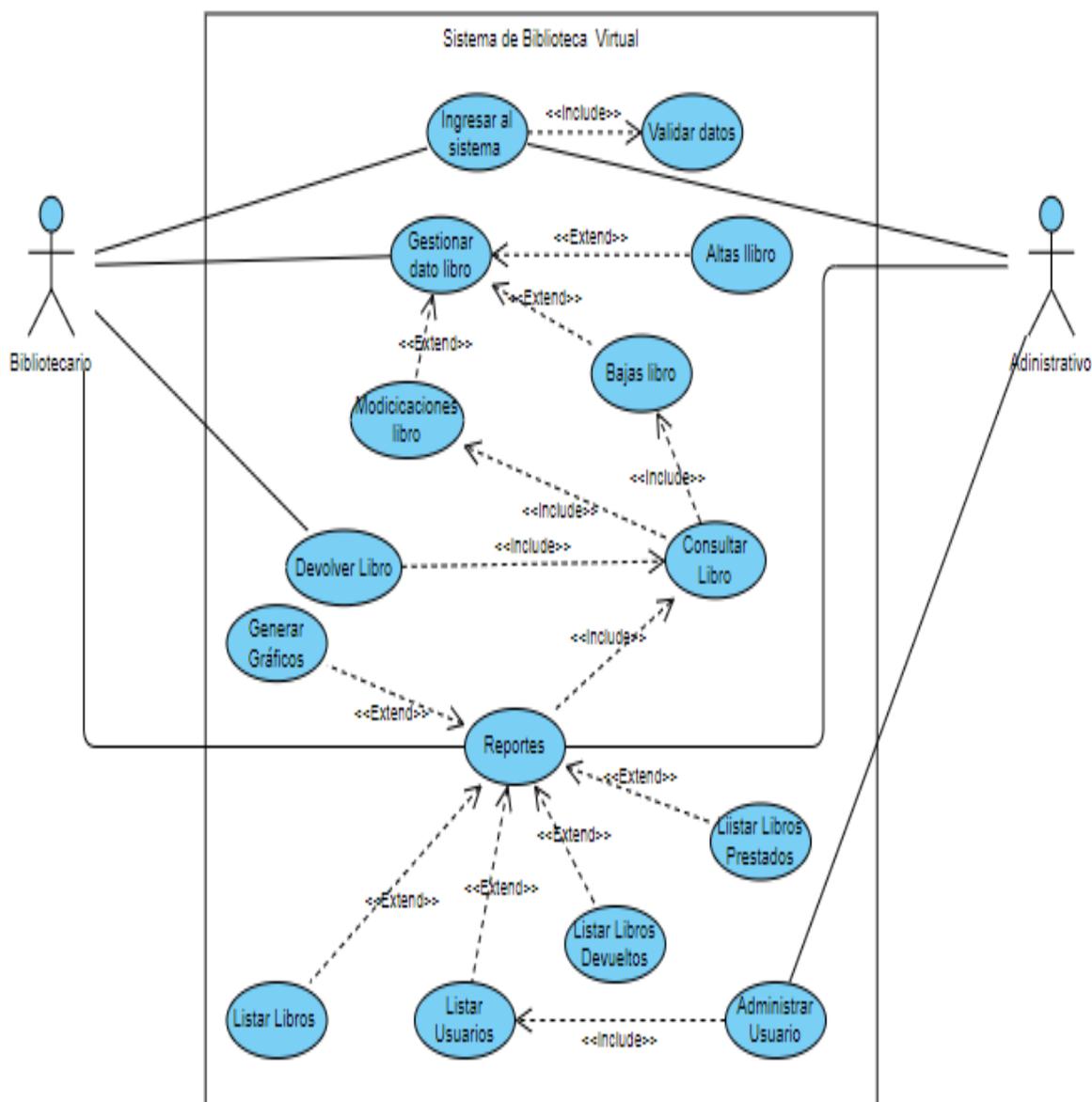
Nombre	Gestionar Préstamo
Autor	Gladys Orihuela Mamani
Fecha	04/2023
Descripción:	Permite realizar el préstamo de libros, según la disponibilidad del libro, puede realizarse previamente una reserva si lo hace un estudiante o lector, como así también puede realizarse el préstamo directamente desde el rol de bibliotecario.
Actores:	

Bibliotecario y lector.	
Precondiciones: El Usuario debe contar con usuario y clave asignados El usuario debe loguearse correctamente en el sistema web. El libro debe estar disponible. El lector debe estar registrado en el sistema.	
Flujo Normal:	
Usuario	Sistema
Carga la página del sistema	El sistema despliega ventana de ingreso
Ingresa usuario y clave	Valida los datos. Despliega rol y funciones del usuario.
El lector solicita reservar libro. El lector hace la reserva del libro. El bibliotecario realiza el préstamo del libro. Si existe reserva ingresa a la opción de libros en reserva. Edita el registro reservado.	Despliega la lista del catálogo por categorías. Muestra la disponibilidad del libro. Registra al lector, libro, fecha de registro de reserva y el tiempo en el que concluye. Cambia el estado del registro del libro a reservado. Muestra la lista de registros de los libros reservados. Muestra los datos de la reserva, registra fecha de préstamo y fecha límite de entrega. Actualiza el estado del libro de reservado a prestado. Actualiza los movimientos de datos.
Realiza préstamo sin reserva. Ingresa a la opción préstamo Ingresa al registro del libro.	Despliega la lista del catálogo por categorías. Muestra la disponibilidad del libro. Registra al lector, registra fecha de préstamo y fecha límite de entrega. Actualiza el estado del libro de disponible a prestado. Actualiza los movimientos de datos.

Nota. Elaboración Propia

3.3.13 Caso de Uso: Reportes

Gráfico 3.10 Caso de Uso Reportes



Nota. Elaboración Propia

Tabla 3.11 Caso de Uso Reportes

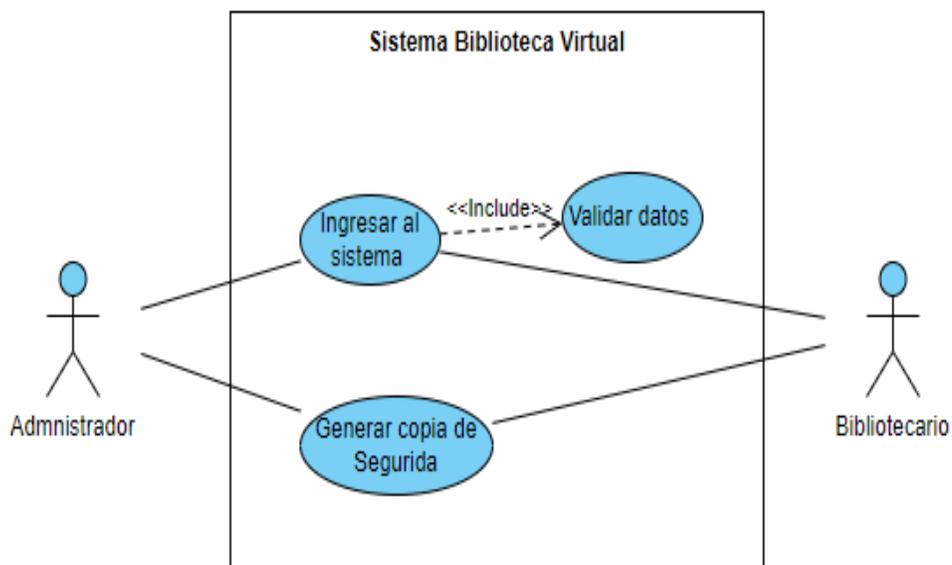
Nombre	Sistema de Biblioteca Virtual
Autor	Gladys Orihuela Mamani

Fecha	04/2023
Descripción:	Permite generar reportes del sistema en lista y gráficamente.
Actores:	Administrativos y bibliotecarios.
Precondiciones:	El Usuario debe contar con usuario y clave asignados El usuario debe loguearse correctamente en el sistema web.
Flujo Normal:	
Usuario	Sistema
Carga la página del sistema	El sistema despliega ventana de ingreso
Ingresa usuario y clave	Valida los datos. Despliega rol y función asignadas al usuario.
Ingresa a la opción generar reporte.	Despliega las opciones actualizadas de reportes de: usuarios, libros prestados, libros devueltos y reservas.
	Permite exportar a Excel, pdf e imprimir los reportes.

Nota. Elaboración Propia

3.3.14 Caso de Uso: Generar Copia de Seguridad

Gráfico 3.11 Caso de Uso Copia de Seguridad



Nota. Elaboración Propia

Tabla 3.12 Caso de Uso Copia de Seguridad

Nombre		Generar Copia de Seguridad
Autor	Gladys Orihuela Mamani	
Fecha	04/2023	
Descripción: Permite descargar copia de seguridad de los datos almacenados.		
Actores: Administrativo.		
Precondiciones: El Usuario debe contar con usuario y clave asignados El usuario debe loguearse correctamente en el sistema web.		
Flujo Normal:		
Usuario	Sistema	
Carga la página del sistema	El sistema despliega ventana de ingreso	
Ingresa usuario y clave	Valida los datos. Despliega rol y funciones del usuario.	
Ingresa a la opción copia de seguridad Presiona sobre el botón generar respaldo.	El sistema despliega pantalla y botón para generar copia. Descarga un respaldo de la Base de Datos.	

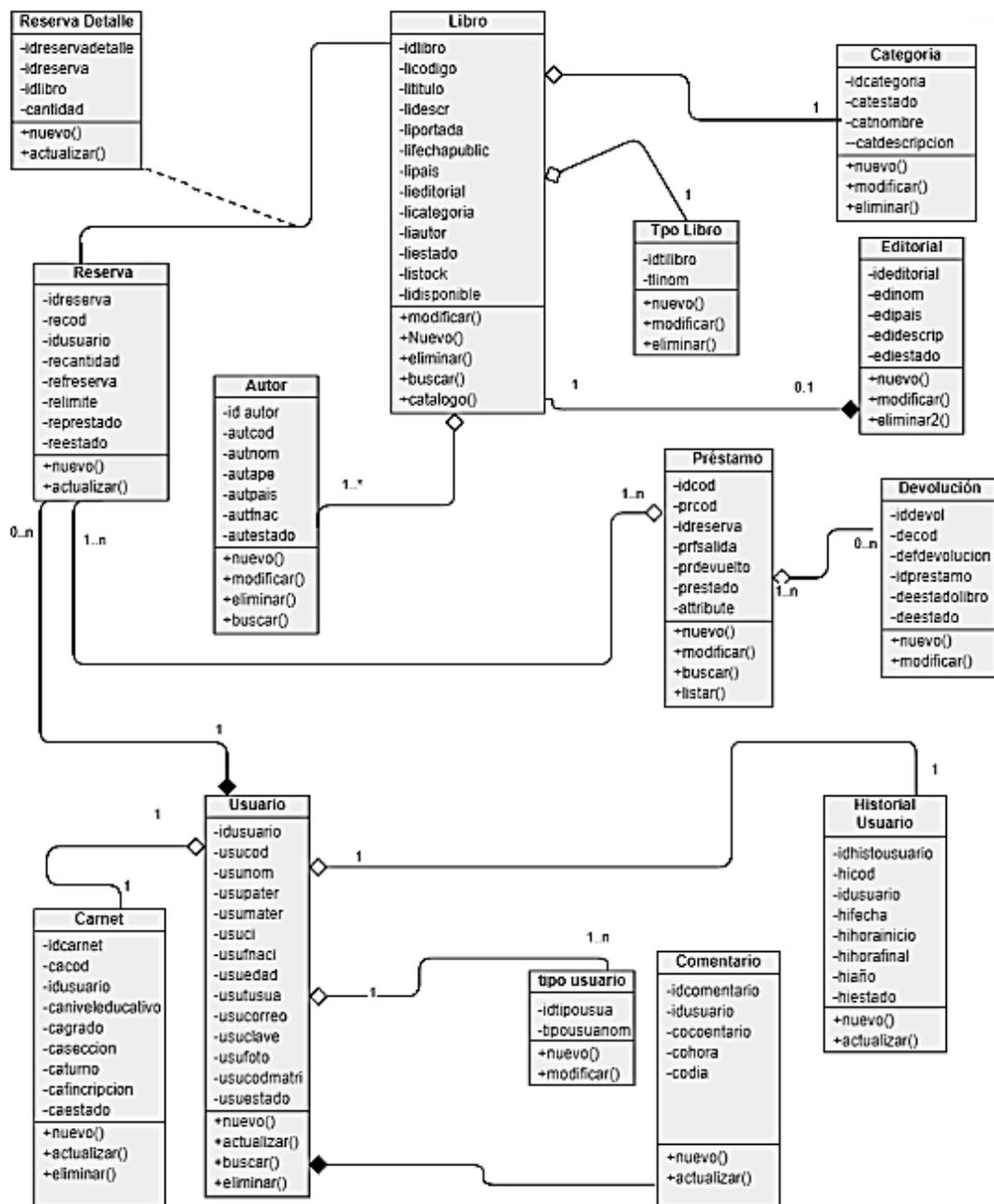
Nota. Elaboración Propia

3.4 MODELO CONTENIDO

El diagrama de contenido tiene por propósito mostrar las relaciones entre las entidades y la estructura de los datos que se encuentran alojados en el sistema el modelo de contenido contiene la información relevante almacenada en el sistema como se estructura como se relaciona. Esto se representa mediante un diagrama de clases UML.

3.4.1 Diagrama de Clases

Gráfico 3.12 Diagrama de Clase



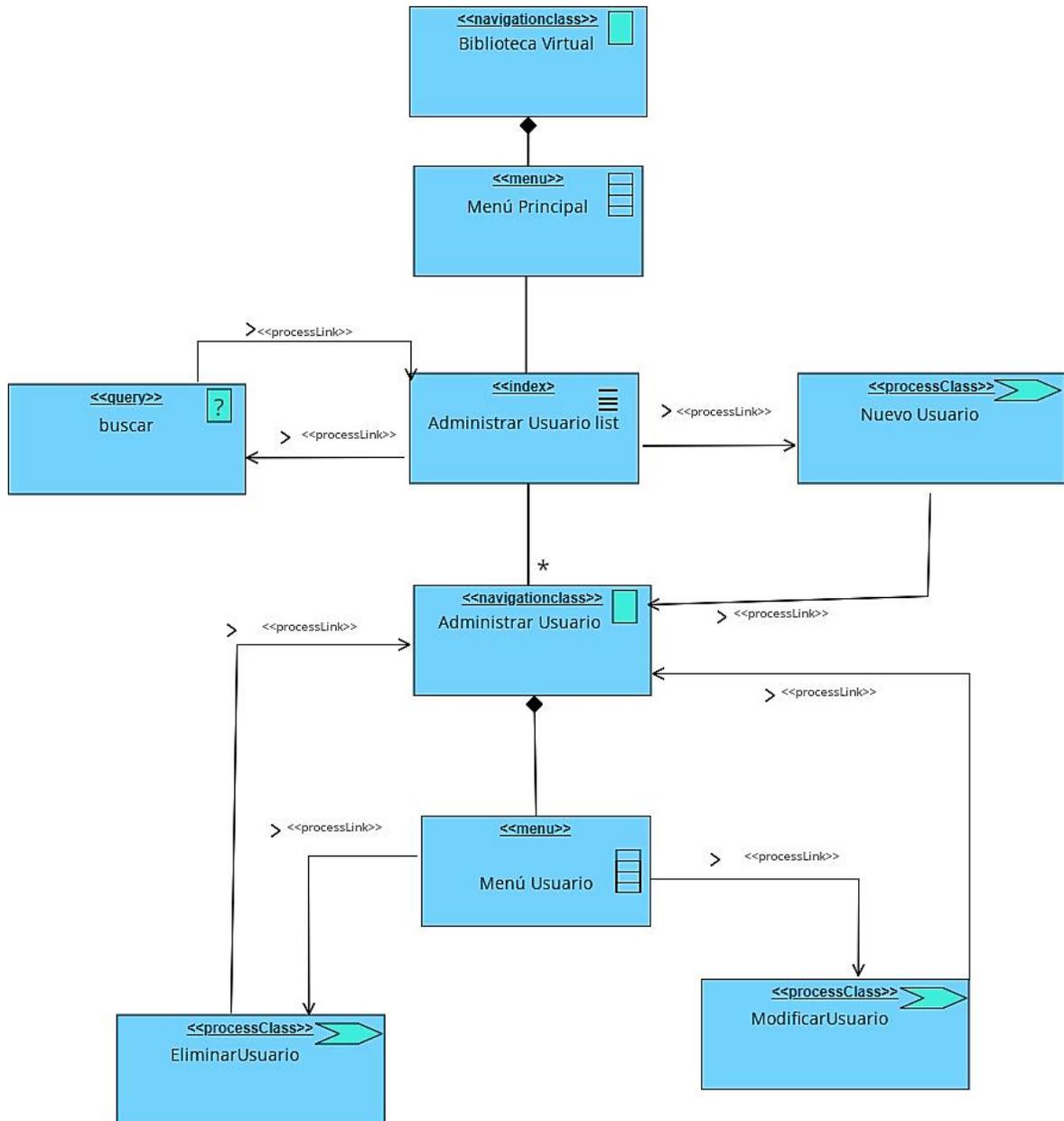
Nota. (Elaboración propia)

3.4.2 Modelo de Navegación

El modelo de navegación tiene como objetivo la representación de los nodos y enlaces de la estructura de hipertexto y el diseño de las rutas de navegación mediante diagramas de clases.

3.4.3 Diagrama de Navegación: Administrar Usuario

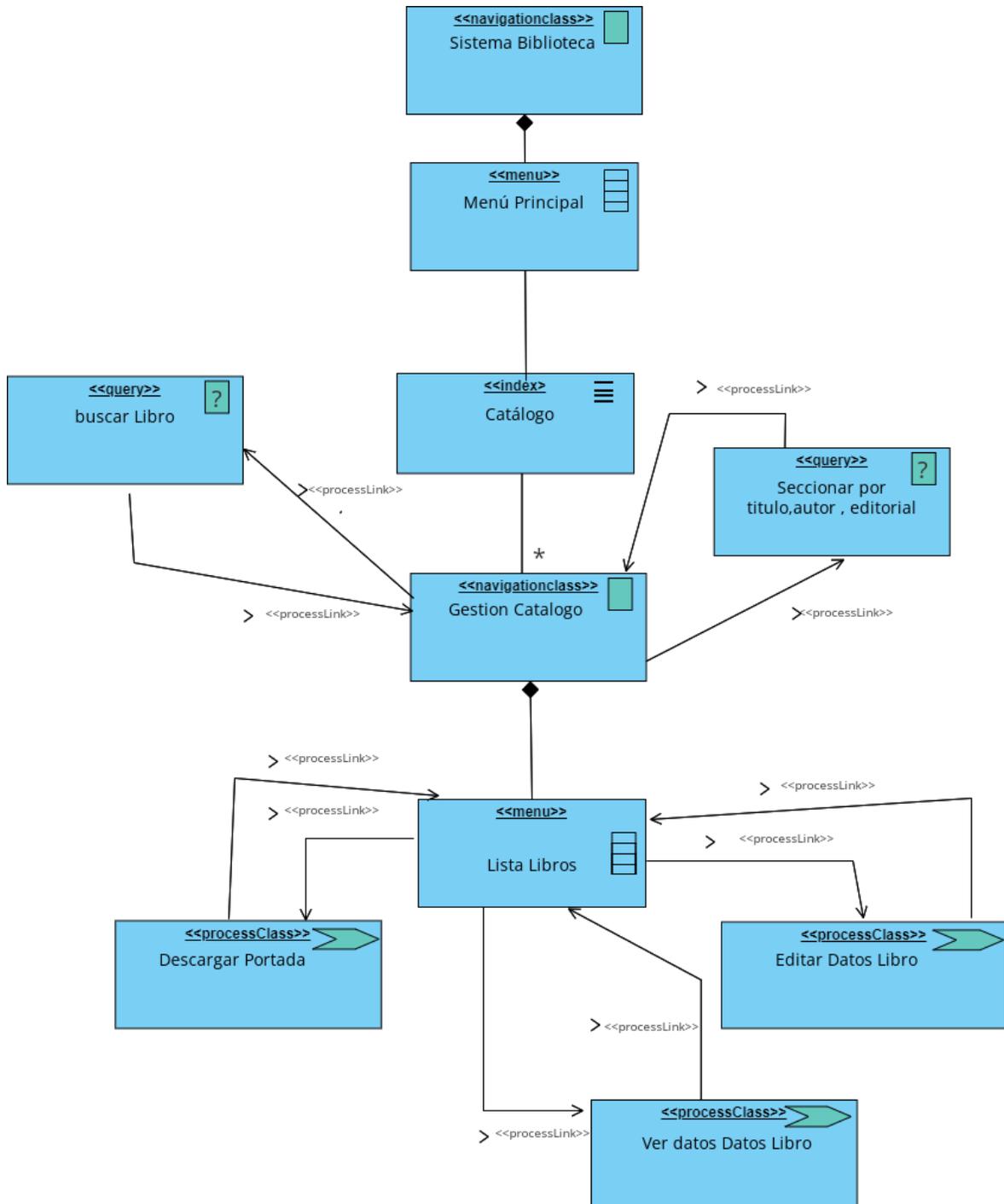
Gráfico 3.13 Diagrama de Navegación: Administrar Usuario



Nota. (Elaboración propia)

3.4.4 Diagrama de Navegación: Catalogo

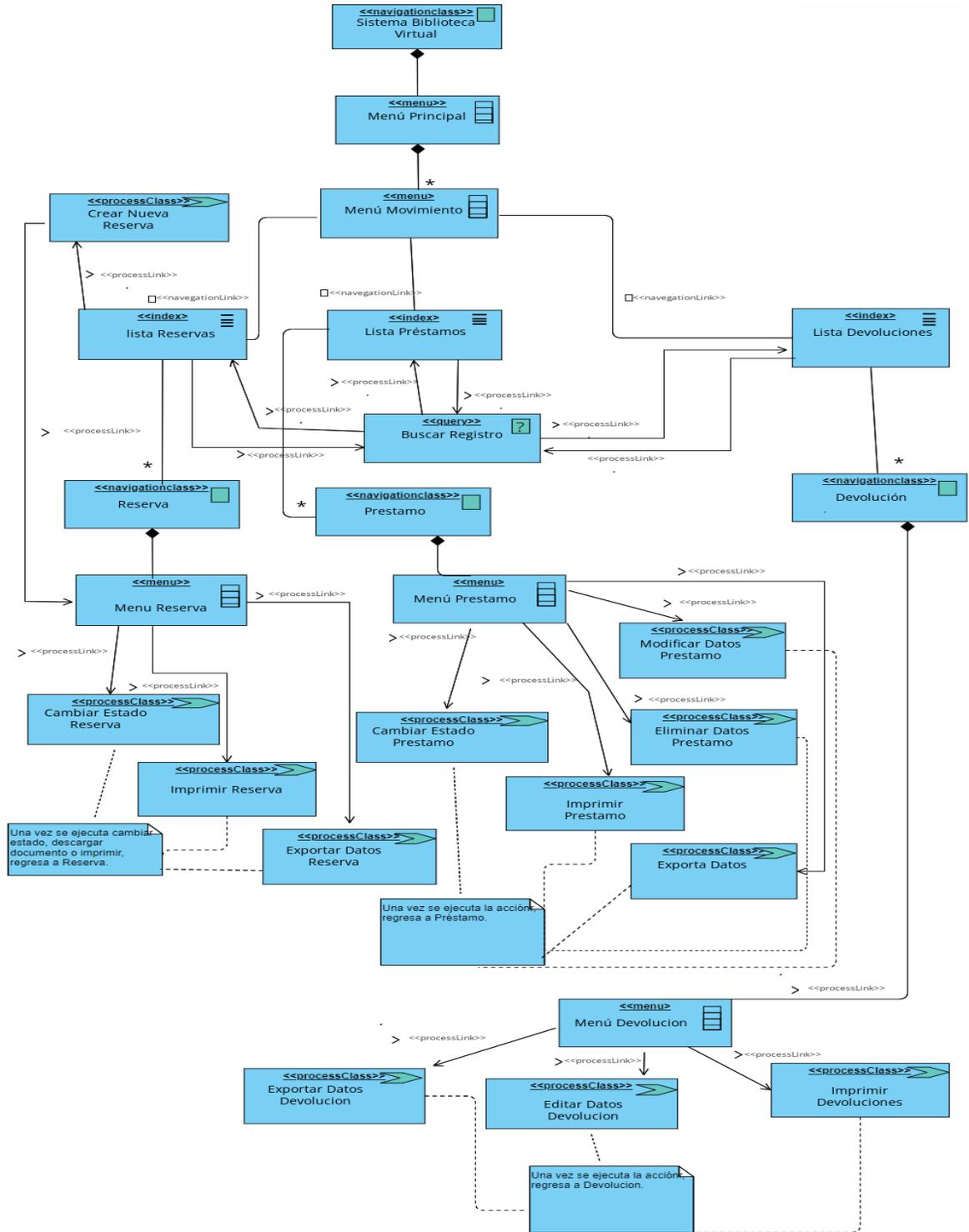
Gráfico 3.14 Diagrama de Navegación: Catalogo



Nota. (Elaboración propia)

3.4.5 Diagrama de Navegación: Préstamo

Gráfico 3.15 Diagrama de Navegación: Préstamo



Nota: (Elaboración propia)

3.5 MODELO DE PRESENTACIÓN

Gráfico 3.16 Diagrama de Presentación: Inicio de la sesión



Diagrama de Presentación: Inicio de la sesión. El formulario está encerrado en un recuadro negro. En la parte superior izquierda hay un círculo azul. A la derecha del círculo, el título "LoginUsuario" está en negrita. Debajo del título, el texto "----INICIAR SESION--" está en rojo y subrayado. Hay dos campos de entrada de texto con líneas punteadas: "Usuario" y "Contraseña". En la parte inferior del formulario, hay un botón rectangular rojo con el texto "Iniciar Sesion" en negro.

Nota. (Elaboración propia)

Gráfico 3.17 Diagrama de Presentación: General (Panel Administrativo)



Diagrama de Presentación: General (Panel Administrativo). El panel tiene un fondo azul claro. En la parte superior izquierda, hay un recuadro negro con el texto "BIBLIOTECA" en blanco. Debajo de "BIBLIOTECA" hay un círculo púrpura y un recuadro negro con el texto "ADMINISTRADOR" en blanco. A la izquierda del panel, hay un menú vertical negro con los siguientes ítems en blanco: "Inicio", "Usuarios", "Carnet", "Mantenimiento", "Logística", "Reportes", "Gráficos" y "Catalogo". En el centro del panel, hay una cuadrícula de ocho botones rectangulares azules con texto negro: "LIBROS", "AUTORES", "EDITORIAL", "USUARIOS" en la primera fila; y "PRESTAMOS", "DEVOLUCION", "CARNET", "RESERVA" en la segunda fila.

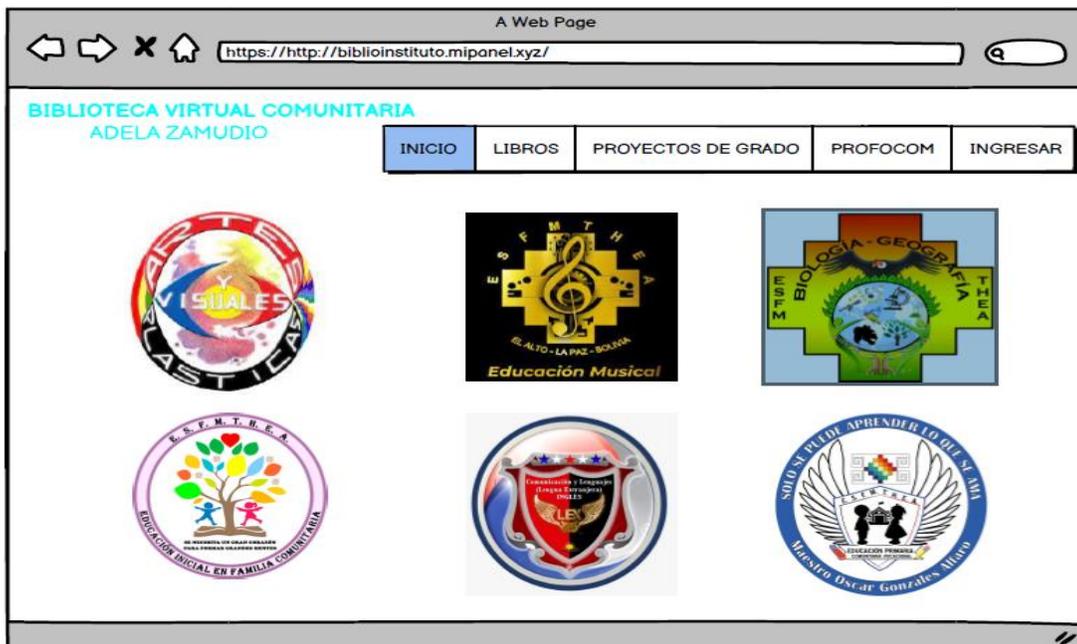
Nota. (Elaboración propia)

Gráfico 3.18 Diagrama de Presentación: Panel Administrativo



Nota. (Elaboración propia)

Gráfico 3.19 Diagrama de Presentación: Panel Administrativo Carreras



Nota. (Elaboración propia)

3.6 IMPLEMENTACIÓN DEL SISTEMA

En la fase de implementación se mostrará el desarrollo de las diversas interfaces que el sistema posee. A continuación, se expone el enlace de acceso.

<http://biblioinstituto.mipanel.xyz/home>

3.6.1 Ingreso al Sistema

Presenta la pantalla o Interfaz de Inicio de Sesión, para el acceso al sistema bibliotecario. El usuario deberá ingresar sus respectivas credenciales para lograr ingresar al sistema y realizar los diversos procesos que en esta se pueden realizar.

Posterior al ingreso de las credenciales del usuario, el sistema permitirá el acceso a la pantalla principal, la cual le despliega una lista de menús, esta acción dependerá de los permisos que el usuario tenga asignado para poder navegar en ella.

Cabe mencionar que esta puede variar de acuerdo al tipo de usuario: Administrador, Bibliotecario.

Gráfico 3.20 Interfaz Principal del Sistema



The image shows a login form with the following elements:

- Header: "Login Usuario" with a user profile icon.
- Section: "INICIAR SESIÓN" in orange text.
- Fields: "Usuario" and "Contraseña" input boxes.
- Toggle: "Mostrar contraseña" with an eye icon.
- Security: A checkbox "No soy un robot" and a reCAPTCHA logo.
- Button: A large red button labeled "Iniciar Sesión".
- Footer: Links for "¿No Tengo una Cuenta?" (with "REGISTRATE" and "SALIR" buttons) and "Me olvide la Contraseña" (with "RECUPERAR CONTRASEÑA" button).

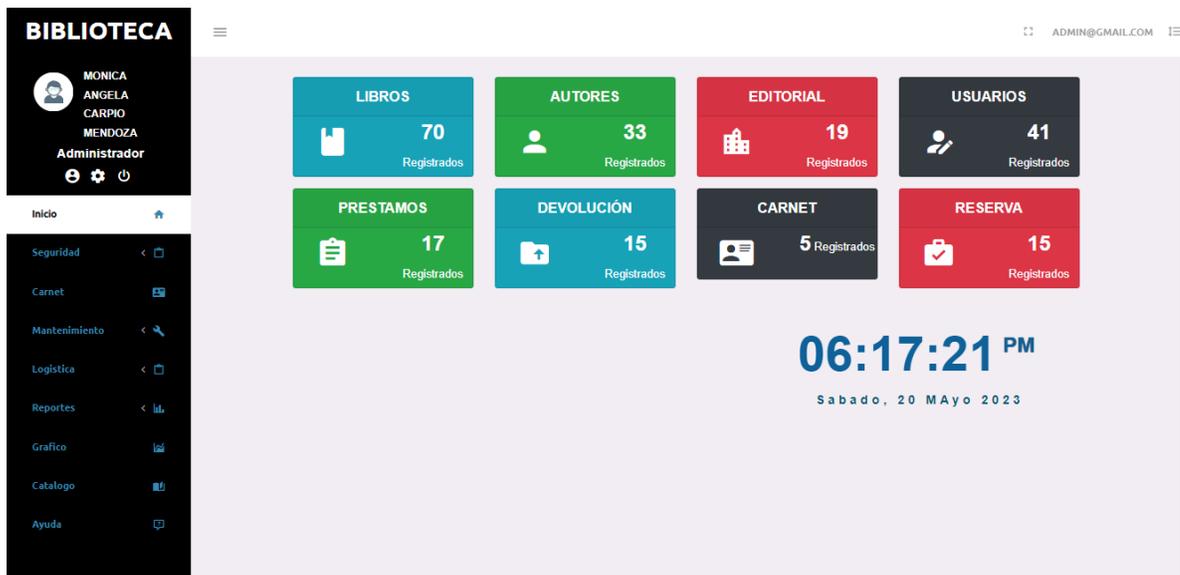
Nota. (Elaboración propia)

3.6.2 Módulos que integran el sistema

Posterior al ingreso de las credenciales del usuario, el sistema permitirá el acceso a la pantalla principal, la cual le despliega una lista de menús, está acción dependerá de los permisos que el usuario tenga asignado para poder navegar en ella.

Cabe mencionar que esta puede variar de acuerdo al tipo de usuario: Administrador, Bibliotecario.

Gráfico 3.21 Pantalla de Inicio Usuario Administrador y Bibliotecario



Nota. (Elaboración propia)

Gráfico 3.22 Lista de Libros Registrados

Nota. (Elaboración propia)

Gráfico 3.23 Lista de Libros Registrados

Codigo ↑	Foto ↑	Nombre ↑	Autor ↑	Editorial ↑	Categoria ↑
LI024-22		IDENTIDAD Y VOCACION	ANONIMO ANONIMO	EDICIONES TRILCE	BIOLOGIA-GEOGRAFIA
LI027-54		PEDAGOGIA DE LA INDIGNACION	ANONIMO ANONIMO	EDICIONES TRILCE	ARTES PLASTICAS Y
LI038-43		COMPOSICIÓN CENTESIMAL , FE Y FM	ANONIMO ANONIMO	EDICIONES TRILCE	QUIMICA
LI039-36		CONFIGURACION ELECTRONICA	ANONIMO ANONIMO	EDICIONES TRILCE	QUIMICA

Nota. (Elaboración propia)

Gráfico 3.24 Lista de Autores Registrados

Autor - Detalle

Código: _____

Nombres: _____ Apellidos: _____

Pais: _____ Fecha Nacimiento: dd/mm/aaaa

Estado:

Cancelar **Guardar**

Nota. (Elaboración propia)

Gráfico 3.25 Lista de Editorial Registrados



Editorial - Detalle [X]

Codigo:

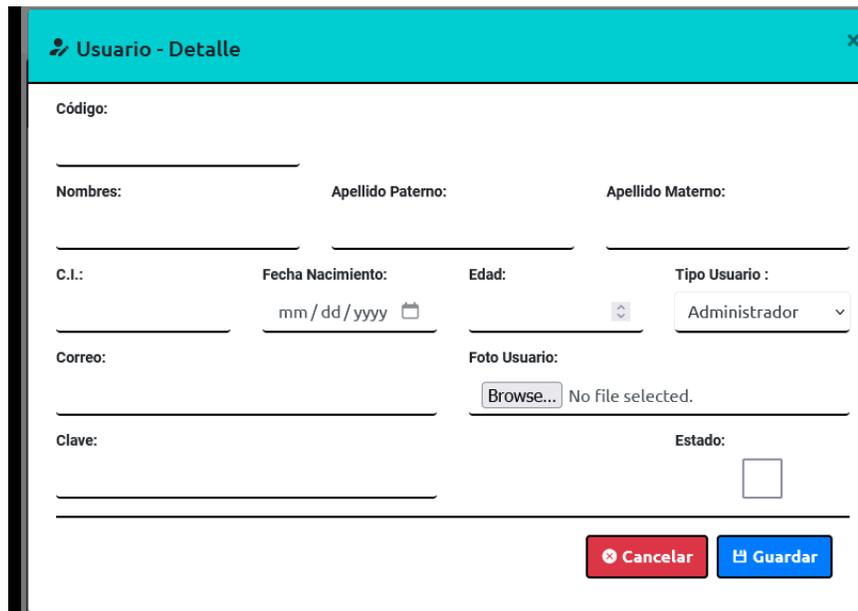
Nombres: _____ Pais: _____

Descripcion: _____ Estado:

Cancelar **Guardar**

Nota. (Elaboración propia)

Gráfico 3.26 Crear Usuario



Usuario - Detalle [X]

Código:

Nombres: _____ Apellido Paterno: _____ Apellido Materno: _____

C.I.: _____ Fecha Nacimiento: mm/dd/yyyy Edad: _____ Tipo Usuario: Administrador

Correo: _____ Foto Usuario: No file selected.

Clave: _____ Estado:

Cancelar **Guardar**

Nota. (Elaboración propia)

Gráfico 3.27 Crear Carnet

Carnet - Detalle [X]

Código: _____

Usuario :
SEBASTIAN CALIXTO RUIZ

Grado: PRIMERO Seccion: A Turno: MAÑANA

Fecha Inscripción: mm/dd/yyyy Fecha Expiración: mm/dd/yyyy Estado:

[Cancelar] [Guardar]

Nota. Elaboración Propia

Gráfico 3.28 Realizar Reserva

Reserva - Detalle [X]

Codigo: _____ Usuario :
GADYS ORIHUELA MAMANI

Libro: 500 ACTIVIDADES PARA EL CURRICULO Cantidad: _____

Fecha Reserva: mm/dd/yyyy Fecha Limite: mm/dd/yyyy Prestado Libro: Estado:

[Cancelar] [Guardar]

Nota. Elaboración Propia

Gráfico 3.29 Lista de Prestamos

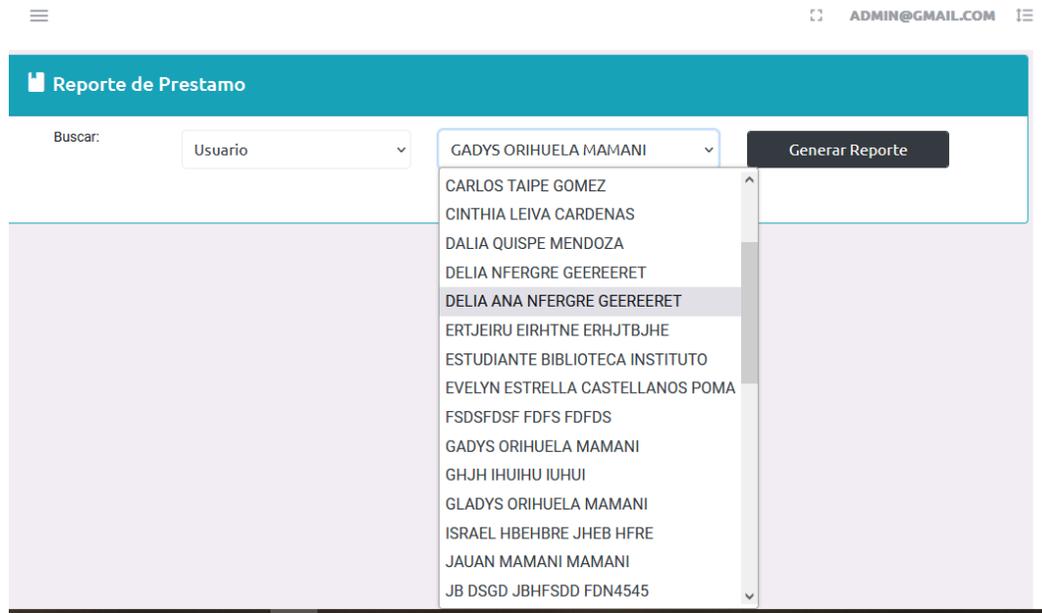
Usuario	Fecha Prestamo	Fecha Devolucion	Fecha Creacion	Estado-Prestamo	Estado	Acciones
ERHJTBJHE	2023-02-04	2023-02-11	2023-06-13 00:00:36	Devuelto	Activo	[Icons]
CASTELLANOS POMA	2021-09-03	2021-09-10	2021-09-02 23:04:10	Devuelto	Activo	[Icons]
NOS POMA	2021-09-08	2021-09-15	2021-09-08 00:33:53	Devuelto	Activo	[Icons]
MAMANI	2023-06-12	2023-06-19	2023-06-11 23:02:42	Devuelto	Activo	[Icons]
CASTELLANOS POMA	2021-11-22	2021-11-29	2021-11-22 14:36:56	Devuelto	Activo	[Icons]

Nota. Elaboración Propia

Gráfico 3.30 Lista de Reportes Libros

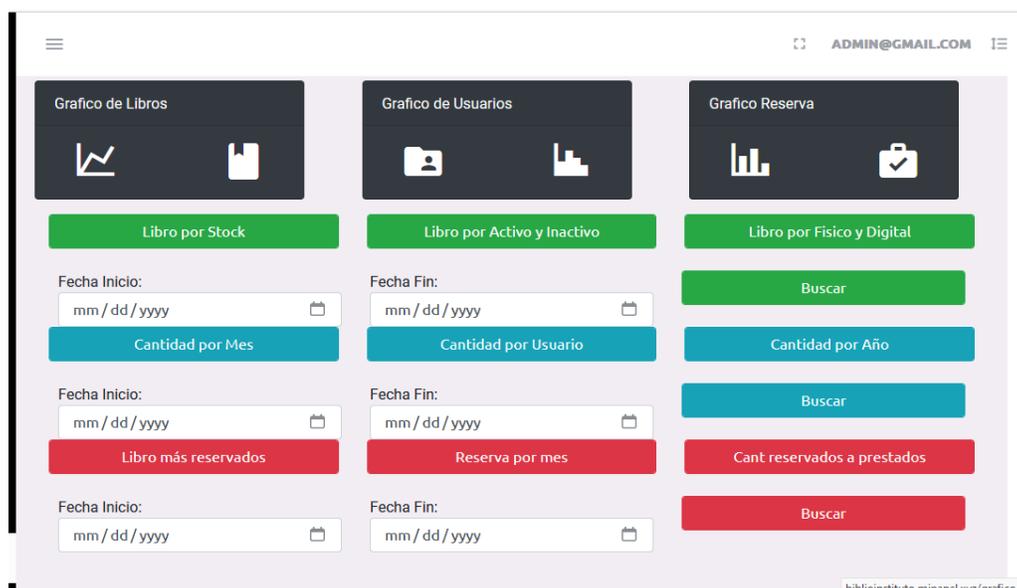
Nota. Elaboración Propia

Gráfico 3.31 Lista de Reportes Usuarios



Nota. Elaboración Propia

Gráfico 3.32 Gráficos y Estadísticas



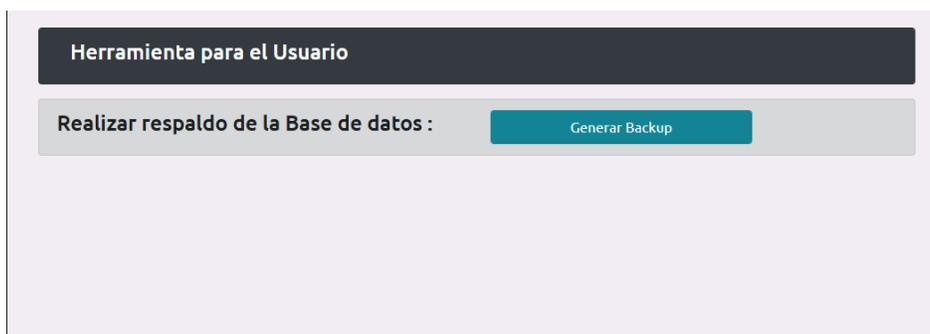
Nota. Elaboración Propia

Gráfico 3.33 *Catalogo por Carreras*



Nota. Elaboración Propia

Gráfico 3.34 *Respaldo Backup*



Nota. Elaboración Propia

**“DISEÑO E IMPLEMENTACION DE UNA BIBLIOTECA
VIRTUAL Y REPOSITORIO DE PROYECTOS DE GRADO
Y TESIS”**

CAPÍTULO IV

CALIDAD Y

SEGURIDAD

2023



4.1 INTRODUCCIÓN

La calidad de software es una actividad primordial que debe aplicarse a lo largo de su desarrollo. Para verificar si las normas de calidad establecidas y los requisitos solicitados por el usuario son cumplidos.

Medir la calidad de software es una tarea difícil, por contar con datos precisos que permitan realizar el análisis objetivo del software desarrollado. Por ello hacemos uso de las técnicas planteadas por Roger Pressman en su libro Ingeniería de software.

4.2 PRUEBAS DE SOFTWARE

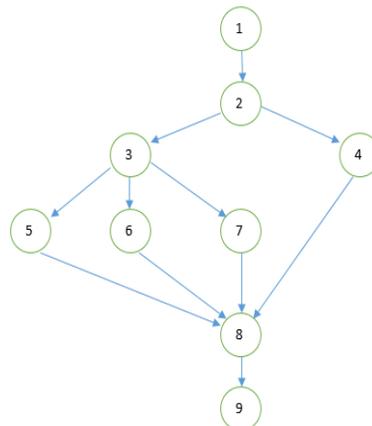
Ahora queda adentrarnos en las evaluaciones dinámicas comúnmente denominadas pruebas de software, para dicha tarea se empleará las técnicas de pruebas de caja negra (pruebas estructurales) y pruebas de caja blanca (pruebas funcionales) mencionadas en capítulos anteriores.

4.2.1 Pruebas de Caja Blanca

Para esta prueba es necesario tener un conocimiento minucioso acerca de la lógica o código a evaluar, pues se toma como prioridad el comportamiento interno y la estructura del programa, entendiendo camino como una sentencia de sentencias encadenadas desde la entrada hasta su salida.

4.2.1.1 Pruebas de Caja Blanca-Prestamos

Gráfico 4.1 Flujo de probabilidades - prestamos



Nota. (Elaboración propia)

Donde:

- Inicio del Sistema (Inicio Flujo) (1)
- Menú Principal (2)
- Módulo de Prestamos (3)
- Módulo de Catalogo Bibliotecario (4)
- Módulo de Reserva (5)
- Registro de Prestamos (6)
- Registro de Devolución (7)
- Fin del Ciclo (8)
- Fin del Sistema (9)

Analizando el grafo generado a partir de las características que posee el sistema, se puede proceder a determinar la complejidad ciclomática del grafo anterior.

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

$$V(G)=P+1$$

Donde:

N : Numero de nodos

A : Numero de aristas

P : Numero de nodos predicados

Reemplazando se obtiene:

$$V(G)=11-9 +2=4$$

$$\therefore V(G)=4$$

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

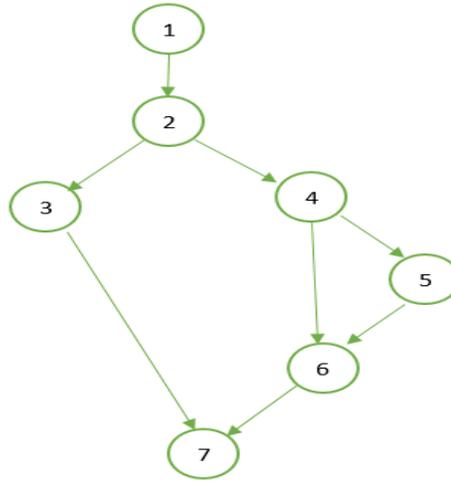
Tabla 3.13 Evaluación de Flujo

Nro.	Ruta	Parámetro de entrada
1	1, 2, 3, 5, 9	Se realiza el listado y se verifica si existen registros de las diferentes solicitudes de préstamos por parte de los lectores.
2	1, 2, 3, 6, 9	Se listan y verifican si existen registros de préstamos.
3	1, 2, 3, 7, 9	Se listan y verifican si existen registros de No Devoluciones.
4	1, 2, 4, 9	Se listan los Catálogos bibliotecarios.

Nota. Elaboración Propia

4.2.1.2 Pruebas de Caja Blanca – Inicio de Sesión

Gráfico 4.2 Prueba de Caja Blanca – Inicio de Sesión



Nota. Elaboración Propia

Donde:

- Inicio de Sesión del sistema (Inicio Flujo) (1)
- Verificación de Ingreso de datos (2)
- Carga de mensajes de errores (3)
- Verificación de Usuario y Contraseña (4)
- Carga de permisos de acceso al sistema (5)
- Carga de mensajes (6)
- Fin del ciclo (7)

Analizando el grafo se obtiene lo siguiente:

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

Reemplazando se obtiene:

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

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

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

Tabla 4.1 Evaluación de Flujo

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

Nota. Elaboración Propia

4.2.2 Pruebas de Caja Negra

4.2.2.1 Pruebas de caja negra Inicio de Sesión

La **Figura 3.19**, muestra el formulario de inicio de sesión, donde se aplicará la prueba de la caja negra, ingresando diferentes valores se registrará el comportamiento y se

verificará si el formulario cumple con la función programada en ella.

Grafico 4.3 Pruebas de Caja Negra- Inicio de Sesión

Nota. Elaboración Propia

Tabla 4.2 Valores Limites (Inicio de sesión)

Campo	Entradas validas	Entradas invalidas
Nombre de usuario	Cadena de texto	Caracteres especiales como, por ejemplo: espacios
Contraseña o clave	Cadena de texto alfanumérico	De igual forma no se permiten caracteres de espacio

Nota. Elaboración Propia

Tabla 4.3 Pruebas de caja negra - Inicio de sesión

Entrada de datos	Salida	Resultados
Usuario Contraseña		

Admin	h0m3r0	Se despliega un mensaje de bienvenida al sistema.	El sistema verifica la existencia del usuario y verifica si la clave o contraseña coinciden con los registros de la base de datos, permitiendo así identificar el tipo de usuario y la carga de los respectivos permisos de acceso al sistema.
Admin	h0m3r0	Despliega un mensaje acerca del error que se está cometiendo con las credenciales.	El sistema verifica la existencia del usuario y verifica si la clave o contraseña coinciden con los registros de la base de datos, si se halla algún error se despliega el mensaje respectivo indicando el error cometido.

Nota. Elaboración Propia

A partir de los resultados de la tabla anterior podemos concluir que la pantalla de inicio de sesión que despliega el sistema cumple con la función programada para un determinado usuario que se identifique y pueda acceder al sistema.

4.2.2.2 Prueba de Caja Negra – Registro de Prestamos

A continuación, se realiza la prueba de caja negra para el formulario de préstamos.

Grafico 4.4 Prueba de caja negra – Registro de préstamos

Nota. Elaboración propia

Tabla 4.4 Valores límites (Registro de Préstamos)

Campo	Entrada valida	Entrada invalida
Reservas Préstamos	Cadena de texto alfanumérico	Debe evitar caracteres de tipo símbolo.
Usuario	Cadena de texto alfanumérico	Debe evitar caracteres de tiposímbolo.
Fecha de reserva para el préstamo	Tipo de dato Fecha	Caracteres del tipoletras, espacios.
Fecha devolución	Tipo de dato fecha	Caracteres del tipoletras, espacios.

Nota. Elaboración propia

Tabla 4.5 Prueba de Caja Negra Registro de Préstamos

Entradas	Ítem bibliotecario	HBSIOA06-0184
	Lector	n
	Fecha de préstamo	08/08/2020
	Fecha devolución	10/08/2020
Salida	Si la información ingresada es correcta se desplegará un mensaje de éxito, caso contrario el sistema despliega los errores cometidos en la inserción.	

Resultado	El sistema valida y verifica la información introducida creando un registro de préstamo y se procede a actualizarla tabla de lista de préstamos, caso contrario se despliega un mensaje de error.
------------------	---

Nota. Elaboración propia

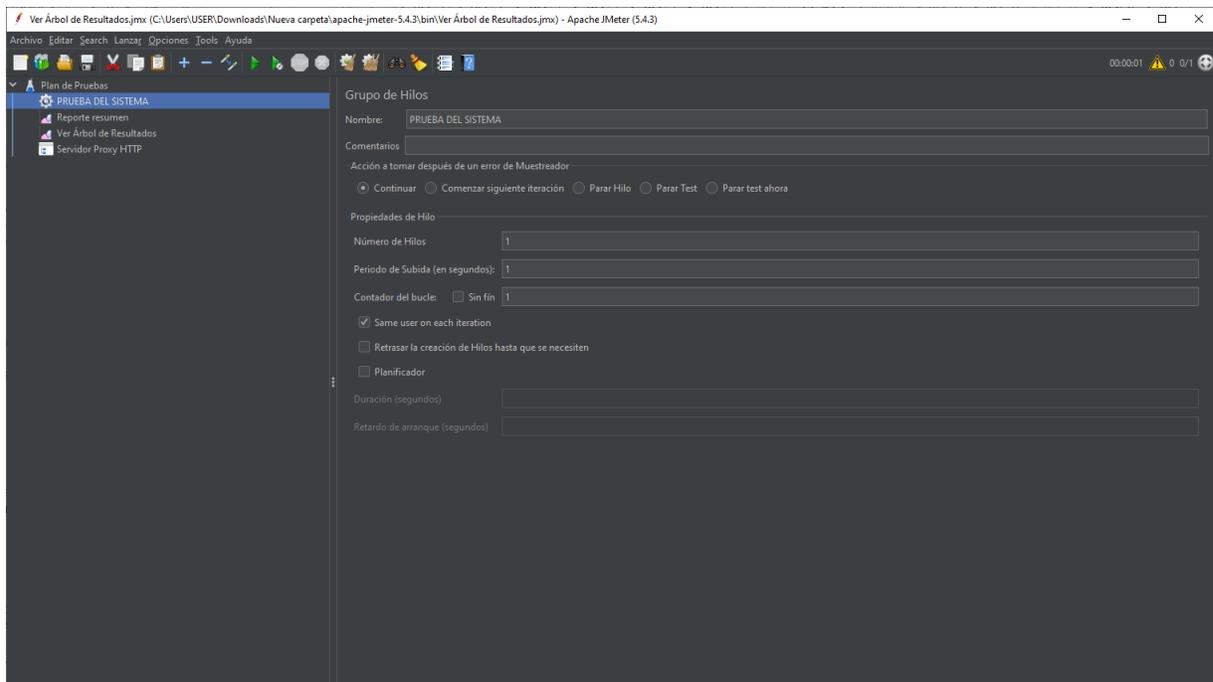
La prueba de caja negra evidencia que el formulario cumple con la función programada de registros de préstamos.

4.2.2.3 Prueba de Estrés

Para realizar las pruebas de estrés al software realizaremos la aplicación de la herramienta JMETER que nos permite la carga de solicitudes al sistema para ver su capacidad de respuesta.

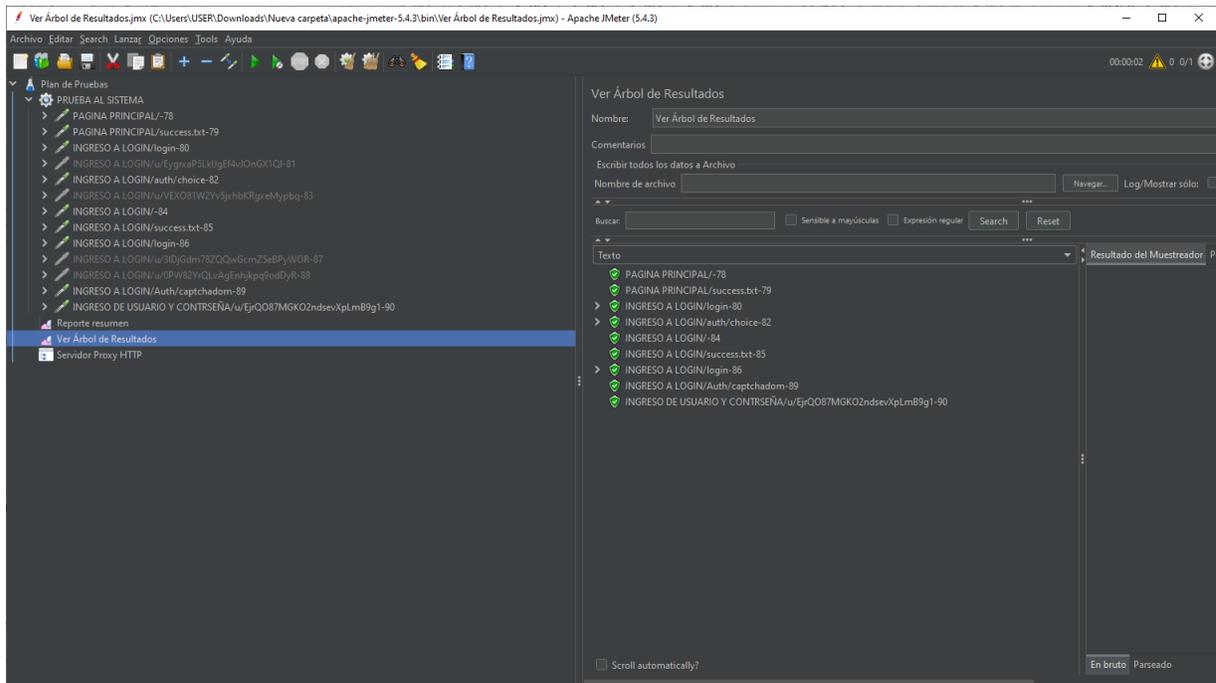
Configuramos todos los datos para la carga del sistema y probamos con un usuario en un segundo.

Grafico 4.5 Software Jmeter



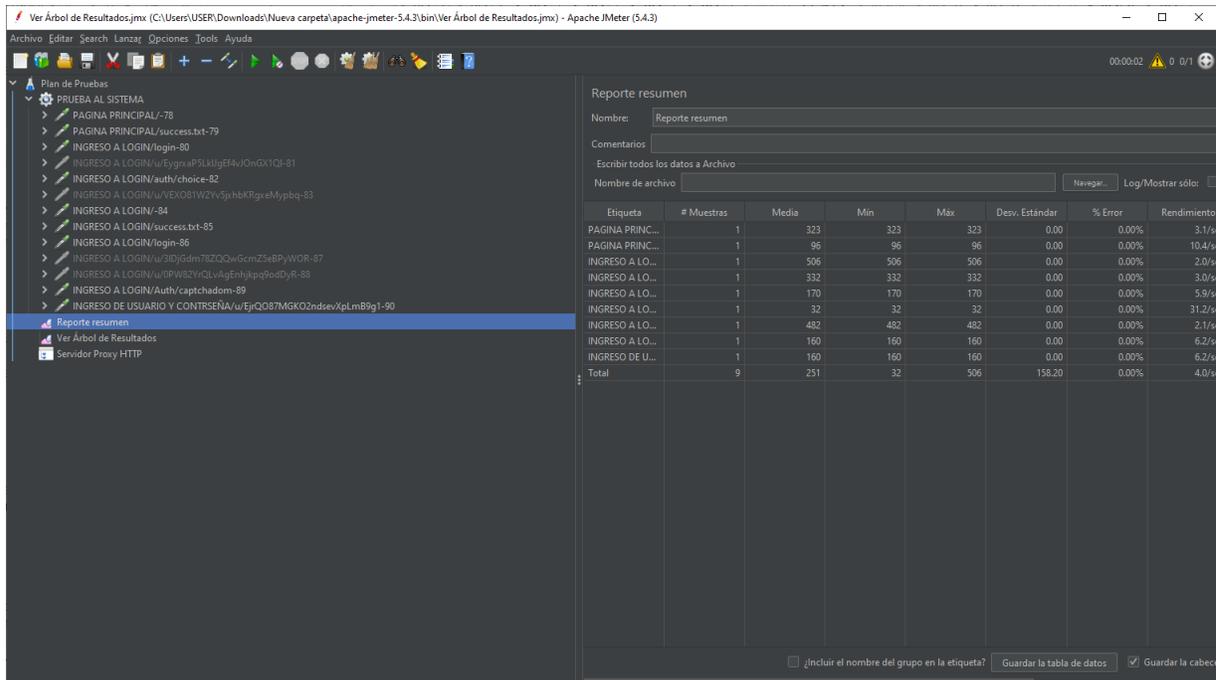
Nota: (Elaboración propia)

Grafico 4.6 Ejecución del programa



Nota: (Elaboración propia)

Grafico 4.7 Resultados de las pruebas



Nota: (Elaboración propia)

Tabla 3.6 Resultados Prueba de estrés

CANTIDAD DE SOLICITUDES	TIEMPO	PORCENTAJE DE ERROR
1 usuarios	1 Segundo	0.00%
100 usuarios	1 Segundo	12.22%
200 usuarios	1 Segundo	26.32%
500 usuarios	1 Segundo	76.98%

Nota: (Elaboración propia)

4.3 NORMA ISO 2500

4.3.1 Funcionalidad

La funcionalidad no puede ser medida directamente, ya que corresponde derivar medidas directas, como los puntos de función que cuantifica el tamaño y la complejidad del sistema en términos de las funciones del usuario, determinando las características del dominio.

Para realizar el cálculo de la cuenta total con factores de ponderación se debe tomar en cuenta lo siguiente:

Tabla 4.7 Características de la Funcionalidad

CARACTERISTICA	DESCRIPCION
Número de entradas de usuario	Se origina en un usuario, cuando este ingresa datos orientados a la aplicación
Número de salidas de usuario	Se cuenta cada salida que proporciona información orientada a la aplicación del usuario
Número de peticiones de usuario	Es una entrada en línea que lleva a la generación de alguna respuesta inmediata por parte del software.
Número de archivos	Se cuenta cada archivo lógico maestro, cada archivo lógico interno que es un agrupamiento lógico de datos como ser parte de una base de datos o archivos independientes.
Numero de interfaces externas	Se cuenta toda las interfaces legibles por la maquina

Nota. (Elaboración propia)

Punto función

El desarrollo de esta técnica de estimación de software, toma en cuenta 5 características:

Número de entradas de usuario: El número de entradas en la que el usuario proporciona datos al sistema

Tabla 4.8 Número de Entradas de usuario

Entradas de Usuario		
1	Administración de administrador	5
2	Administración de docentes	3
3	Administración de estudiantes	3
Total		11

Nota. (Elaboración Propia)

Número de salidas de usuario: El número de salidas que el sistema proporciona al sistema: mensajes, notificaciones, reportes, alertas, etc

Tabla 4.9 Número de Salidas Usuario

Salidas de Usuario		
1	Administración de usuarios	3
2	Administración de docentes	2
3	Administración de estudiantes	3
Total		8

Nota. (Elaboración Propia)

Número de peticiones de usuario: Es el número de entradas interactivas que genera una respuesta del software como salidas interactivas.

Tabla 4.10 Número de Peticiones del Usuario

Peticiones del Usuario		
1	Administración de usuarios	5
2	Administración de técnicos	2
3	Administración de estudiantes	2
Total		9

Nota. (Elaboración Propia)

Número de Archivos: Es el número de archivos, se toma en los grupos lógicos de datos, o archivos independientes.

Tabla 4.11 Archivos Lógicos Internos

N°	ARCHIVOS DE INTERFAZ EXTERNO	CANTIDAD
1	Front-end	1
2	Back-end	1
Total		2

Nota. Elaboración propia

Numero de archivos lógicos internos (ALI): se tomó en cuenta la base de datos y cada tabla cuenta como un archivo.

Tabla 4.12 Archivos Lógicos Internos

N°	CONSULTAS EXTERNAS ARCHIVO LOGICO INTERNOS	CANTIDAD
1	Tablas de la base de datos	23
Total		23

Nota. Elaboración propia

Tabla 4.13 Parámetros de Medición

PARÁMETRO DE MEDICIÓN	CUENTA
Número de entradas de usuario	11
Número de salidas de usuario	8
Número de peticiones de usuario	9
Número de archivos	23
Número de interfaces externas	2

Nota. (Elaboración propia)

Para calcular el punto de función se tiene que realizar el cálculo de la cuenta total con los factores de ponderación especificados en la siguiente tabla:

Tabla 4.14 Factores de Ponderación Punto Función

PARÁMETROS DE MEDICIÓN	CUENTA	FACTOR			TOTAL
		Simple	Medio	Complejo	
Número de entradas de usuario	11	x 3	4	6 =	44
Número de salidas de usuario	8	x 4	5	7 =	40
Número de peticiones de usuario	9	x 3	4	6=	36
Número de archivos	23	x 7	10	15 =	161
Número de interfaces externas	2	x 5	7	10 =	14
	CUENTA TOTAL				295

Nota. (Elaboración propia)

Los valores de ajuste de complejidad son tomadas en cuenta a partir de la siguiente valoración, respondiendo las siguientes interrogantes:

Tabla 4.15 Valores de Ajustes de Complejidad

IMPORTANCIA	0	2	4	6	8	10	FI
	%	0 %	%	0 %	%	0 %	
Escala Factor	No influencia	Incidencia	Moderado	Medio	Significati	Esencial	
	0	1	2	3	4	5	
1. ¿Requiere el sistema copia de seguridad y fiable?					si		4
2. ¿Se requiere comunicación de datos?					si		4
3. ¿Existen funciones de procesamiento distribuido?			si				2
4. ¿Es crítico el rendimiento?				si			3
5. ¿Se ejecuta el sistema en un entorno operativo existente y fuertemente utilizado?					si		4

6. ¿Requiere el sistema entrada de datos interactiva?					si		4
7. ¿Requiere la entrada de datos interactiva que lastransacciones de entrada se lleven a cabo sobre múltiples pantallas o variadas opciones?					si		3
8. ¿Se actualizan los archivos maestros de forma interactiva?					si		3
9. ¿Son complejos las entradas, las salidas, los archivos o peticiones?					si		3
10. ¿Es complejo el procesamiento interno?			si				2
11. ¿Se ha utilizado el código para ser reutilizable?					si		4
12. ¿Están incluidas en el diseño la conversión y la instalación?					si		3
13. ¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?					si		3
14. ¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario?					si		4
TOTAL							46

Nota. (Elaboración propia)

Para el cálculo de los puntos de función (PF) se toma la siguiente relación:

$$PF=cuentaTotal*[0,65+0,01* \Sigma(Fi)]$$

Donde:

$$\Sigma(fi)$$

- Sumatoria de los valores de los factores de ajuste

Se tiene:

$$\Sigma(fi)= 46$$

Dónde: Cuenta Total, es la suma de todas las entradas de los factores de ponderación obtenidas en la tabla anterior. Entonces se tiene.

$$PF=295* [0,65+0,01*46]$$

$$PF=327.45$$

$$\therefore PF =327$$

Entonces considerando al 100%, el nivel de confianza con la sumatoria de $Fi=70$ como valor máximo de ajuste de complejidad entonces se tiene lo siguiente:

$$PF_{maximo}=295*[0,65+0,01*70]$$

$$PF_{maxima}=398.25$$

$$\therefore PF_{maximo}= 398.25$$

Por lo tanto, la funcionalidad tomara el siguiente valor:

$$Funcionalidad = \frac{PF}{PF_{max}} * 100$$

$$Funcionalidad = \frac{327}{398.25}$$

$$Funcionalidad = 0.8210 * 100$$

$$\mathbf{Funcionalidad = 82.1\%}$$

La funcionalidad del sistema se pondera en un 82.1% sin riesgo a fallas y operatividad y un 17.78% susceptible a fallas o colapso del sistema.

4.3.2 Confiabilidad

La confiabilidad del sistema define la probabilidad de operación libre de fallos en un entorno determinado y durante un tiempo específico. Para determinar la confiabilidad de un software especificamos desde el instante que empieza a funcionar es decir $t=0$, a partir de ese momento se realiza las observaciones pertinentes hasta un $t=n$.

- $P(T \leq t)$ Probabilidad de fallas (termino en el cual el sistema trabaja sin fallas)

- $(T \leq t) = 1 - F(t)$, probabilidad de trabajo sin fallas (Tiempo en el cual no ocurren fallas en el sistema).

Para calcular la confiabilidad del sistema se debe tomar en cuenta el periodo de tiempo en el que se ejecuta el sistema, a partir de ello se irán obteniendo las muestras respectivas.

$$F(t) = f * e^{-\mu * t}$$

Donde:

f : Funcionalidad del sistema.

μ : Es la probabilidad de error que puede tener el sistema.

t : Tiempo de duración de gestión en el sistema

Entonces, se considerará un periodo de 20 días como tiempo de prueba donde se define que de cada diez ejecuciones se presenta un fallo con el sistema.

Procedemos a realizar los cálculos respectivos:

Calculando:

$$F[t] = 0.8210 * e^{-\frac{1}{10} \times 20}$$

$$F[t] = 0.82 * e^{-\frac{1}{10} \times 20}$$

$$F[t] = 0.14 * 100 = 14.13\%$$

Tomando en cuenta el resultado anterior y reemplazando el resultado en las fórmulas de probabilidades se tiene lo siguiente:

$$P(T \leq t) = F(t) \text{ donde } P(T \leq t) = 0.1413 = 14.13\%$$

$$P(T \leq t) = 1 - F(t) \text{ donde } P(T \leq t) = 1 - 0.1413$$

$$P(T \leq t) = 0,8587 = 85.87\%$$

Entonces, la confiabilidad del sistema es del **85.87%** en un periodo de 20 días como tiempo de prueba.

4.3.3 Mantenibilidad

Se refiere al conjunto de atributos relacionados con el esfuerzo necesario para realizar determinadas modificaciones en el producto. Los atributos son la capacidad de ser analizado, capacidad para ser modificado, estabilidad y capacidad para ser probado.

El estándar IEEE-982.1 sugiere un índice de madurez del software (IMS) que proporcione una indicación de la estabilidad del producto de software, se determina de la siguiente relación:

$$IMS = \frac{[Mt - (fa + fc + fd)]}{Mt}$$

Donde:

Mt: Número de módulos en la versión actual.

Fc: Número de módulos en la versión actual que se han cambiado.

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

Fd: Numero de módulos en la versión anterior que se han borrado en la versión actual.

Tabla 4.16 Valores para la mantenibilidad

Descripción	Valor
Mt = Numero de módulos de la versión actual	4
Fc = Numero de módulos en la versión actual que se han modificado	1
Fa = Numero de módulos en la versión actual que se han añadido	0
Fd = Numero de módulos de la anterior versión que se han borrado en la versión actual	0

Nota. (Elaboración propia)

Entonces se procede al cálculo del IMS:

$$IMS = \frac{[4 - (0 + 1 + 0)]}{4}$$

$$IMS = 0.75 * 100 = 75\%$$

Entonces se puede indicar que el software tiene una estabilidad del 75% para su mantenibilidad y un 25% restante sería el margen de error por los cambios o modificaciones que sufriera el sistema.

4.3.4 Usabilidad

La usabilidad se refiere al conjunto de atributos que miden el esfuerzo cognitivo necesario que deben realizar los usuarios para utilizar el sistema de software, es decir nos indica la facilidad de uso. Misma que presenta la siguiente ecuación para su respectivo calculo:

$$FU = \left[\frac{\sum x_i}{n} \times 100 \right]$$

Donde:

Xi: Valores de evaluación de cada pregunta.

n: Es el número de preguntas.

Se toma en cuenta la siguiente tabla:

Tabla 4.17 Parámetros de medición de usabilidad

Escal a	Valor
Muy bueno	5
Bueno	4
Regular	3
Malo	2
Pésimo	1

Nota. (Elaboración propia)

Para responder las preguntas, se deben considerar los siguientes valores:

Tabla 4.18 Preguntas para Determinar la Usabilidad

N o	Preguntas	S I	N O	Evaluac ión
1	¿Puede utilizar con fiabilidad el sistema?	5	0	1

2	¿Puede controlar operaciones que el sistema solicite?	4	1	0.8
3	¿El sistema permitió la retroalimentación de información?	4	1	0.8
4	¿El sistema cuenta con interfaz gráfica agradable a la vista?	4	1	0.8
5	¿La respuesta del sistema es satisfactoria?	4	1	0.8
6	¿Le parecen complicadas las funciones del sistema?	3	2	0.6
7	¿Los resultados que proporciona el sistema facilitan el trabajo?	5	0	1
8	¿Durante el uso del sistema se produjo errores?	2	3	0.4
	TOTAL			6.2

Nota. (Elaboración propia)

Calculando la usabilidad:

$$FU = \left[\frac{\sum x_i}{n} \times 100 \right]$$

$$FU = \left[\frac{6.2}{8} \times 100 \right]$$

$$FU = 77.5\%$$

Por lo tanto, existe un 46% de comprensión o entendimiento de los usuarios con respecto a la capacidad del sistema

4.3.5 Eficiencia

La eficiencia mide si el sistema emplea óptimamente los recursos del sistema. Para calcular el grado de eficiencia que tiene el sistema se toma en cuenta la siguiente escala:

Tabla 4.19 Escala de valores de eficiencia

Escala	Valor
Excelente	5
Bueno	4
Aceptable	3

Deficiente	2
Pésimo	1

Nota. (Elaboración propia)

Para valorar la eficiencia del sistema se tiene la siguiente tabla:

Tabla 4.20 Valoración de la eficiencia del sistema

Nº	PREGUNTA	PORCENTAJE
1	¿La distribución y estilo de la interfaz permite que un usuario introduzca con eficiencia las operaciones y la información?	5
2	¿Una secuencia de operaciones (o entrada de datos) puede realizarse con facilidad de movimientos?	4
3	¿Los datos de salida están presentados de modo que se entienden de inmediato?	5
4	¿Las operaciones jerárquicas están organizadas de manera que minimizan la navegación del usuario para hacer que alguna se ejecute?	4
5	¿Procesa y responde adecuadamente cuando realiza alguna consulta o búsqueda?	5
Total		23

Nota. (Elaboración propia)

Para calcular la eficiencia tenemos la siguiente fórmula:

$$E = \frac{\sum x_i}{n} \times \frac{100}{n}$$

Donde:

- $\sum x_i$ = Sumatoria de los valores de eficiencia
- n = Numero de preguntas

Reemplazando se tiene lo siguiente:

$$E = \frac{23}{5} \times \frac{100}{5}$$

$$E = 92\%$$

Con esto concluimos que la eficiencia del sistema es del 92%

4.3.6 Portabilidad

Es la capacidad de un software para ser transferido de un ambiente a otro donde se considera lo siguiente:

- **Adaptabilidad:** Evalúa la oportunidad para adaptar el software a diferentes ambientes sin necesidad de aplicarle modificaciones.
- **Facilidad de Instalación:** Es el esfuerzo necesario para instalar el software en un ambiente determinado.
- **Conformidad:** Permite evaluar si el software se adhiere a estándares o convenciones relativas a portabilidad.
- **Capacidad de reemplazo:** Se refiere a la oportunidad y el esfuerzo usado en sustituir el software por otro producto con funciones similares

El sistema de información web está diseñado en un entorno de acceso vía web mide la portabilidad en: lado del servidor y lado del cliente, la portabilidad del software se enfoca entres aspectos.

- Hardware del servidor
- Sistema operativo del servidor
- Software del servidor

Por lo mencionado anteriormente el sistema de información web es portable en sus diferentes entornos tanto en hardware y software

Para obtener la portabilidad, se tiene la siguiente formula:

Portabilidad= $1 - (\text{numero de dia para portar el sistema} / \text{numero de dias para implementar el sistema})$

$$portabilidad = 1 - \left(\frac{\text{numero de dia para portar el sistema}}{\text{numero de dias para implementar el sistema}} \right) * 100$$

Reemplazando la formula se obtiene el siguiente resultado:

$$portabilidad = 1 - \left(\frac{1}{7} \right) * 100$$

$$portabilidad = 0.85 * 100\%$$

$$portabilidad = 86\%$$

4.3.7 Resultado

De acuerdo a los resultados obtenidos se puede establecer la calidad total del sistema en base a los parámetros medidos anteriormente. La calidad está directamente relacionada con el grado de satisfacción con el usuario que ingresa al sistema.

Tabla 4.21 Resultado

Características	Resultados
Funcionalidad	82.1 %
Confiabilidad	85.87 %
Usabilidad	77.5%
Eficiencia	92%
Mantenibilidad	75%
Portabilidad	86%
Evaluación total	83.08 %

Nota. (Elaboración propia)

Evaluación de calidad total del sistema es de un **83.08 %**.

Según Roger Pressman si la calidad y seguridad sobrepasa de un 60% el sistema es satisfactorio con el usuario que ingresa al sistema. La evaluación de calidad total es de un **83.08%** ósea que cumple con la satisfacción del usuario.

4.4 SEGURIDAD

Es uno de los factores importantes, del software desarrollado para la web, viendo los datos almacenados de suma importancia. Por lo cual las políticas y estrategias de control que se consideran son:

- Seguridad de la base de datos.
- Seguridad con autenticación.
- Seguridad de la aplicación.
- Seguridad y Privacidad de la Información
- Seguridad de la Información en la Institución
- Verificación del Control de Acceso
- Políticas de Privacidad y Confidencialidad
- Políticas de Disponibilidad del Servicio y la Información
- Porcentaje de Implementación de Controles.

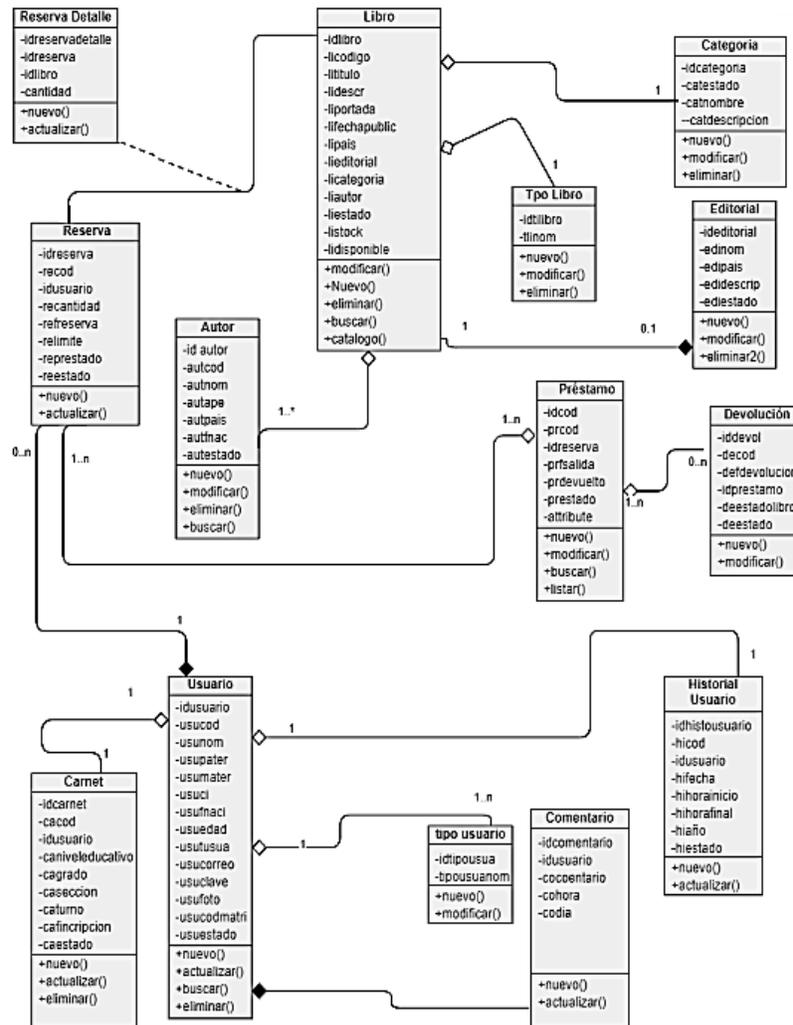
4.4.1 Seguridad de la Base de Datos

Se utilizó como base de datos MySQL, para el resguardo de los datos que se realizó.

- Cuando una acción del usuario en el sistema requiere o pide algunos registros de la base de datos, hay una conexión segura para dicha acción realizada.
- En la seguridad de datos del sistema están registrados el nombre de usuario y contraseña del acceso, según el nivel de acceso que se pueda realizar actividades del sistema.

Toda información de una institución académica es muy valiosa, por lo tanto, el resguardo es muy fundamental, a la hora de la conexión de la base de datos y como en el cierre de forma automática. Pero hay amenaza y la más conocida y comúnmente es el SQL Injection. Y como la implementación de medidas en la restricción de caracteres especiales en los campos de ingresos de texto.

Grafico 4.8 Modelo de base de datos entidad relación



Nota: Elaboración propia

4.4.2 Seguridad de autenticación

Nos referimos al control de sesión o verificación en la autenticación del usuario; como el nombre de usuario y una contraseña, siendo asignados anteriormente.

Como el usuario ingresa la contraseña, esta no se puede mostrar en pantalla, por lo tanto, toda contraseña de cada usuario este encriptado.

Gráfico 4.9 Pantalla de usuario y contraseña

The image shows a login interface for a user. At the top left, there is a circular profile picture of a man in a suit. To its right, the text 'Login Usuario' is displayed in a white serif font on a dark blue background. Below this, the main content area is white and features a red heading 'INICIAR SESIÓN'. There are two input fields: the first is labeled 'Usuario' and the second is labeled 'Contraseña'. Below the password field is a toggle for 'Mostrar contraseña'. A reCAPTCHA widget is present with the text 'No soy un robot' and a small robot icon. Below the reCAPTCHA is a large red button labeled 'Iniciar Sesión'. At the bottom, there is a link '¿No Tengo una Cuenta?' with two buttons: 'REGISTRATE' and 'SALIR'. Below that is another link 'Me olvidé la Contraseña' with a button labeled 'RECUPERAR CONTRASEÑA'.

Nota: Elaboración propia

4.4.3 Seguridad de la aplicación

Se creó un módulo de control acceso al sistema para la restricción del acceso a usuario no autorizado. Esto significa que se verificara y autorizara al usuario por medio de permisos otorgados por el adiestrador del sistema, utilizando uso de las sesiones de MySQL. Se realiza el registro del usuario que modifica la información la base de datos, siendo que se registra en cada tabla el identificador del usuario que modifica la información.

4.4.4 Seguridad y privacidad de la información.

Esto permite que el indicador determine la eficiencia en el tratamiento de eventos relacionados la seguridad de la información, cada evento es reportado por el usuario. El objetivo es reflejar la transparencia de la gestión y evolución del modelo de seguridad y privacidad de información al interior de la institución. Se procede al

seguimiento respectivo de la vulneración a la privacidad y seguridad que reportan los usuarios a la hora de manipular el sistema.

Se detectó anomalías de vulnerabilidad de seguridad que se encontraron son 7, y de los cuales se resolvieron 5.

$$= \left(\frac{\text{numero de anomalias cerradas}}{\text{numero total de anomalias encontradas}} \right) * 100$$

$$= \frac{5}{7} * 100 = 71.43 \%$$

Por lo que refiere que la privacidad de la información es satisfactoria.

- **Sensibilización**

Aquí el indicador mide y permite que la aplicación de los temas sensibilizados en seguridad de la información de los usuarios finales.

Se realizó una capacitación al encargado de la Biblioteca para la utilización del sistema continuamente, también cabe recalcar que hubo fallas y funciones que no contemplan al funcionamiento esperado del sistema; haciendo notar un total de 41.

$$= \left(\frac{\text{Numero de fallas}}{\text{Numero de personal capacitado}} \right) * 100$$

$$E = \frac{38}{45} * 100 = 84.44 \%$$

Por lo que vemos de la sensibilización de información es satisfactoria.

4.4.5 Seguridad de la Información en la Institución

En cumplimiento de las políticas de seguridad de la información en la entidad. La institución trabaja bajo un compendio normativo que regula la seguridad de la información y el personal a cargo del mismo, ya que se trabaja con datos académicos de mucha relevancia e importancia.

4.4.6 Verificación del Control de Acceso

El grado del control al acceso en la entidad; bajo el compendio normativo que regula a todas las ESFMTHEA's a nivel nacional, donde se refleja funciones y atribuciones del responsable de archivo Kardex quien es el encargado de resguardar la información académica ya sea tangible o intangible.

4.4.7 Políticas de privacidad y confidencialidad

El grado de aplicación de políticas de privacidad y confiabilidad de la entidad. Todo registro de estudiantes, docentes y administrativos es de manera personal, ya que cada información es privada y no se puede solicitar a terceras personas.

El compendio normativo regula esta acción, también las autoridades de la institución aplicaron reglamentos y normas internas para fortalecer la política de privacidad y confidencialidad en la información interna de la institución.

4.4.8 Políticas de Disponibilidad del servicio y la información

El grado de cumplimiento en las políticas de disponibilidad el servicio y la información. Los datos ingresados a la institución bajo normas actuales dentro la institución son priorizados de manera que se evite la burocracia, prebendo el tiempo del usuario.

4.4.9 Porcentaje de Implementación de Controles

El grado de avance en la aplicación de controles de seguridad. En el ámbito de la institución que es académicamente se toma mucho control de seguridad, ya que no debe sufrir alteraciones de la información. Se planteó y se aplicaron 3 controles de seguridad y se logró aplicar 5.

$$= \left(\frac{\text{Numero de controles implementados}}{\text{Numero de controles planteados}} \right) * 100$$

$$= \frac{3}{5} \times 100 = 60\%$$

Como podemos ver los controles de seguridad de la información son satisfactorias.

4.5 SISTEMA DE GESTIÓN DE SEGURIDAD DE LA INFORMACIÓN ISO – 27002

Esta ISO evalúa la implementación para cumplir mediante las normas, así como el conjunto de controles que reducen el riesgo de sufrir incidentes de seguridad del funcionamiento de la institución en cuanto a la seguridad de información, ahora veremos los siguientes tipos de seguridad:

4.5.1 Seguridad Lógica

- **Copias de seguridad**

El Bibliotecario como administrador deberá periódicamente cada 7 días sacar copias de seguridad para el resguardo del sistema en la base de datos que se va subiendo cada día.

- **Identificación y autenticación**

Evita el ingreso de personas ajenas que no están autorizados, por eso el sistema cuenta con el control estricto del ingreso Usuario y Contraseña debidamente controlada.

- **Encriptación**

Se implementa la seguridad en cuanto a la encriptación para la contraseña, siendo un dato importante del ingreso al sistema, utilizando el algoritmo de Hash de alta seguridad de encriptación.

Tabla 4.22 Tabla Hash

Descripción	Duración
Longitud fija de salida	23 bytes o 184 bits
Clave base de Datos	60 caracteres

Nota: Elaboración propia

4.5.2 Seguridad física

Se recomienda los back – up, 3 copias que sean almacenadas en distintos lugares para resguardar datos o errores de mala manipulación del sistema. El back – up de la base de datos deberán ser protegidos de forma segura y solo permitido el acceso al personal autorizado de la institución.

- **Equipamiento:**

Se deberá realizar un mantenimiento preventivo de los equipos e instalaciones para la protección física de los activos de la institución.

- **Sistema de control y acceso físico**

Son restringidos los accesos de áreas críticas a usuarios no autorizados, y reducir el riesgo de información fraudulenta.

4.5.3 Seguridad Organizada

Toda información dentro del sistema deberá recibir protección muy adecuada como ser:

- **Gestión de archivos**

Para manejar los Back – ups se clasifican de acuerdo a la fecha en que se realizan los mismos.

**“DISEÑO E IMPLEMENTACION DE UNA
BIBLIOTECA VIRTUAL Y REPOSITORIO DE
PROYECTOS DE GRADO Y TESIS”**

CAPÍTULO V

**ANALISIS DE
COSTOS**

2023



5 CAPITULO V ANÁLISIS DE COSTOS

5.1 INTRODUCCIÓN

Existen distintos métodos establecidos para la estimación de costos de desarrollo de software, estos métodos son relaciones matemáticas entre el esfuerzo y el tiempo de desarrollo.

5.1.1 Método de Estimación COCOMO II

Los métodos de estimación de costos en proyectos de software ayudan a indicar las razones presupuestarias, planificar y controlar el proyecto y enriquecer el análisis de inversión de los proyectos de software, estos métodos nos ayudan a establecer una relación matemática entre el esfuerzo y el tiempo de desarrollo.

COCOMO II posee tres modelos:

- **ORGANICO:** Proyectos relativamente sencillos, menores a 5000 líneas de código, implica procesamiento de datos, uso de la base de datos se focaliza en transacciones y recuperación de datos.
- **SEMIACOPLADO:** Proyectos intermedios en complejidad y tamaño. La experiencia en este tipo de proyectos es variable y las restricciones intermedias.
- **EMPOTRADO:** Proyectos bastantes complejos, en los que apenas se tiene experiencia y en un entorno de gran innovación técnica. Para la estimación de costos de este proyecto se empleará el submodelo post arquitectura.

Para calcular el esfuerzo, necesitaremos hallar la variable **KLDC** (kilo-líneas de código).

Este proyecto se implementa 9055 Líneas de Código en el lenguaje Visual Studio. Aplicando Conversiones se tiene:

LCD = 9055

$$KLCD = \frac{LCD}{1000}$$

$$KLCD = \frac{10130}{1000}$$

$$KLCD = 10.13$$

Es un Modelo **Semi acoplado**.

Tabla 5.1 Coeficientes de a,b,c,d de COCOMO II

Proyecto Software	A	b	c	d
Orgánico	3.2	1.05	2.5	0.38
Semi acoplado	3.0	1.12	2.5	0.35
Empotrado	2.8	1.20	2.5	0.32

Nota. (COCOMO, 2013)

Debido a que el proyecto no supera las 50000 líneas de código se utilizarán los coeficientes para proyectos **semi-acoplado**. Ecuaciones para calcular el costo de software:

Tabla 5.2 Ecuaciones del Método COCOMO II

Variable	Ecuación	Tipo / Unidad
Esfuerzo requerido por el proyecto	$E=a*(KLDC)b*FAE$	Persona/mes
Tiempo requerido por el proyecto	$T=c*(E)d$	Mes
Número de personas requeridos para el proyecto	$NP=ET$	Personas
Costo Total	$CT=SueldoMes*NP*T$	\$us

Nota. (Prentice – Hall, 1981)

Tabla 5.3 Coeficientes del Modelo COCOMO II

Modo	a	b	c	d
Orgánico	3.20	1.05	2.50	0.38
Semi Orgánico	3.00	1.12	2.50	0.35

Empotrado	2.80	1.20	2.50	0.32
------------------	------	------	------	------

Nota. (Elaboración propia)

5.1.1.1 Aplicación del modelo COCOMO Intermedio

COCOMO es un modelo de estimación de costos de software, orientado a la magnitud del producto final midiendo el tamaño del proyecto. Este modelo ayuda a estimar esfuerzo, tiempo, gente y costo del proyecto en este caso se emplea el modelo de aplicación intermedio.

a) Estimación de Puntos de Función sin Ajustar

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

Para obtener el punto de Función sin Ajustar y el Ajuste de Complejidad nos dirigimos a las tablas: Tabla N° 4.8 y Tabla N° 4.9 donde los valores son:

$$\text{Conteo Total: } 295 \text{ y } \sum(Fi) : 46$$

$$\text{Multiplicador} = 0.65 + 0.01 * \sum(Fi)$$

$$\text{Multiplicador} = (0.65 + 0.01 * 46) = 1.11$$

Remplazando en la ecuación:

$$PFA = \text{Conteo Total} * \text{Multiplicador}$$

$$PFA = 295 * 1,11 = 327.45$$

Conversión de los Puntos de Función a KLDC

Convertimos los PFA a miles de líneas de código. Para ello veremos la **Tabla 5.4**

Tabla 5.4 Conversión de Puntos de Función a KLDC

Lenguaje	Nivel es	Factor LDC / PF
C	2.5	128
Ansi Basic	5	64
Java	6	53
Ansi Cobol	3	107

Visual Basic	7	46
ASP	9	36
PHP	11	29
Visual Code	9.5	34

Nota: Elaboracion Propia

Aplicando las conversiones:

$$KLDC = \frac{\text{Factor LCD} * PFA}{1000}$$

$$KLDC = \frac{34 * 327.45}{1000}$$

$$KLDC = 11.13$$

b) Determinar Factor de Ajuste del Esfuerzo (FAE)

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

Tabla 5.5 Cálculo de Atributos FAE

Atributos	Valor					
	Muy bajo	Bajo	Nominal	Alto	Muy alto	Extra alto
Atributos del Software						
Atributos del Software						
Fiabilidad	0,75	0,88	1,00	1,15	1,40	
Tamaño de base de datos		0,94	1,00	1,08	1,16	
Complejidad	0,70	0,85	1,00	1,15	1,30	1,65
Atributos del Hardware						
Restricciones de tiempo de ejecución			1,00	1,11	1,30	1,66
Restricciones de memoria virtual			1,00	1,06	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 del 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 en la máquina virtual	1,21	1,10	1,00	0,90		
Experiencia en lenguaje programación	1,14	1,07	1,00	0,95		
Atributos del Proyecto						
Uso de técnicas actuales de programación	1,24	1,10	1,00	0,91	0,82	
Utilización de herramientas software	1,24	1,10	1,00	0,91	0,83	
Restricciones de tiempo de desarrollo	1,23	1,08	1,00	1,04	1,10	
TOTAL, FAE = 0.3782						

Nota: Valores de atributo de costes FAE. Fuente: (elaboración propia)

A partir de las demarcaciones en los diferentes atributos que posee nuestro proyecto podemos calcular el valor de Factor de Ajuste de Esfuerzo (FAE).

$$FAE = 1.15 * 1.00 * 0.85 * 1.00 * 1.00 * 1.00 * 1.00 * 0.86 * 0.82 \\ * 0.86 * 0.90 * 0.95 * 0.82 * 0.91 * 1.00 = \mathbf{0.3782}$$

$$\therefore FAE = \mathbf{0.3782}$$

Aplicando las ecuaciones anteriores, así también las diferentes constantes se obtienen los siguientes resultados.

Esfuerzo:

$$E = a * (KLDC)^b * FAE \text{ (persona/mes)}$$

$$E = 3 * (11.13)^{1.12} * 0.3782 \text{ (persona/mes)}$$

$$E = \mathbf{16.86 \text{ (persona/mes)}} \sim \mathbf{16 \text{ personas/Mes}}$$

Calculando el Tiempo de desarrollo:

$$T = c * (E)^d$$

$$T = 2.5 * (16.87)^{0.35} (\text{Meses})$$

T = 6.7 Equivale a **6.7 Meses**

Cálculo de la Productividad

$$PR = \frac{LCD}{Esfuerzo} \left(\frac{LCD}{Personas} Mes \right)$$

$$PR = \frac{10.13}{16} \left(\frac{LCD}{Personas} Mes \right)$$

$$PR = 0.63 \left(\frac{LCD}{Personas} Mes \right)$$

Cálculo del personal requerido

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

$$NP = \frac{16}{6.7} \text{Personas}$$

$$NP = 2.38 \cong 2 \text{Personas}$$

Cálculo de Costo Personas mes (Salario promedio = 350 \$ o 2450 Bs)

Costo Mes = Salario promedio entre programadores

$$\text{Costo Mes} = 350 \$$$

$$\text{Costo Mes} = \text{Número Personas} * \text{Salario Promedio}$$

Cálculo de Costo Total del Sistema.

$$CT = \text{Sueldo Mes} * NP * T$$

$$CT=350*2*6.7$$

$$CT= 4.690\$us$$

En resumen, se requiere **2 personas** estimando un trabajo de **6.7 meses** y con costo total de **4690 \$us** Equivalente en bolivianos a **32.830 Bs**

Tabla 5.6 resultado Final

Variable	Tipo / Unidad
Tiempo requerido por el proyecto	6.5 meses
Número de personas requeridos para el proyecto	2 personas
Costo Total	4550\$us

Nota. Elaboración Propia

**“DISEÑO E IMPLEMENTACIÓN DE UNA BIBLIOTECA
VIRTUAL Y REPOSITORIO DE PROYECTOS DE
GRADO Y TESIS”**

CAPÍTULO VI

2023

**CONCLUSIONES Y
RECOMENDACIONES**



6 CAPITULO VI CONCLUSIONES Y RECOMENDACIONES

6.1 CONCLUSIONES

Una vez finalizado con el desarrollo del “DISEÑO E IMPLEMENTACION DE UNA BIBLIOTECA VIRTUAL Y REPOSITORIO DE PROYECTOS DE GRADO Y TESIS” se ha logrado alcanzar con el objetivo principal planteado bajo los requerimientos de la institución.

Si bien el proceso de registro de la información era semiautomático y de forma individual por parte de los técnicos bibliotecarios, con el presente proyecto se logró centralizar la información y efectuar un control adecuado de los materiales bibliográficos. Coadyuvando en una mejor administración y gestión de la información en la escuela superior de formación de maestras y maestros tecnológico humanístico el Alto (ESFMTHEA).

- Analizar los requerimientos mínimos y herramientas necesarias tanto en hardware como en el software para albergar e implementar el sistema bibliotecario repositorio institucional; Este objetivo se alcanzo al 100% con la actividad de encuestas, test.
- Realizar el diseño correspondiente del sistema, desarrollando los diferentes módulos que poseerá para la administración de la información centralizada. Este objetivo se alcanzo al 100%.
- Construir la plataforma web de la biblioteca y repositorio para incrementar la difusión e interacción con los usuarios. Este objetivo se alcanzo al 100%.
- Implementar una biblioteca virtual, que incluya como recurso tecnológico a nuevas tecnologías de la información y comunicación, por alta demanda de los usuarios; este objetivo alcanzo el 100% del objetivo trazado.
- Sistematizar el registro y respaldo de documentos de investigación facilitando el alcance de los mismos a los investigadores; este objetivo alcanzo el 100% del objetivo trazado

- Suministro del material bibliográfico con el uso correcto de normas de catalogación (Dewey) este objetivo alcanzo el 70% del objetivo trazado.
- Generar Reportes de libros, estudiantes; este objetivo alcanzo el 100% del objetivo señalado.

Finalmente se concluye que en el desarrollo del “DISEÑO E IMPLEMENTACION DE UNA BIBLIOTECA VIRTUAL Y REPOSITORIO DE PROYECTOS DE GRADO Y TESIS” se cumplieron todos los objetivos, por lo que es un aporte tecnológico oportuno ya que se redujo el tiempo de registro, búsqueda y el acceso a los materiales digitales, cabe recalcar que el manejo de esta información se realiza de forma segura, rápida, precisa y confiable.

6.2 RECOMENDACIONES

Debido a las diferentes características que presenta el sistema se recomienda lo siguiente:

- Capacitar a los nuevos usuarios, con el fin de operar el sistema de forma correcta.
- Se recomienda realizar las respectivas copias de seguridad a la base de datos de forma periódica (cada 24 horas), con el fin de poseer respaldos de la información, que aseguren la integridad del sistema.
- Para posteriores mejoras del sistema se recomienda desarrollar un módulo para sugerir la adquisición de nuevos ejemplares bibliográficos.
- Aumentar el personal para la suministración del material bibliográfico total.

BIBLIOGRAFÍA

- A, D. (2023). *hostinger.es*. Recuperado el 2023, de hostinger.es:
<https://www.hostinger.es/tutoriales/que-es-bootstrap>
- Aduviri, E. (2013). *Capterra*. Recuperado el 2023, de Capterra:
<https://www.capterra.mx/glossary/206/deployment#:~:text=El%20concepto%20de%20%22implementaci%C3%B3n%22%20se,de%20ejecuci%C3%B3n%20y%20la%20edici%C3%B3n.>
- Alegsa, L. (2023). *ALEGSA.com*. Recuperado el 2023, de ALEGSA.com:
https://www.alegsa.com.ar/Dic/implementacion_de_sistemas.php#gsc.tab=0
- Cabrera, E. D. (2015). *Repositorio (contenido digital)*. Recuperado el 2023, de wikipedia.org:
[https://es.wikipedia.org/wiki/Repositorio_\(contenido_digital\)](https://es.wikipedia.org/wiki/Repositorio_(contenido_digital))
- Eby, K. (2017). *Smartsheet Inc*. Recuperado el 2023, de Smartsheet Inc:
<https://es.smartsheet.com/ultimate-guide-project-cost-estimating>
- Espitia, M. E. (2020). *Análisis y Diseño de Sistemas de Información*. Recuperado el 2023, de unipamplona.edu.co:
https://www.unipamplona.edu.co/unipamplona/portallG/home_109/recursos/octubre2014/administraciondeempresas/semestre7/11092015/analisisydisenosistinformacion.pdf
- Et, B. (2010). *Repositorios: definición y tipos*. Recuperado el 2023, de poliscience.blogs.upv.es: <https://poliscience.blogs.upv.es/acceso-abierto/repositorios-2/>
- Gómez, J. (2017). *laboratorioti.com*. Recuperado el 2023, de laboratorioti.com:
<https://www.laboratorioti.com/2013/04/15/estimacion-de-costes-con-cocomo-81-i/>
- Lapiente, M. J. (2013). *Metadatos Dublin Core*. Recuperado el 2023, de hipertexto.info:
http://www.hipertexto.info/documentos/dublin_core.htm

Lorenzon, E. (2020). Sistemas y organizaciones. En E. Lorenzon, *Sistemas y organizaciones* (pág. 10). edulp.editorial@gmail.com.

Luca, D. D. (2015). *damiandeluca.com.ar*. Recuperado el 2023, de damiandeluca.com.ar:
<https://damiandeluca.com.ar/visual-studio-code-caracteristicas-principales>

MAGALLAN, A. V. (2016). *es.scribd.com*. Recuperado el 2023, de es.scribd.com:
<https://es.scribd.com/presentation/615399225/PRUEBA-Y-CALIDAD-DEL-SOFTWARE#>

Martines, G. R. (2022). *developer*. Recuperado el 2023, de developer:
<https://developer.mozilla.org/es/docs/Web/CSS>

martinez, h. i. (2013). *abdonflores.wordpress.com*. Recuperado el 2023, de
abdonflores.wordpress.com: <https://abdonflores.wordpress.com/2013/11/19/que-es-wamp-server/>

Matos, P. (2008). *scielo.sld.cu*. Recuperado el 2023, de scielo.sld.cu:
http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1024-94352008001100002

Mendez, G. (2015). *isotools.us*. Recuperado el 2023, de isotools.us:
<https://www.isotools.us/normas/riesgos-y-seguridad/iso-27001/#1608132521160-18c8e38d-41e91435-010d1d7b-3f6396e6-8793>

Menéndez, R. (2019). *um.es*. Recuperado el 2023, de um.es:
<https://www.um.es/docencia/barzana/DAWEB/Lenguaje-de-programacion-JavaScript-1.pdf>

MUÑOZ, C. (2021). *DOC.PLAYER*. Recuperado el 2023, de DOC.PLAYER:
<http://docplayer.es/195037722-Programacion-web-introduccion-al-html-5-cesar-munoz.html>

Ordoñez, J. A. (2008). *bitstream*. Recuperado el 2023, de bitstream:

<https://www.tdx.cat/bitstream/handle/10803/10913/AriasOrdonez.pdf>

Ortega, J. (2009). *Dataprix*. Recuperado el 2023, de Dataprix:

<https://www.dataprix.com/es/bases-datos-master-software-libre-uoc/bases-datos-mysql>

Pantoja, E. B. (2004). *scielo.org.bo*. Recuperado el 2023, de scielo.org.bo:

http://www.scielo.org.bo/scielo.php?script=sci_arttext&pid=S1683-07892004000100005

Pasaje, J. L. (2005). *Metodología y Herramientas UML*. Recuperado el 2023, de tesisenred.net :

<https://www.tesisenred.net/bitstream/handle/10803/10633/JMPtesis.pdf>

Quiroga, E. (2019). *northware.mx*. Recuperado el 2023, de northware.mx:

<https://www.northware.mx/blog/que-es-jquery/#:~:text=jQuery%20es%20una%20biblioteca%20de,AJAX%20para%20el%20desarrollo%20web.>

Rodriguez, J. (2012). *unitedpress.info*. Recuperado el 2023, de unitedpress.info:

http://unitedpress.info/ocacalifax2/?page_id=265

RODRÍGUEZ, M. (2002). *Metadatos*. Recuperado el 2023, de ecured:

<https://www.ecured.cu/Metadatos>

Saborido, S. (2013). *Importancia de los repositorios para preservar y recuperar la información*. Recuperado el 2023, de scielo.sld.cu:

http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1029-30192015001000014

Santiago, S. (2013). *Importancia de los repositorios para preservar y recuperar la información*. Recuperado el 2023, de scielo.sld.cu/:

http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1029-30192015001000014

Souza, I. d. (2020). *rockcontent.com*. Obtenido de rockcontent.com:

<https://rockcontent.com/es/blog/php/>

Yaguas, A. (2008). *eprints.rclis.org*. Recuperado el 2023, de eprints.rclis.org:

<http://eprints.rclis.org/15472/1/La%20biblioteca%20espacio%20de%20cultura%20y%20participacion.pdf>

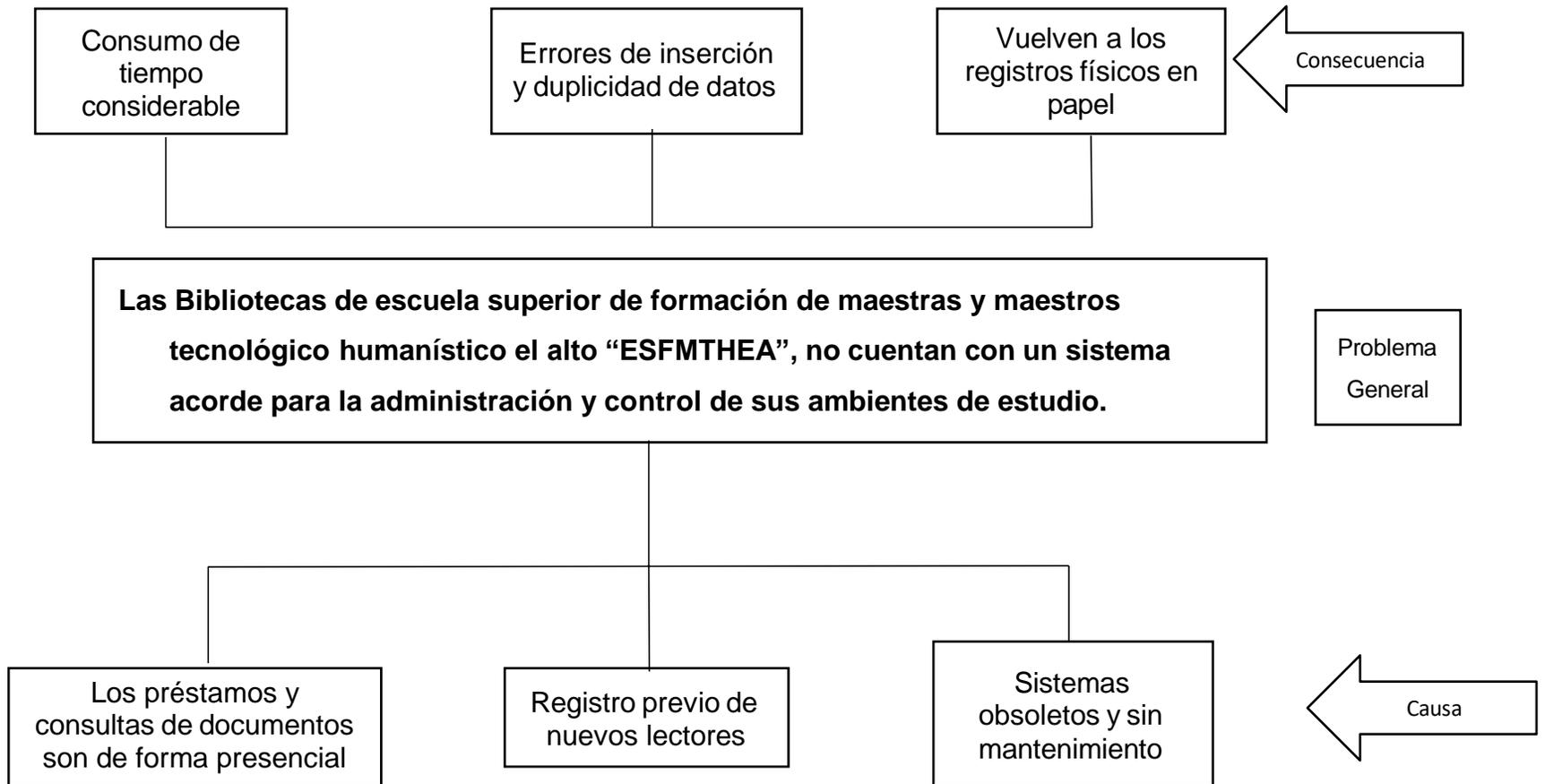
ZAPATA, A. A. (2019). *Metodologías UWE y OOHDM*. Recuperado el 2023, de studocu:

<https://www.studocu.com/es-mx/document/instituto-tecnologico-de-acapulco/fundamentos-de-ingenieria-de-software/metodologias-uwe-y-ooohdm/8566429>

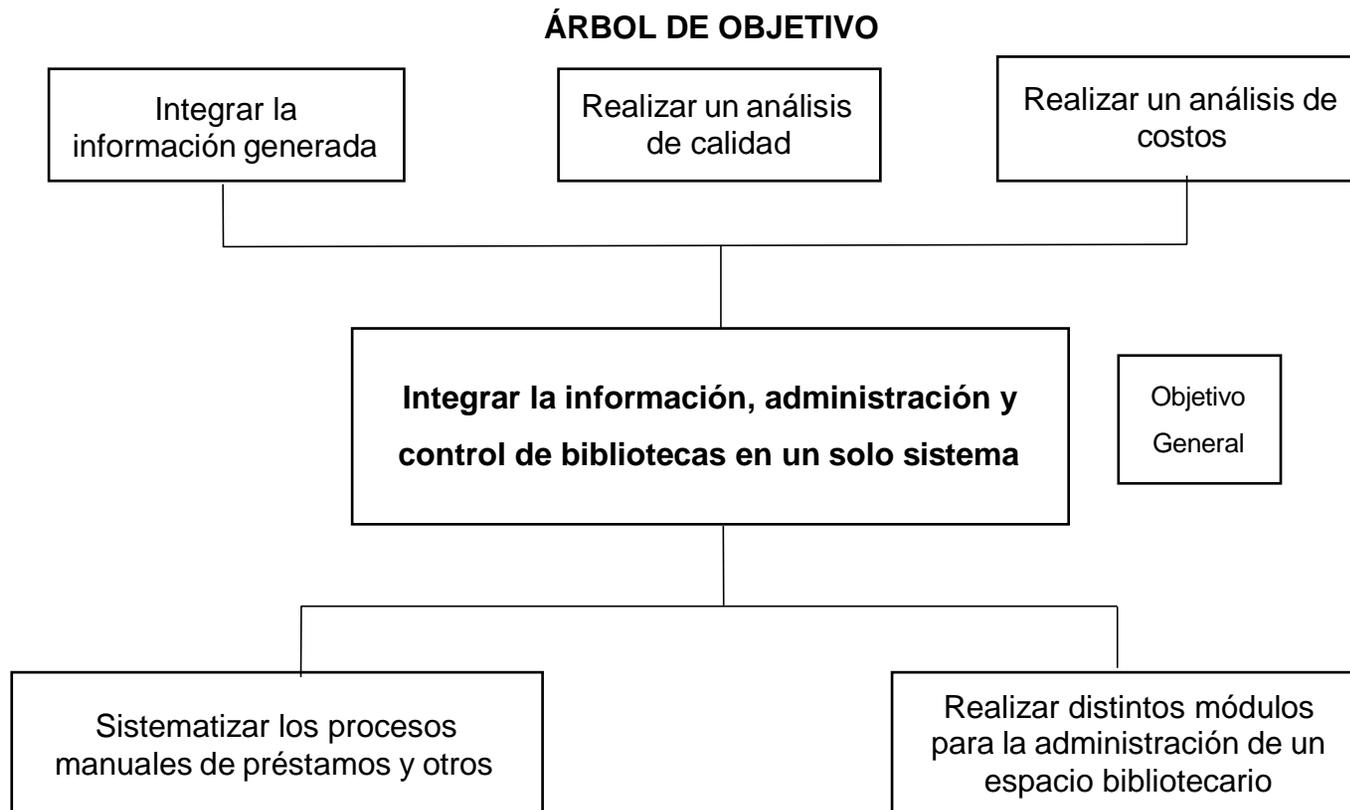
ANEXO

Anexo 1: Árbol de Problemas

ÁRBOL DE PROBLEMAS



Anexo 2: Árbol de Objetivos



Señor:
M.Sc.Ing. David Carlos Mamani Quispe
DIRECTOR DE CARRERA
INGENIERÍA DE SISTEMAS
Presente. -

El Alto, 9 de junio de 2023

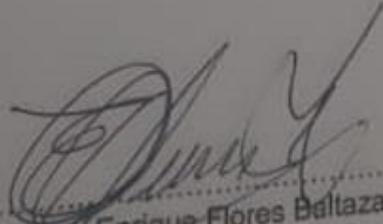
REF. AVAL DE CONFORMIDAD

Distinguido director de carrera:
Mediante la presente tengo a bien comunicarle mi conformidad del Trabajo de Grado:

TITULO: DISEÑO E IMPLEMENTACIÓN DE UNA BIBLIOTECA VIRTUAL Y
REPOSITORIO DE PROYECTOS DE GRADO Y TESIS
CASO: ESCUELA SUPERIOR DE FORMACION DE MAESTRAS Y MAESTROS
TECNOLOGICO HUMANISTICO EL ALTO "ESFMTHEA"
MODALIDAD: PROYECTO DE GRADO
Univ. Gladys Orihuela Mamani
Registro Universitario: 12000436
Cedula de Identidad: 6988992 LP

Para su defensa pública y evaluación correspondiente a la materia de Taller de Grado II, de acuerdo al reglamento vigente de la Carrera de Ingeniería de sistemas de la Universidad Pública de El Alto.

Atentamente,


M.Sc.Ing. Enrique Flores Beltazar
TUTOR METODOLÓGICO
TALLER DE GRADO II

Señor:
M.Sc.Ing. Enrique Flores Baltazar
TUTOR METODOLÓGICO
TALLER DE GRADO II
Presente. –

El Alto, 9 de junio de 2023

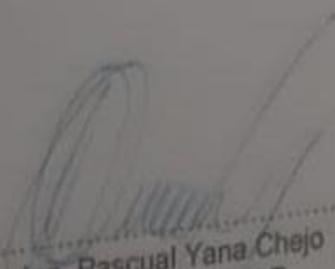
REF. AVAL DE CONFORMIDAD

Distinguido tutor metodológico:
Mediante la presente tengo a bien comunicarle mi conformidad del Trabajo de Grado:

TITULO: DISEÑO E IMPLEMENTACIÓN DE UNA BIBLIOTECA VIRTUAL Y REPOSITORIO DE PROYECTOS DE GRADO Y TESIS
CASO: ESCUELA SUPERIOR DE FORMACION DE MAESTRAS Y MAESTROS TECNOLÓGICO HUMANÍSTICO EL ALTO "ESFMTEHA"
MODALIDAD: PROYECTO DE GRADO
Univ. Gladys Orihuela Mamani
Registro Universitario: 12000436
Cedula de Identidad: 6988992 LP

Para su defensa pública y evaluación correspondiente a la materia de Taller de Grado II, de acuerdo al reglamento vigente de la Carrera de Ingeniería de sistemas de la Universidad Pública de El Alto.

Atentamente,


Ing. Pascual Yana Chejo
TUTOR REVISOR

El Alto, 9 de junio de 2023

Señor:
M.Sc.Ing. Enrique Flores Baltazar
TUTOR METODOLÓGICO
TALLER DE GRADO II
Presente. -

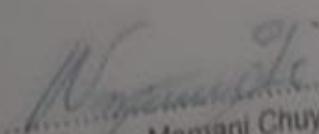
REF. AVAL DE CONFORMIDAD

Distinguido tutor metodológico:
Mediante la presente tengo a bien comunicarle mi conformidad del Trabajo de Grado:

TITULO: DISEÑO E IMPLEMENTACIÓN DE UNA BIBLIOTECA VIRTUAL Y
REPOSITORIO DE PROYECTOS DE GRADO Y TESIS
CASO: ESCUELA SUPERIOR DE FORMACION DE MAESTRAS Y MAESTROS
TECNOLOGICO HUMANISTICO EL ALTO "ESFMTHEA"
MODALIDAD: PROYECTO DE GRADO
Univ. Gladys Orihuela Mamani
Registro Universitario: 12000436
Cedula de Identidad: 6988992 LP

Para su defensa pública y evaluación correspondiente a la materia de Taller de Grado II, de acuerdo al reglamento vigente de la Carrera de Ingeniería de sistemas de la Universidad Pública de El Alto.

Atentamente,


.....
Ing. Noemy Mamani Chuyma
TUTOR ESPECIALISTA



ESCUELA SUPERIOR DE FORMACIÓN DE MAESTRAS
Y MAESTROS TECNOLÓGICO Y HUMANÍSTICO EL ALTO
Fundado el 6 de marzo de 2006 por D.S. 28625 y Ley 3441

Señor:

El Alto, 12 de junio de 2023

M.Sc. Ing. Enrique Flores Baltazar

TUTOR METODOLÓGICO

TALLER DE GRADO II

Presente. –

REF.- AVAL DE CONFORMIDAD

Distinguido tutor metodológico:

Mediante la presente tengo a bien comunicarle mi conformidad del Trabajo de Grado:

TITULO: DISEÑO E IMPLEMENTACION DE UNA BIBLIOTECA VIRTUAL Y REPOSITORIO DE PROYECTOS DE GRADO Y TESIS

CASO: ESCUELA SUPERIOR DE FORMACION DE MAESTRAS Y MAESTROS TECNOLÓGICO HUMANÍSTICO EL ALTO "ESFMTHA"

MODALIDAD: PROYECTO DE GRADO

Univ. Gladys Orihuela Mamani

Registro Universitario: 12000436

Cedula de Identidad: 6988992 LP

El siguiente sistema fue **IMPLEMENTADO** satisfactoriamente.

De tal forma cabe recalcar que el PROYECTO DISEÑO E IMPLEMENTACION DE UNA BIBLIOTECA VIRTUAL Y REPOSITORIO DE PROYECTOS DE GRADO Y TESIS satisface los requerimientos de la institución, de esta forma se dio cumplimiento de los objetivos del presente.

En cuanto certifico, en honor a la verdad, para fines consiguientes del interesado para su defensa pública y evaluación correspondiente a la materia de Taller de Grado II, de acuerdo al reglamento vigente de la Carrera de Ingeniería de sistemas de la Universidad Pública de El Alto.

Atentamente,



Alberto Croner Quiroga Pardo
M.Sc. Alberto Croner Quiroga Pardo
DIRECTOR GENERAL
E.S.F.M.T.M. EL ALTO

MANUAL DE USUARIO ADMINISTRADOR



DISEÑO E IMPLEMENTACION DE UNA BIBLIOTECA VIRTUAL Y REPOSITORIO DE PROYECTOS DE GRADO Y TESIS ESFMTHEA

INDICE

1	INTRODUCCION	4
2	OBJETIVO	4
3	REQUERIMIENTOS	4
3.1	AUTENTICACIÓN (ADMINISTRADOR)	4
3.2	PANTALLA DE INICIO (ADMINISTRADOR).....	6
3.3	ADMINISTRACIÓN DE USUARIOS	7
3.4	MODULO CARNET	8
3.5	ADMINISTRAR LIBROS	9
3.6	MODULO ADMINISTRADOR AUTOR.....	10
3.7	MODULO EDITORIAL	11
3.8	MODULO RESERVA.....	12
3.9	ADMINISTRACION DE MODULO (PRESTAMOS)	13
3.10	ADMINISTRACION MODULO REPORTES.....	14
3.11	MODULO GRAFICAS Y ESTADISTICAS	15
3.12	MODULO CATALOGACION	16
3.13	MODULO HERRAMIENTAS	16
4	INICIO DE SESION – LECTORES	17
4.1	LECTURA DE CONTENIDO LECTOR.....	18
4.2	Elemento Bibliografico	19
4.3	Elemento de talleres y actividades sociales	19

FIGURAS

<i>Figura 1. login.....</i>	<i>5</i>
<i>Figura 2 Pantalla de Inicio</i>	<i>6</i>
<i>Figura 3. Crear Usuario</i>	<i>7</i>
<i>.....</i>	<i>7</i>
<i>Figura 4 Crear Carnet</i>	<i>8</i>
<i>Figura 5. Administrar Libros</i>	<i>9</i>
<i>Figura 6 Lista de Libros Registrados.....</i>	<i>10</i>
<i>Figura 7. Lista de autores registrados.....</i>	<i>10</i>
<i>Figura 8. Lista de Editorial Registrados.....</i>	<i>11</i>
<i>Figura 9. Realizar Reserva</i>	<i>12</i>
<i>Figura 10. Lista de Prestamos</i>	<i>13</i>
<i>Figura 11. Lista de Reportes Libros.....</i>	<i>14</i>
<i>Figura 12. Lista de Reportes Usuarios.....</i>	<i>14</i>
<i>Figura 13. Gráficos y Estadísticas.....</i>	<i>15</i>
<i>Figura 14. Gráficos y Estadísticas por libros.....</i>	<i>15</i>
<i>Figura 15. Catalogo por Carreras</i>	<i>16</i>
<i>Figura 16. Respaldo Backup</i>	<i>16</i>
<i>Figura 17. Pantalla de inicio de usuarios.....</i>	<i>17</i>
<i>Figura 18. Interfaz de Acceso de Lectores</i>	<i>17</i>
<i>Figura 19. Lista de Catalogo por Carreras.....</i>	<i>18</i>
<i>Figura 20. Elemento Bibliográfico</i>	<i>19</i>
<i>Figura 21. Talleres y actividades sociales.....</i>	<i>19</i>

1 INTRODUCCION

El sistema denominado diseño e implementación de una biblioteca virtual y repositorio de proyectos de grado y tesis, fue elaborado con herramientas de desarrollo de código abierto. El sistema requiere un sistema gestor de bases de datos MySQL, Visual Studio las cuales permitirán el acceso desde diferentes puntos a través de una conexión a internet.

El sistema permite administrar los diferentes procesos que se llevan a cabo en una determinada biblioteca, la cual le permite realizar acciones como el registro de préstamos, registro de documentos bibliográficos, entre otros.

El presente manual explica paso a paso las diferentes funcionalidades del sistema y como deben ser manipulados correctamente.

2 OBJETIVO

El manual fue desarrollado con la finalidad de ayudar a los nuevos usuarios del sistema, misma que presenta una descripción detallada de los diferentes módulos y formularios que este presenta.

3 REQUERIMIENTOS

Para poder acceder al sistema en línea se requiere lo siguiente:

- Requisitos Hardware
 - Un ordenador de escritorio o dispositivo alternativo como: Laptop o tableta inteligente.
 - Conexión estable a internet
- Requisitos Software
 - Navegar web Chrome, MS Edge, Firefox Mozilla y otros

3.1 AUTENTICACIÓN (ADMINISTRADOR)

El acceso al sistema se lo hace a través de cualquier navegador web, apuntando a la siguiente dirección o enlace web.

<http://localhost/biblioinstituto.info>

El enlace anterior direccionará al usuario a la pantalla o interfaz de inicio de sesión que el sistema presenta. El formulario solicita al usuario las credenciales de acceso. En el caso de un error el sistema desplegará los mensajes respectivos indicando el problema o error cometido por el usuario.

Figura 1. login



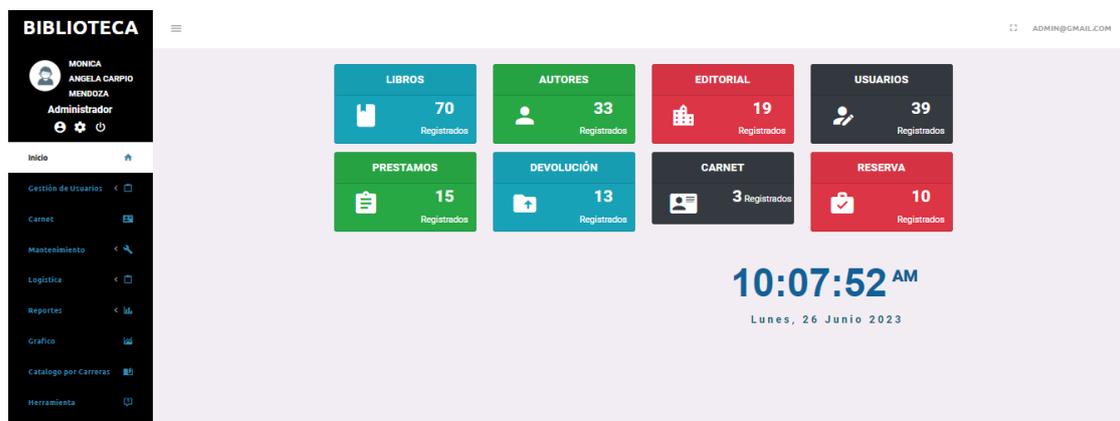
The image shows a login form with a dark blue header containing a profile icon and the text 'Login Usuario'. Below the header, the title 'INICIAR SESIÓN' is centered. The form includes input fields for 'Usuario' and 'Contraseña', a 'Mostrar contraseña' toggle, a reCAPTCHA widget with the text 'No soy un robot', and a large red 'Iniciar Sesión' button. At the bottom, there are links for '¿No Tengo una Cuenta?' (with 'REGISTRATE' and 'SALIR' buttons) and 'Me olvide la Contraseña' (with a 'RECUPERAR CONTRASEÑA' button).

Nota: elaboración propia

- Ingresar nombre de usuario
- Ingresar contraseña o clave
- Verificar recaptcha
- Botonera de inicio de sesión al servidor

3.2 PANTALLA DE INICIO (ADMINISTRADOR)

Figura 2 Pantalla de Inicio



Nota. Elaboración Propia

La pantalla de inicio o tablero, brinda un breve resumen acerca de unos determinados registros o módulos que presenta el sistema, para el caso del usuario administrador se presenta lo siguiente:

- Se trata de una etiqueta que tiene como tarea visualizar los diferentes nombres de los diferentes espacios bibliotecarios, para el caso de un usuario administrador se muestra la etiqueta “Administrador”.
- Muestra el logotipo de una determinada biblioteca. Para el caso de usuarios administradores se muestra una imagen por defecto, en este caso el logotipo de la universidad.
- Menú de navegación, muestra una lista de navegación, mismas que direccionan a diferentes módulos que presenta el sistema. Esta lista de navegación puede variar dependiendo del tipo de usuario que esté usando el sistema.
- Etiqueta de usuario, indica el nombre del usuario con sesión activa. De la misma forma despliega un menú de opciones:
 - Cambiar Contraseña: esta opción permite modificar la contraseña del usuario.
 - Cerrar Sesión: permite el cierre de la sesión del usuario.
- El último espacio hace referencia al cuerpo del sistema, en ella se irán mostrando las diferentes listas, formularios y otros; Que el sistema necesita para interactuar con el usuario. En el ejemplo se puede apreciar

el tablero del usuario administrador, que visualiza lo siguiente:

- Listado de libros más consultados por los lectores.
- Resumen de los movimientos de préstamos por mes, realizados por cada biblioteca que haga uso del sistema

3.3 ADMINISTRACIÓN DE USUARIOS

- Botonera de agregación de nuevo usuario, despliega el formulario de registro para agregar usuarios del tipo bibliotecario.
- Selector de Bibliotecas, permite listar los diferentes registros de usuarios de una determinada biblioteca.

Figura 3. Crear Usuario

Usuario - Detalle

Código: _____

Nombres: _____ Apellido Paterno: _____ Apellido Materno: _____

C.I.: _____ Fecha Nacimiento: mm / dd / yyyy Edad: _____ Tipo Usuario : Administrador

Correo: _____ Foto Usuario: No file selected.

Clave: _____ Estado:

Nota. Elaboración Propia

Agregar los siguientes datos: Nombres, apellido paterno, apellido Materno, C.I., fecha de nacimiento, tipo de usuario

3.4 MODULO CARNET

En este módulo se creará una credencial o carnet para cada usuario de la escuela de formación de maestros y maestras por lo que ellos podrán ingresar libremente a instalaciones de la biblioteca.

Figura 4 Crear Carnet

USUARIO,
GRADO,
SECCIÓN,
TURNO, F. DE
INCRIPCION,
FECHA DE
EXPIRACION

Nota. Elaboración Propia

- Se trata de una etiqueta que tiene como tarea visualizar los diferentes nombres de los diferentes espacios de usuarios.
- Se determinará un código para cada estudiante.
- Menú de navegación, muestra una lista de navegación, mismas que direccionan a diferentes módulos que presenta el sistema. Esta lista de navegación puede variar dependiendo del tipo de usuario que esté usando el sistema.
- Etiqueta de usuario, indica el nombre del usuario con sesión activa. De la misma forma despliega un menú de opciones.

3.5 ADMINISTRAR LIBROS

Figura 5. Administrar Libros

Libro - Detalle

Codigo :

Titulo Libro: Autor : ANA GONZALES (BOLIVIA) Editorial : EDITORIAL LOSADA (ARGE) Categoria : EDUCACION MUSICAL

Tipo Libro : Fisico Pais: Portada Libro: Seleccionar archivo | Nin...Jec

Fecha publicacion: dd/mm/aaaa Stock: Stock Disponible: Costo Perdida:

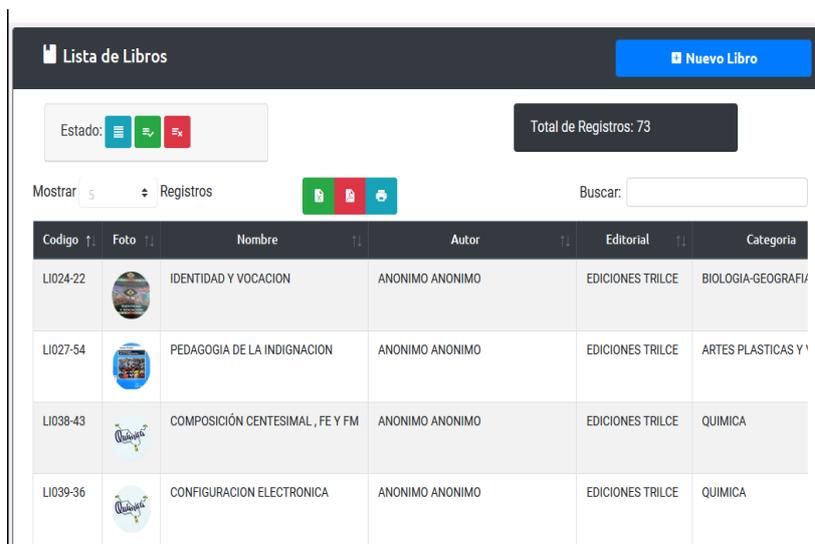
Descripcion : Estado:

Cancelar Guardar

Nota. Elaboración Propia

- En la administración de libros; este módulo se agregará nuevos libros, proyectos de grado, profocom etc. De igual manera podremos llenar los datos correspondientes para cada libro.
- Se trata de una etiqueta que tiene como tarea visualizar los diferentes nombres de los diferentes espacios bibliotecarios, para el caso de un usuario administrador se muestra la etiqueta “Administrador”.
- Muestra el logotipo de una determinada biblioteca. Para el caso de usuarios administradores se muestra una imagen por defecto, en este caso el logotipo de la universidad.
- Menú de navegación, muestra una lista de navegación, mismas que direccionan a diferentes módulos que presenta el sistema. Esta lista de navegación puede variar dependiendo del tipo de usuario que esté usando el sistema.
- Etiqueta de usuario, indica el nombre del usuario con sesión activa. De la misma forma despliega un menú de opciones.

Figura 6 Lista de Libros Registrados



Codigo	Foto	Nombre	Autor	Editorial	Categoria
LI024-22		IDENTIDAD Y VOCACION	ANONIMO ANONIMO	EDICIONES TRILCE	BIOLOGIA-GEOGRAFIA
LI027-54		PEDAGOGIA DE LA INDIGNACION	ANONIMO ANONIMO	EDICIONES TRILCE	ARTES PLASTICAS Y
LI038-43		COMPOSICIÓN CENTESIMAL , FE Y FM	ANONIMO ANONIMO	EDICIONES TRILCE	QUIMICA
LI039-36		CONFIGURACION ELECTRONICA	ANONIMO ANONIMO	EDICIONES TRILCE	QUIMICA

Nota. Elaboración Propia

Por otro lado al registrar un libros esto nos llevara a un formulario de vista de cada librop que este en la escuela de maestras y maestros.

3.6 MODULO ADMINISTRADOR AUTOR

En lo que se refiere al módulo autor esto nos conlleva a agregar un nuevo autor también procedamos la llegar sus datos correspondientes.

Figura 7. Lista de autores registrados



Autor - Detalle

Código:

Por defecto

Nombres: Apellidos:

Agregar Datos

Pais: Fecha Nacimiento:

Agregar Datos

Estado:

Cancelar Guardar

Nota. Elaboración Propia

- Se trata de una etiqueta que tiene como tarea visualizar los diferentes nombres de los diferentes espacios bibliotecarios, para el caso de un usuario administrador se muestra la etiqueta “Administrador”.
- Muestra el logotipo de una determinada biblioteca. Para el caso de usuarios administradores se muestra una imagen por defecto.
- Menú de navegación, muestra una lista de navegación, mismas que direccionan a diferentes módulos que presenta el sistema. Esta lista de navegación puede variar dependiendo del tipo de usuario que esté usando el sistema.
- Etiqueta de usuario, indica el nombre del usuario con sesión activa. De la misma forma despliega un menú de opciones.

3.7 MODULO EDITORIAL

Figura 8. Lista de Editorial Registrados

The image shows a web form titled "Editorial - Detalle". The form includes the following elements:

- Codigo:** A dropdown menu with the selected option "Por defecto".
- Nombres:** A text input field. A red box labeled "Agregar Datos" with an arrow points to this field.
- Pais:** A text input field.
- Descripcion:** A text area.
- Estado:** A checkbox.
- Buttons:** "Cancelar" (red) and "Guardar" (blue) buttons at the bottom.

Nota. (Elaboración propia)

En este modulo se encarga de crear nuevas editoriales de libros: mediante combobox Esta pestaña es muy importante para crear un nuevo libro y tener ya el dato correspondiente.

3.8 MODULO RESERVA

Figura 9. Realizar Reserva

Reserva - Detalle X

Codigo: _____

Usuario : GADYS ORIHUELA MAMANI v

Libro 500 ACTIVIDADES PARA EL CURRICULO v

Cantidad: _____

Fecha Reserva: mm/dd/yyyy 📅 Fecha Limite: mm/dd/yyyy 📅

Prestado Libro: Estado:

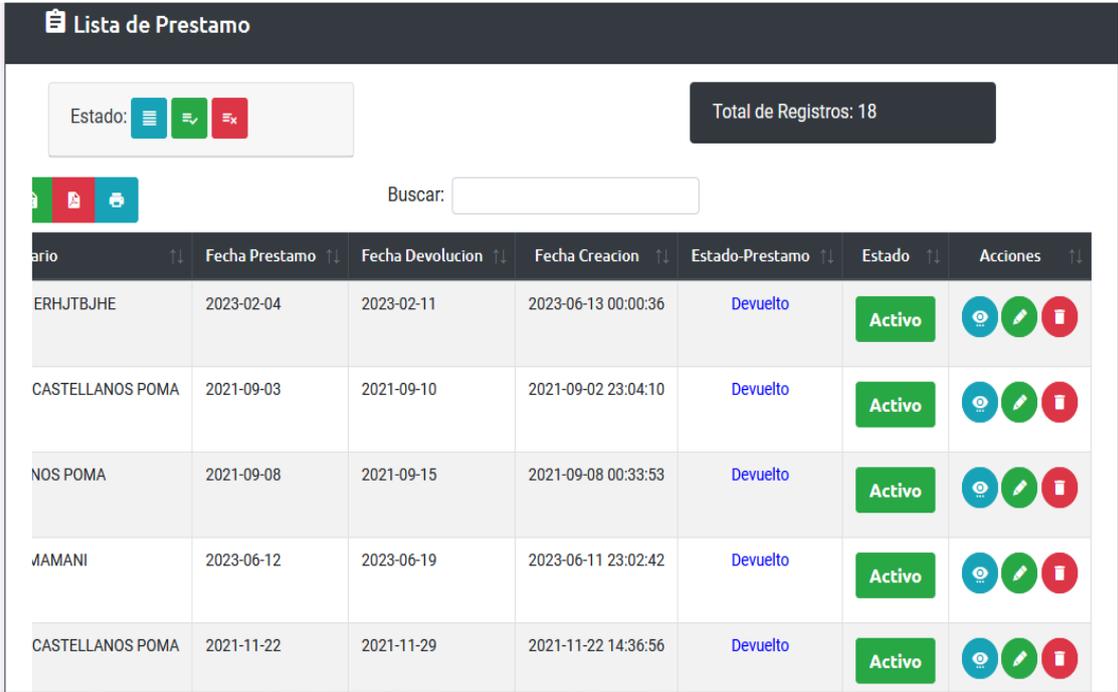
Cancelar **Guardar**

Nota. Elaboración Propia

Este módulo es para la verificación de las reservas pedidas por los usuarios (docentes y estudiantes) esta herramienta es muy importante indispensable este modulo es pieza fundamental del modulo préstamo.

3.9 ADMINISTRACION DE MODULO (PRESTAMOS)

Figura 10. Lista de Prestamos



The screenshot shows a web interface titled "Lista de Prestamo". At the top, there is a header with a menu icon and the title. Below the header, there is a filter section with "Estado:" and three buttons (list, refresh, delete). To the right, a dark box displays "Total de Registros: 18". Below this is a search bar with "Buscar:" and an input field. The main content is a table with the following columns: "Ario", "Fecha Prestamo", "Fecha Devolucion", "Fecha Creacion", "Estado-Prestamo", "Estado", and "Acciones". The table contains five rows of data, each with a borrower name, two dates, a creation timestamp, a status ("Devuelto"), and a green "Activo" button. The "Acciones" column contains three icons: a blue eye, a green pencil, and a red trash can.

Ario	Fecha Prestamo	Fecha Devolucion	Fecha Creacion	Estado-Prestamo	Estado	Acciones
ERHJTBJHE	2023-02-04	2023-02-11	2023-06-13 00:00:36	Devuelto	Activo	  
CASTELLANOS POMA	2021-09-03	2021-09-10	2021-09-02 23:04:10	Devuelto	Activo	  
NOS POMA	2021-09-08	2021-09-15	2021-09-08 00:33:53	Devuelto	Activo	  
MAMANI	2023-06-12	2023-06-19	2023-06-11 23:02:42	Devuelto	Activo	  
CASTELLANOS POMA	2021-11-22	2021-11-29	2021-11-22 14:36:56	Devuelto	Activo	  

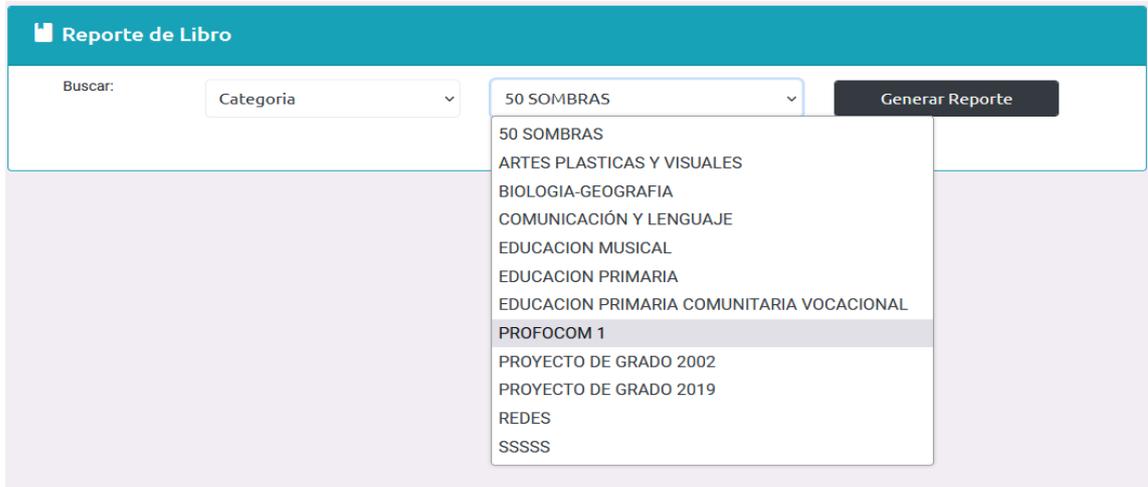
Nota. Elaboración Propia

La interfaz de movimientos cuenta con tres secciones que visualizan diferentes registros.

- Etiqueta de interfaz, indica el nombre de la interfaz en la que el usuario se encuentra.
- Selector de vistas, permite navegar por las tres diferentes vistas que presentala interfaz.
- Botoneras de acción, para el caso de la interfaz de solicitudes se presentanlas siguientes botoneras:
 - **Confirmar:** Permite confirmar una solicitud de préstamos entrante porparte del usuario.
 - **Descartar:** Permite descartar una solicitud de préstamo

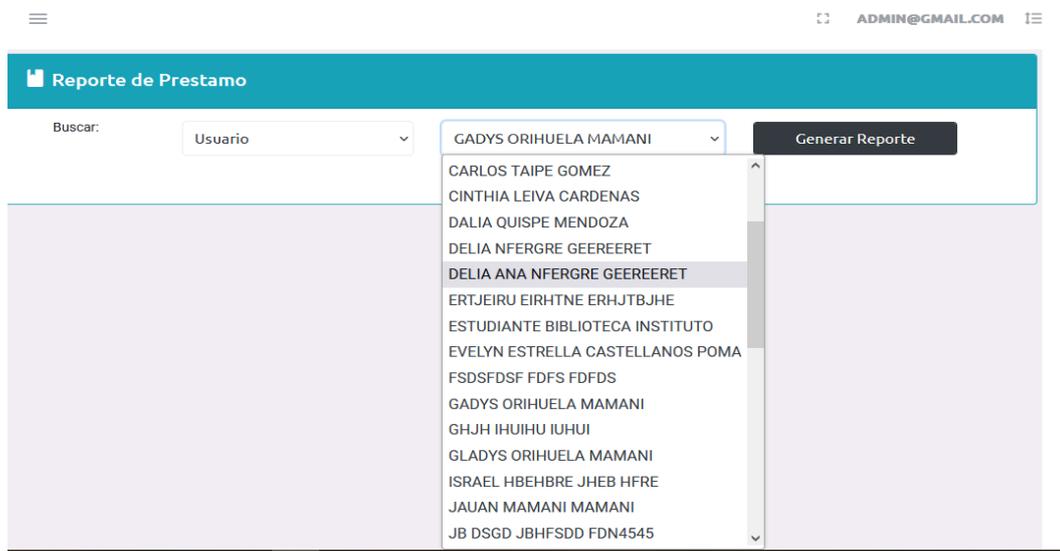
3.10 ADMINISTRACION MODULO REPORTES

Figura 11. Lista de Reportes Libros



Nota. Elaboración Propia

Figura 12. Lista de Reportes Usuarios

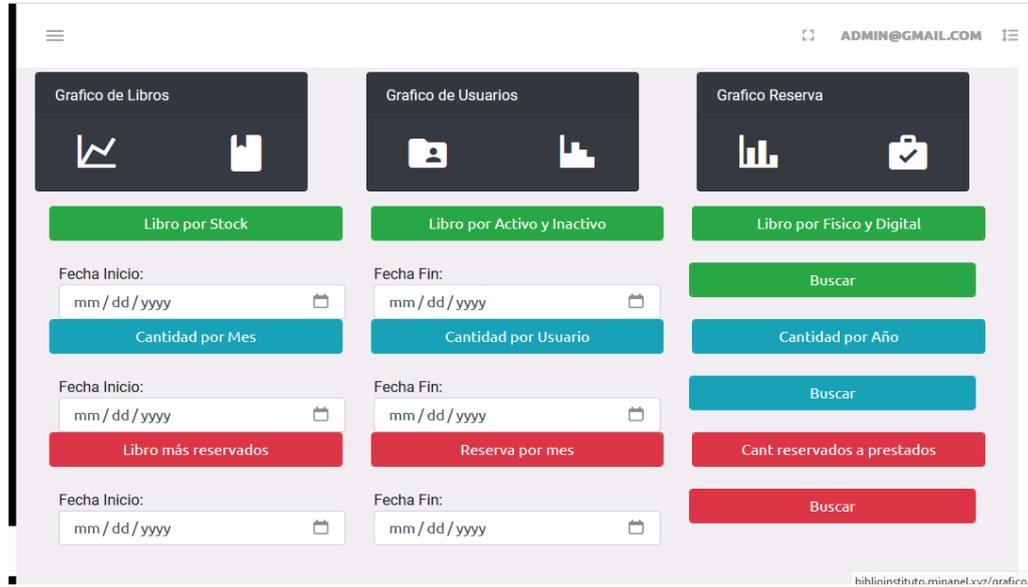


Nota. Elaboración Propia

Modulo Reportes en este módulo se podrá imprimir los distintos reportes que existen en el sistema (libro, usuario, reserva, préstamo)

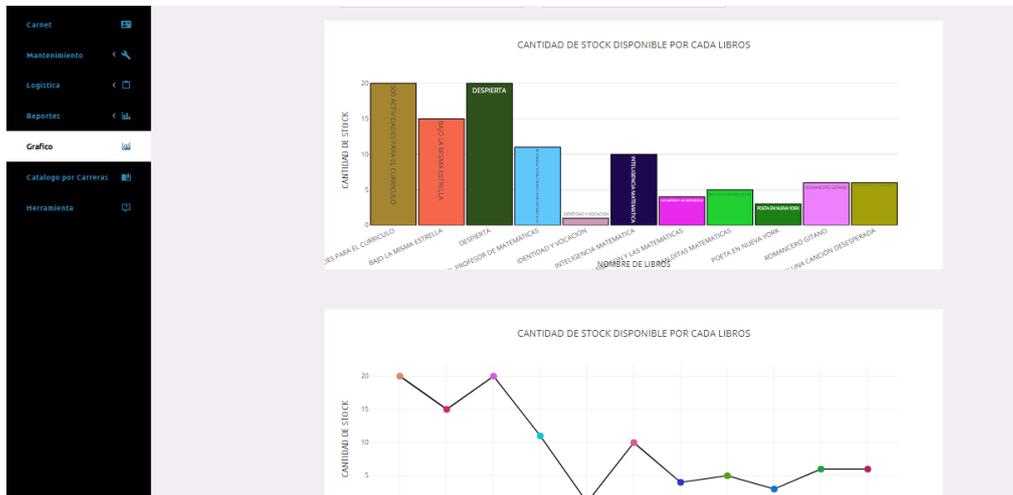
3.11 MODULO GRAFICAS Y ESTADISTICAS

Figura 13. Gráficos y Estadísticas



Nota. Elaboración Propia

Figura 14. Gráficos y Estadísticas por libros



Nota. Elaboración Propia

En este módulo se visualizará las gráficas de libros lectores etc.

3.12 MODULO CATALOGACION

Figura 15. Catalogo por Carreras

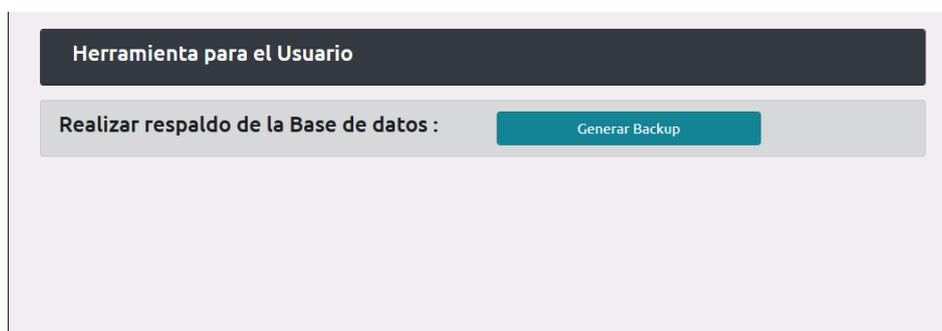


Nota. Elaboración Propia

Modulo donde podremos ver la catalogación de todas las carreras; podremos visualizar los textos de las carreras también podremos ver los datos de los libros por carrera.

3.13 MODULO HERRAMIENTAS

Figura 16. Respaldo Backup



Nota. Elaboración Propia

Copias de seguridad donde el administrador podrá realizar las copias periódicamente recomendada cada 24 horas.

4 INICIO DE SESION – LECTORES

Figura 17. Pantalla de inicio de usuarios



Nota. Elaboración Propia

Figura 18. Interfaz de Acceso de Lectores



Nota. Elaboración propia

- Imagen de código captcha
- Identificador, dependiendo del tipo de lector que se trate, el sistema permitidos tipos de registros

- C.I., para estudiantes universitarios.
- Cedula de Identidad, para docentes.
- Campo de confirmación de código captcha.
- Botonera de iniciar sesión o ingreso, por parte de los lectores.

4.1 LECTURA DE CONTENIDO LECTOR

Una vez ingresado, el sistema despliega la siguiente pantalla donde cada lector podrá visualizar el catálogo bibliográfico de la unidad bibliotecaria que le corresponda.

Figura 19. Lista de Catalogo por Carreras

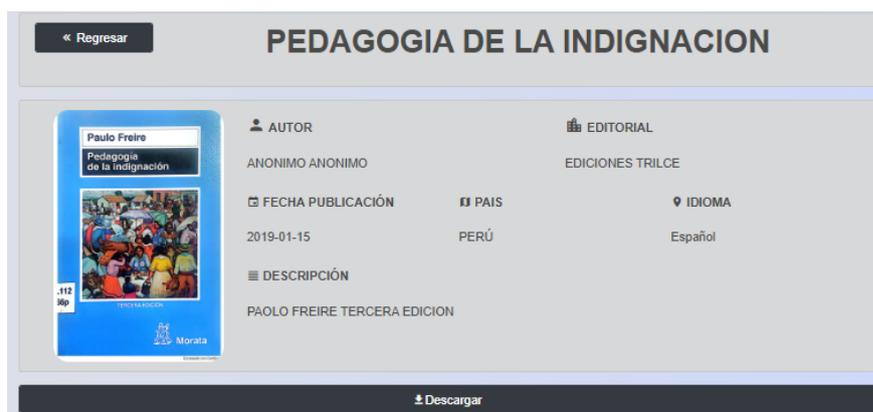


Nota. Elaboración Propia

- Filtro, permite encontrar un determinado elemento bibliográfico, esta realiza la búsqueda a través de los siguientes campos: Título, Autor y ISBN.
- Switch de consulta externa, permite extender la búsqueda a bibliotecas externar que hagan uso del sistema, estas pueden ser bibliotecas de Áreas y Carreras.
- Panel informativo del lector, visualiza la información acerca de un lector.
- Selector de páginas, de contenido, permite cambiar de página en la lista de catálogo.
- Lista de catálogo, es el espacio donde se irán listando los diferentes ejemplares bibliográficos que posee una determinada biblioteca.

4.2 ELEMENTO BIBLIOGRAFICO

Figura 20. Elemento Bibliográfico



Nota. Elaboración Propia

Representa un documento o libro que poseen las diferentes unidades bibliotecarias, este elemento se encuentra dividido en cuatro secciones que describen un poco el documento bibliotecario que representan:

- Título.
- Autor.
- Tipo de documento y año de publicación.
- Nombre de la biblioteca que posee el documento bibliotecario representado

4.3 ELEMENTO DE TALLERES Y ACTIVIDADES SOCIALES

Figura 21. Talleres y actividades sociales



Nota. Elaboración Propia

En este elemento el usuario podrá visualizar los talleres que sean programados como también podrán estar al tanto de las actividades sociales etc.