

UNIVERSIDAD LA SALLE

LICENCIATURA EN INGENIERÍA DE SISTEMAS



**“SISTEMA DE GESTIÓN DE CONTENIDO EMPLEANDO
TÉCNICAS DE ENMASCARAMIENTO DE DATOS PARA LA
ACADEMIA BOLIVIANA DE HISTORIA MILITAR”**

**MODALIDAD DE TITULACIÓN: PROYECTO DE GRADO
POSTULANTE: JOSÉ ANTONIO AGUILAR FERNANDEZ
TUTOR: ING. OSAMU MANUEL YOKOSAKI PEÑARANDA**

LA PAZ-BOLIVIA

2025

DEDICATORIA

Este Trabajo está dedicado a:

A Dios, por concederme la vida y la sabiduría para concluir este proyecto.

A mi familia, por brindarme su apoyo incondicional en cada etapa de este proceso.

La Universidad La Salle Bolivia, por brindarme la oportunidad de formarme académicamente y por inculcarme los valores lasallistas que guían mi vida profesional y personal.

AGRADECIMIENTOS

A: Dios, por darme la vida, la fortaleza en los momentos de cansancio y desaliento, y por concederme la sabiduría y el entendimiento necesarios para llevar a cabo este proyecto de grado.

A: Mi Familia, por brindarme su apoyo incondicional en cada etapa de este proceso, especialmente en los momentos más difíciles, cuando su aliento y comprensión fueron fundamentales para seguir adelante.

A: La Universidad La Salle Bolivia, por brindarme la oportunidad de formarme académicamente y por inculcarme los valores lasallistas que guían mi vida profesional y personal, agradezco profundamente a mis tutores por su valioso tiempo, orientación académica y apoyo incondicional, cada uno de sus aportes fue fundamental e indispensable para la culminación exitosa de este proyecto.

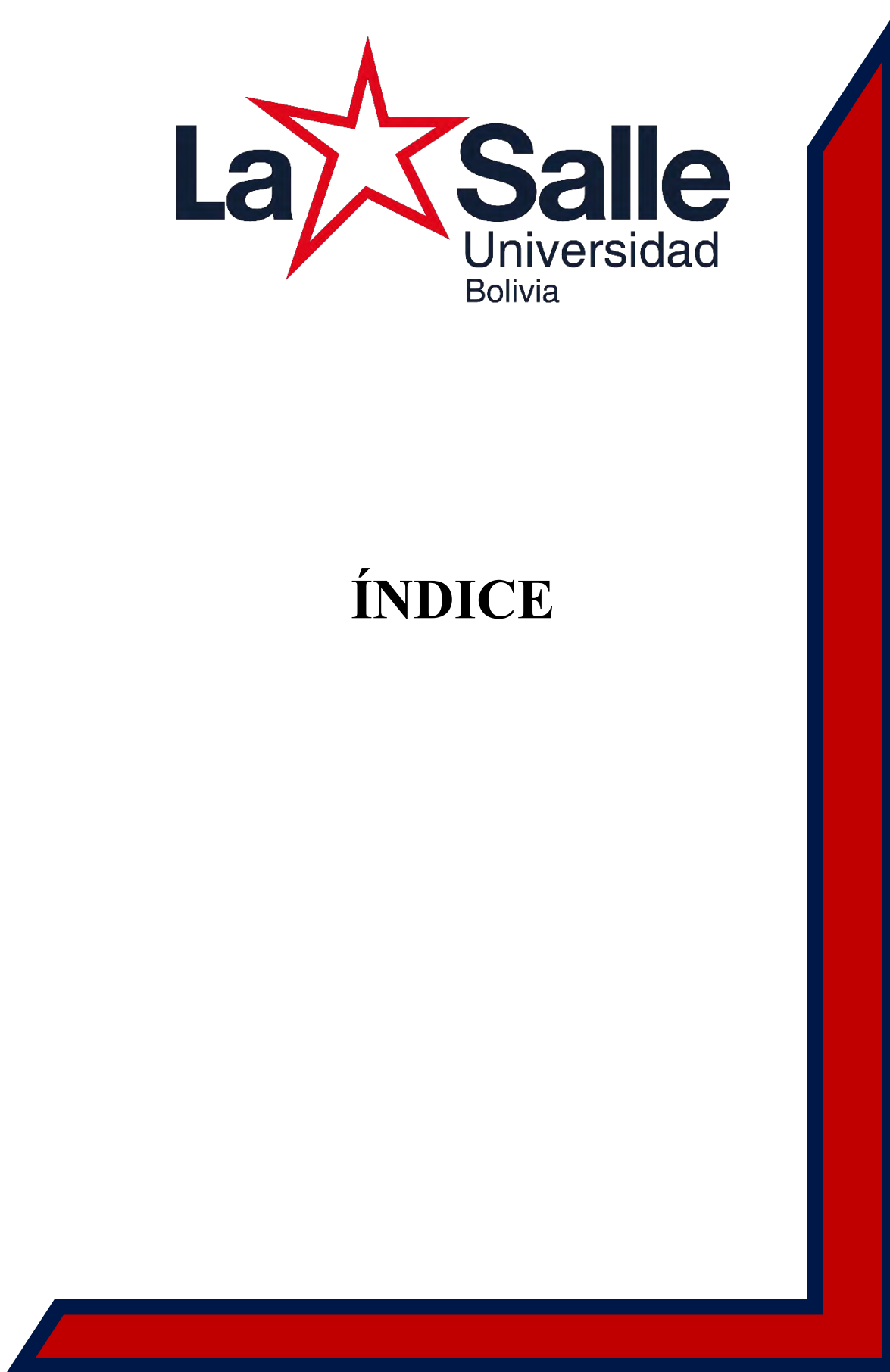
A: La Academia Boliviana de Historia Militar, por su invaluable labor en la preservación, difusión y enseñanza de la historia militar del país, y por inspirarme a valorar nuestras raíces históricas y culturales, su compromiso con la formación intelectual y cívica ha sido una fuente de motivación para el desarrollo de este trabajo.

A: La empresa Webstyle Studios, por haberme brindado la oportunidad de aprender y fortalecer las bases fundamentales de la programación, así como por enseñarme metodologías de trabajo en equipo y ofrecerme una formación sólida que ha sido clave en mi desarrollo profesional.

A: La empresa Star Tigo, por haber contribuido de manera significativa a mi formación profesional, brindándome la oportunidad de desarrollar habilidades en el manejo de la comunicación, la oratoria con clientes y el liderazgo de equipos de trabajo, competencias que han sido fundamentales en mi crecimiento personal y laboral.



ÍNDICE



ÍNDICE DE CONTENIDOS

CAPITULO 1. GENERALIDADES

1.1.	INTRODUCCIÓN	1
1.2.	ANTECEDENTES	2
1.2.1.	Antecedentes académicos	2
1.2.2.	Antecedentes institucionales	4
1.2.2.1.	Misión de la institución.....	5
1.2.2.2.	Visión de la institución	5
1.3.	PLANTEAMIENTO DEL PROBLEMA	5
1.3.1.	Identificación del problema	6
1.3.2.	Formulación del problema.....	6
1.4.	OBJETIVOS	6
1.4.1.	Objetivo general	7
1.4.2.	Objetivos específicos	7
1.5.	JUSTIFICACIONES	7
1.5.1.	Justificación social	7
1.5.2.	Justificación económica.....	8
1.5.3.	Justificación técnica	8
1.6.	ALCANCES	8
1.6.1.	Alcance temático.....	8
1.6.2.	Alcance geográfico	9
1.6.3.	Alcance temporal	9
1.7.	LIMITES.....	9

CAPITULO 2. MARCO TEÓRICO

2.1.	INTRODUCCIÓN	10
2.2.	TÉCNICAS DE RECOLECCIÓN DE DATOS	10
2.2.1.	La técnica de entrevista.....	11

2.2.1.1. Tipos de entrevistas.....	11
2.2.2. Investigación documental	12
2.2.2.1. Clasificación de Documentos	12
2.3. INGENIERÍA DE SOFTWARE	13
2.3.1. Actividades y procesos	13
2.3.2. Especificación del software	13
2.3.3. Diseño e implementación del software.....	14
2.3.4. Validación del software.....	16
2.3.5. Evolución del software	17
2.4. METODOLOGÍA EXTREME PROGRAMMING (XP)	18
2.4.1. Herramientas de la metodología XP	18
2.4.1.1. Historias de usuario.....	18
2.4.1.2. Tareas de ingenierías.....	19
2.4.1.3. Pruebas de aceptación	20
2.4.1.4. Tarjetas CRC (Clase – Responsabilidades - Colaboradores).....	21
2.4.2. Roles de la metodología XP	22
2.4.3. Fases de la metodología XP	22
2.4.3.1. Planificación.....	22
2.4.3.2. Diseño.....	23
2.4.3.3. Codificación	23
2.4.3.4. Pruebas.....	24
2.5. ANÁLISIS Y DISEÑO DE SISTEMAS.....	24
2.5.1. Tipos de sistemas	25
2.5.2. Integración de las tecnologías en sistemas	26
2.5.3. Roles del analista de sistemas.....	26
2.5.4. El ciclo de vida del desarrollo de sistemas.....	27
2.6. DISEÑO DE PROCESOS	28
2.6.1. Pasos para un diseño de procesos	29

2.7.	LENGUAJE UNIFICADO DE MODELADO (UML)	29
2.7.1.	Definición de UML.....	29
2.7.2.	Tipos de diagramas UML	30
2.7.3.	Diagrama de clases.....	31
2.7.4.	Diagrama de casos de uso	33
2.8.	MODELOS DE BASES DE DATOS	35
2.8.1.	Modelo Relacional.....	35
2.9.	INTERFAZ DE USUARIO	36
2.9.1.	Definición de la interfaz de usuario (UI)	36
2.9.2.	Características de la interfaz de usuario (UI)	36
2.9.3.	Tipos de interfaces de usuario.....	37
2.10.	FRAMEWORK DE DESARROLLO	37
2.10.1.	Laravel	38
2.11.	LENGUAJES Y HERRAMIENTAS DE PROGRAMACIÓN	38
2.11.1.	HTML.....	38
2.11.2.	CSS	39
2.11.3.	JAVASCRIPT.....	39
2.11.4.	PHP	39
2.11.5.	MySQL	39
2.11.6.	Laragon.....	41
2.11.7.	Docker.....	42
2.12.	SEGURIDAD INFORMÁTICA.....	43
2.12.1.	Enmascaramiento de datos.....	44
2.12.1.1.	Tipos de enmascaramiento de datos.....	44
CAPITULO 3. MARCO APLICATIVO		
3.1.	INTRODUCCIÓN	45
3.2.	PLANIFICACIÓN DEL MARCO APLICATIVO.....	46
3.2.1.	Entrevistas con el cliente	46

3.2.1.1. Entrevista general	46
3.2.1.2. Descripción de necesidades	47
3.2.1.3. Propuesta del programador	48
3.2.2. Requerimientos funcionales y no funcionales.....	48
3.2.3. Asignación de roles del proyecto	49
3.2.4. Historias de usuario.....	49
3.2.5. Plan de entregas del proyecto	53
3.3. DISEÑO DEL MARCO APLICATIVO	54
3.3.1. Diagrama de bases de datos	54
3.3.2. Diagrama de clases del sistema	55
3.3.3. Diagrama de casos de uso del sistema	59
3.3.4. Descripción del apoyo a las actividades institucionales	62
3.4. CODIFICACIÓN DEL MARCO APLICATIVO.....	63
3.4.1. Primera Iteración.....	63
3.4.1.1. Tareas de Ingeniería primera iteración	64
3.4.1.2. Descripción de tareas de ingeniería	64
3.4.1.3. Pruebas de aceptación	67
3.4.1.4. Descripción de pruebas de aceptación	67
3.4.1.5. Capturas de pantalla primera iteración	70
3.4.1.6. Descripción de resultados primera iteración	72
3.4.2. Segunda Iteración	72
3.4.2.1. Tareas de ingeniería segunda iteración.....	72
3.4.2.2. Descripción de las tareas de ingeniería	73
3.4.2.3. Pruebas de aceptación	75
3.4.2.4. Descripción de pruebas de aceptación	75
3.4.2.5. Capturas de pantalla segunda iteración.....	79
3.4.2.6. Descripción de resultados segunda iteración	82
3.4.3. Tercera Iteración	82

3.4.3.1. Tareas de Ingeniería tercera iteración	82
3.4.3.2. Descripción de las tareas de ingeniería	83
3.4.3.3. Pruebas de aceptación	84
3.4.3.4. Descripción de pruebas de aceptación	85
3.4.3.5. Capturas de pantalla tercera iteración	87
3.4.3.6. Descripción de resultados tercera iteración	89
3.4.4. Cuarta Iteración.....	89
3.4.4.1. Técnica de enmascaramiento de datos	89
3.4.4.2. Técnica de cifrado / encriptación.....	89
3.4.4.3. Técnica de hashing.....	90
3.4.4.4. Técnica de tokenización.....	90
3.4.4.5. Desarrollo del Algoritmo de enmascaramiento.....	91
3.4.4.6. Implementación del algoritmo.....	92
3.4.4.7. Capturas de pantalla cuarta iteración.....	98
3.4.4.8. Descripción de resultados cuarta iteración	100
3.4.5. Pruebas unitarias	102
3.4.5.1. Prueba unitaria para el módulo de colección de presidentes.....	102
3.4.5.2. Prueba unitaria para el módulo de blogs	105
3.4.5.3. Prueba unitaria para el módulo de eventos	107
3.4.5.4. Conclusión de las pruebas unitarias	110
3.5. EVALUACIÓN TÉCNICA.....	111
3.5.1. Evaluación técnica.....	111
3.5.1.1. Funcionalidad.....	111
3.5.1.2. Usabilidad.....	115
3.5.1.3. Mantenibilidad	116
3.5.1.4. Portabilidad	117
3.5.1.5. Calidad Total.....	118
3.6. EVALUACIÓN ECONÓMICA.....	119

3.6.1.	Costos de estudio	119
3.6.2.	Calculamos (KLOC)	119
3.6.3.	Calcular esfuerzo (PM).....	122
3.6.4.	Tiempo de desarrollo (TDEV)	122
3.6.5.	Calcular costo de software.....	123
3.6.6.	Calcular total del proyecto	123

CAPITULO 4. CONCLUSIONES Y RECOMENDACIONES

4.1.	CONCLUSIONES	124
4.2.	RECOMENDACIONES	125

BIBLIOGRAFÍA

ANEXOS

ÍNDICE DE FIGURAS

Figura 1: El proceso de la ingeniería de requerimientos.	14
Figura 2: El modelo general del proceso de diseño	15
Figura 3: El proceso de depuración	16
Figura 4: El proceso de pruebas.....	17
Figura 5: Evolución del sistema	18
Figura 6: Las siete fases del ciclo de desarrollo de sistemas	27
Figura 7: Clasificación de los diagramas UML	30
Figura 8: Ejemplo de una clase.....	31
Figura 9: Ejemplo de relación empresa - empleado	32
Figura 10: Ejemplo de diagrama de clases	32
Figura 11: Representación de un actor	33
Figura 12: Representación de un caso de uso	34
Figura 13: Ejemplo de relación actor - caso de uso	34
Figura 14: Ejemplo de diagrama de casos de uso.....	35
Figura 15: Ejemplo de un modelo relacional	36
Figura 16: Interfaz de inicio de Laragon.....	41
Figura 17: Interfaz de Inicio de Docker.....	42
Figura 18: Funcionamiento y fases de la metodología XP.....	46
Figura 19: Diagrama de bases datos del sistema.....	55
Figura 20: Diagrama de clases para el usuario administrador	56
Figura 21: Diagrama de clases para el usuario editor.....	57
Figura 22: Diagrama de clases para el usuario afiliado	58
Figura 23: Diagrama de casos de uso de nivel 0	59
Figura 24: Diagrama de casos de uso para el usuario administrador	60
Figura 25: Diagrama de casos de uso para el usuario editor	61
Figura 26: Diagrama de casos de uso para el usuario afiliado.....	62
Figura 27: Mockup de la pantalla de acceso al sistema.....	69
Figura 28: Mockup del formulario y dashboard de usuarios	69
Figura 29: Pantalla de acceso al sistema.....	70
Figura 30: Llenando el formulario con las credenciales	70
Figura 31: Pantalla de formulario y permisos de usuario	71
Figura 32: Pantalla de gestión de usuarios.....	71
Figura 33: Mockup del formulario y dashboard de colección de presidentes	77

Figura 34: Mockup del formulario y dashboard de visitas.....	78
Figura 35: Mockup del formulario y dashboard de eventos	78
Figura 36: Pantalla gestión de colección de presidentes	79
Figura 37: Pantalla de formulario de colección de presidentes	79
Figura 38: Pantalla para gestión de visitas	80
Figura 39: Pantalla de formulario para visitas	80
Figura 40: Pantalla para gestión de eventos.....	81
Figura 41: Pantalla de formulario para eventos.....	81
Figura 42: Mockup del formulario y dashboard de noticias	86
Figura 43: Mockup del formulario y dashboard de blogs	86
Figura 44: Pantalla para gestión de noticias.....	87
Figura 45: Pantalla de formulario para noticias.....	87
Figura 46: Pantalla para gestión de blogs	88
Figura 47: Pantalla de formulario para blogs.....	88
Figura 48: Algoritmo de enmascaramiento de datos	92
Figura 49: Declarando funciones.....	92
Figura 50: Código algoritmo 1	93
Figura 51: Código de algoritmo 2	94
Figura 52: Código de algoritmo 3	95
Figura 53: Código de algoritmo 4	96
Figura 54: Código de algoritmo 5	98
Figura 55: Algoritmo aplicado en gestión de usuarios	98
Figura 56: Algoritmo aplicado en gestión de colección de presidentes	99
Figura 57: Algoritmo aplicado en gestión de blogs.....	99
Figura 58: Algoritmo aplicado en gestión de eventos.....	99
Figura 59: Algoritmo aplicado en gestión de noticias.....	100
Figura 60: Algoritmo aplicado en gestión de visitas	100
Figura 61: Creación del registro de presidente	102
Figura 62: Tabla de visualización de registro de presidentes.	103
Figura 63: Primera vista del sitio web, mostrando una tarjeta de presentación	103
Figura 64: Segunda vista del sitio web, mostrando una venta modal	104
Figura 65: Presentación de la fotografía del presidente seleccionado.....	104
Figura 66: Vista de la información completa del presidente seleccionado	104
Figura 67: Creación del registro de blog.....	105
Figura 68: Tabla de visualización de registro de blogs.....	106

Figura 69: Primera vista del sitio web, mostrando	106
Figura 70: Presentación de la fotografía y título del blog seleccionado.....	107
Figura 71: Vista de la información completa del blog seleccionado.....	107
Figura 72: Creación del registro de eventos.....	108
Figura 73: Tabla de visualización de registro de eventos.....	108
Figura 74: Vista desde el sitio web, una tarjeta de presentación	109
Figura 75: Vista de la fotografía y el título del último evento	109
Figura 76: Vista de la información completa del evento realizado.....	110
Figura 77: Fotografía del expositor a cargo del evento realizado	110

ÍNDICE DE TABLAS

Tabla 1: Plantilla para las historias de usuario	19
Tabla 2: Plantilla para las tareas de ingeniería	20
Tabla 3: Plantilla para las pruebas de aceptación	21
Tabla 4: Plantillas para las tarjetas CRC	21
Tabla 5: Descripción de los requerimientos funcionales.....	48
Tabla 6: Descripción de los requerimientos no funcionales.....	49
Tabla 7: Asignación de roles del proyecto.....	49
Tabla 8: Historia de usuario para el acceso al sistema	50
Tabla 9: Historia de usuario creación de permisos	50
Tabla 10: Historia de usuario gestión de usuarios	50
Tabla 11: Historia de usuario gestión colección de presidentes.....	51
Tabla 12: Historia de usuario gestión de visitas.....	51
Tabla 13: Historia de usuario gestión de eventos	51
Tabla 14: Historia de usuario gestión de noticias	52
Tabla 15: Historia de usuario gestión de blogs	52
Tabla 16: Historia de usuario análisis comparativo	52
Tabla 17: Historia de usuario algoritmo de enmascaramiento	53
Tabla 18: Historia de usuario implementación de algoritmo	53
Tabla 19: Plan de entregas del proyecto	54
Tabla 20: Apoyo a las actividades institucionales.....	63
Tabla 21: Historias de usuario primera iteración.....	64
Tabla 22: Tareas de ingeniería primera iteración.....	64
Tabla 23: Tarea de ingeniería 1, historia de usuario 1	65
Tabla 24: Tarea de ingeniería 2, historia de usuario 1	65
Tabla 25: Tarea de ingeniería 3, historia de usuario 2.....	65
Tabla 26: Tarea de ingeniería 4, historia de usuario 2.....	66
Tabla 27: Tarea de ingeniería 5, historia de usuario 3.....	66
Tabla 28: Tarea de ingeniería 6, historia de usuario 3.....	66
Tabla 29: Pruebas de aceptación primera iteración	67
Tabla 30: Caso de prueba, acceso al sistema.....	67
Tabla 31: Caso de prueba, creación de permisos.....	68
Tabla 32: Caso de prueba, gestión de usuario	68
Tabla 33: Historias de usuario segunda iteración	72

Tabla 34: Tareas de ingeniería segunda iteración	73
Tabla 35: Tarea de ingeniería 7, historia de usuario 4	73
Tabla 36: Tarea de ingeniería 8, historia de usuario 4	73
Tabla 37: Tarea de ingeniería 9, historia de usuario 5	74
Tabla 38: Tarea de ingeniería 10, historia de usuario 5	74
Tabla 39: Tarea de ingeniería 11, historia de usuario 6	74
Tabla 40: Tarea de ingeniería 12, historia de usuario 6	75
Tabla 41: Pruebas de aceptación segunda iteración	75
Tabla 42: Caso de prueba gestión de colección de presidentes	76
Tabla 43: Caso de prueba gestión de visitas	76
Tabla 44: Caso de prueba gestión de eventos	77
Tabla 45: Historias de usuario tercera iteración	82
Tabla 46: Tareas de ingeniería segunda iteración	83
Tabla 47: Tarea de ingeniería 13, historia de usuario 7	83
Tabla 48: Tarea de ingeniería 14, historia de usuario 7	83
Tabla 49: Tarea de ingeniería 15, historia de usuario 8	84
Tabla 50: Tarea de ingeniería 16, historia de usuario 8	84
Tabla 51: Pruebas de aceptación segunda iteración	84
Tabla 52: Caso de prueba gestión de noticias	85
Tabla 53: Caso de prueba gestión de blogs	85
Tabla 54: Incorporación de elementos visuales	101
Tabla 55: Tabla de las entradas de usuario	111
Tabla 56: Tabla de las salidas de usuario	112
Tabla 57: Tabla de las peticiones de usuario	112
Tabla 58: Tabla de los archivos lógicos	113
Tabla 59: Calcular total punto de función del sistema	113
Tabla 60: Valores de complejidad	113
Tabla 61: Ajustes de complejidad	114
Tabla 62: Escala de ajuste de usabilidad	115
Tabla 63: Cuestionario de usabilidad	115
Tabla 64: Mantenibilidad del sistema	117
Tabla 65: Portabilidad del sistema	118
Tabla 66: Resultados calidad total	118
Tabla 67: Tabla de costos de estudio	119
Tabla 68: Factor LDC/PF de lenguajes de programación	120

ÍNDICE DE ECUACIONES

Ecuación 1: Ecuación punto de función	114
Ecuación 2: Ecuación de usabilidad del software.....	116
Ecuación 3: Ecuación de índice de madurez del software	116
Ecuación 4: Ecuación de líneas de código	120
Ecuación 5: Ecuación de líneas de código distribuidas en el proyecto	122
Ecuación 6: Ecuación de esfuerzo	122
Ecuación 7: Ecuación tiempo de desarrollo de software.....	122
Ecuación 8: Ecuación costo de software.....	123
Ecuación 9: Ecuación costo total del software	123

ÍNDICE DE ANEXOS

ANEXOS A: Árbol de problemas.....	133
ANEXOS B: Ubicación academia boliviana de historia militar	134
ANEXOS C: Tabla de objetivos específicos y sus acciones	135
ANEXOS D: Tabla de contenido temático	136
ANEXOS E: Cuadro de entrevistas.....	138



ABSTRACT



ABSTRACT

Nowadays, Information and Communication Technologies (ICT) are a fundamental tool for supporting academic and cultural institutions in achieving their objectives. The use of information systems allows for the optimization of processes, ensures data security, and facilitates the efficient management of digital content. In this context, the present thesis project titled “CONTENT MANAGEMENT SYSTEM WITH MASKING TECHNIQUE FOR THE BOLIVIAN ACADEMY OF MILITARY HISTORY” was developed with the aim of strengthening the institutional mission of preserving and disseminating Bolivia’s military history through a secure, functional, and adaptable technological solution.

The implemented system includes modules for the management of users, content, publications, events, and other resources of historical interest. One of its distinctive features is the incorporation of a data masking technique, the result of a comparative analysis of different alternatives, which ensures the confidentiality of information stored in the database without compromising the system’s usability. User-centered design elements were also considered, with friendly and easy-to-navigate interfaces.

The system development was based on the principles of the Extreme Programming (XP) methodology to enable incremental deliveries and continuous validations. The defined objectives were achieved, from the requirements gathering to the technical validation through functional and unit tests. As a result, a robust system with a scalable architecture was obtained, aligned with the specific requirements of the Bolivian Academy of Military History.

Keywords: Content management system, data masking, military history, information security, extreme programming.

RESUMEN

En la actualidad, las Tecnologías de la Información y la Comunicación (TIC) representan una herramienta fundamental para apoyar a instituciones académicas y culturales en el cumplimiento de sus objetivos, el uso de sistemas informáticos permite optimizar procesos, garantizar la seguridad de la información y facilitar la gestión eficiente de contenidos digitales. En ese marco, el presente proyecto de grado titulado “SISTEMA DE GESTIÓN DE CONTENIDOS CON TÉCNICA DE ENMASCARAMIENTO PARA LA ACADEMIA BOLIVIANA DE HISTORIA MILITAR”, fue desarrollado con el fin de fortalecer la misión institucional de preservar y difundir la historia militar de Bolivia mediante una solución tecnológica segura, funcional y adaptable.

El sistema implementado contempla módulos para la gestión de usuarios, contenidos, publicaciones, eventos y otros recursos de interés histórico, una de sus características distintivas es la incorporación de una técnica de enmascaramiento de datos, producto de un análisis comparativo entre diferentes alternativas, que permite asegurar la confidencialidad de la información almacenada en la base de datos sin afectar la usabilidad del sistema, se consideraron elementos de diseño centrados en el usuario, con interfaces amigables y de fácil navegación.

El desarrollo del sistema se basó en los principios de la metodología de programación extrema (XP) para lograr entregas incrementales y validaciones continuas, cumpliendo los objetivos planteados, tanto en el levantamiento de requerimientos como en la validación técnica mediante pruebas funcionales y unitarias, dando como resultado, se obtuvo un sistema robusto, con arquitectura escalable y alineado a los requerimientos específicos de la Academia Boliviana de Historia Militar.

Palabras clave: Sistema de gestión de contenidos, enmascaramiento de datos, historia militar, seguridad informática, programación extrema.



CAPÍTULO 1.

GENERALIDADES

1.1. INTRODUCCIÓN

En el corazón de Bolivia, entre las páginas gastadas de archivos antiguos y documentos relegados por el tiempo, reposa el legado de una nación forjada por su tradición, la Academia Boliviana de Historia Militar, como custodio de este invaluable patrimonio, enfrenta el reto de proyectar este tesoro hacia las nuevas generaciones en un entorno cada vez más digital, pero la carencia de una plataforma moderna para la administración de contenidos ha limitado su capacidad de cumplir a cabalidad su propósito de salvaguardar y difundir la memoria histórica vinculada a las gestas militares del país.

La era digital ha transformado la forma en que compartimos y accedemos al conocimiento histórico, la falta de presencia en línea de la Academia limita su alcance y su capacidad para llegar a un público más amplio, desde investigadores y estudiantes hasta apasionados de la historia militar, es por esta razón que, en respuesta a este desafío, reconociendo la importancia de digitalizar y poner en línea este tesoro patrimonial y al mismo tiempo que se asegura su preservación a largo plazo.

El propósito central de este proyecto de grado es brindar a la Academia Boliviana de Historia Militar las herramientas necesarias para cumplir su misión en la era digital. a través del desarrollo de un sistema de gestión de contenidos en línea, se busca abrir las puertas de este vasto tesoro histórico a un público global, proporcionando un acceso seguro y eficiente a sus archivos y recursos.

A lo largo de este documento, se explorará la metodología que guiará la implementación de este sistema, se examinarán las etapas clave del proyecto, desde la evaluación de necesidades hasta la validación del sistema, además, se discutirán los beneficios que se esperan lograr, tanto para la Academia como para aquellos que buscan descubrir y aprender de la rica historia militar de Bolivia.

Este proyecto no solo representa un testimonio de nuestro firme compromiso con la preservación y promoción de la herencia histórica de nuestra nación, sino que también marca un hito en el camino para asegurar que la voz de nuestro pasado resuene vibrante y relevante en la era digital y para las generaciones venideras. A través de iniciativas innovadoras y colaborativas, buscamos no solo conservar la historia, sino también revitalizarla, haciéndola accesible y significativa para todos.

1.2. ANTECEDENTES

Los antecedentes del presente proyecto se definirán de la siguiente manera:

1.2.1. Antecedentes académicos

IMPLEMENTACIÓN DE UNA TIENDA VIRTUAL UTILIZANDO EL CMS JOOMLA Y VIRTUEMART PARA LA OFERTA DE CURSOS Y SEMINARIOS DE CAPACITACIÓN EN LA FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN DE LA ESPOL, este proyecto fue desarrollado por los Licenciados John Humberto Moreno Merchán, Lidia Judith Lucero Gonzáles, en el año 2015 en la Escuela Superior Politécnica del Litoral, en la Facultad de Ingeniería en Electricidad y Computación, cuyo objetivo fue: “Implementar una Tienda Virtual Interactiva para Ofertar Cursos y Seminarios de Capacitación que contribuya con el perfeccionamiento y desarrollo formativo de los participantes, así recopilar toda la información necesaria de las personas interesadas, permitiéndoles el acceso a la información de forma oportuna y rápida.” (Moreno Merchán & Lucero Gonzáles, 2015)

El proyecto mencionado garantizará un acceso oportuno y rápido a la información relevante para mejorar la experiencia de aprendizaje y facilitar el proceso de inscripción y participación en los cursos y seminarios ofrecidos.

OPTIMIZACIÓN DEL DISEÑO DEL SISTEMA DE GESTIÓN DE CONTENIDOS (CMS) PARA MEJORAR EL POSICIONAMIENTO WEB DE LA UNIDAD DE POSGRADO DE LA FACULTAD DE INGENIERÍA MECÁNICA DE LA UNIVERSIDAD NACIONAL DE INGENIERÍA, este proyecto fue desarrollado por la Licenciada Diana Carolina Segovia Martínez, en el año 2017 en la Universidad Tecnológica de Perú, en la Facultad de Ingeniería Industrial y Mecánica Carrera de Ingeniería de Diseño Gráfico, cuyo objetivo fue: “Optimizar el diseño del Sistema de Gestión de Contenidos (CMS) para mejorar el posicionamiento web la Unidad de Posgrado de la Facultad de Ingeniería Mecánica de la Universidad Nacional de Ingeniería.” (Segovia Martínez, 2017)

El proyecto mencionado anteriormente se centra en potenciar el posicionamiento en línea y la promoción de programas académicos, proporcionándonos herramientas sólidas para implementar y optimizar un sistema integral de gestión de contenido, abarcando tanto el desarrollo frontend como backend.

ENMASCARAMIENTO EN LA BASE DE DATOS ORACLE PARA RESGUARDAR LA INFORMACIÓN FINANCIERA Y PERSONAL EN LA COOPERATIVA DE AHORRO Y CRÉDITO DE LA PEQUEÑA EMPRESA DE COTOPAXI, este proyecto fue desarrollado por el licenciado Diego Marcelo Coronel Montoya, en el año 2018 en la Universidad Técnica de Ambato, en la Facultad de Ingeniería en Sistemas Electrónica e Industrial, cuyo objetivo fue: “Implementar un sistema de Enmascaramiento en la base de datos Oracle para resguardar la información financiera y personal en la Cooperativa de Ahorro y Crédito de la Pequeña Empresa de Cotopaxi.” (Coronel Montoya, 2018)

El proyecto mencionado garantiza la privacidad y seguridad de los datos sensibles, cumpliendo con estándares de protección y brindando confianza a los usuarios en la gestión de sus información financiera y personal.

HERRAMIENTA PARA EL ENMASCARADO DE DATOS EN BASES DE DATOS POSTGRESQL, MYSQL Y SQL SERVER, este proyecto fue desarrollado por el Licenciados Josué Mojena Marín, Osvaldo Víctor Melo Pérez, en el año 2014 en la Universidad de Ciencias Informáticas, en la Facultad 6 cuyo objetivo fue: “Desarrollar una herramienta que garantice con el enmascarado la integridad y confidencialidad de los datos en bases de datos PostgreSQL, MySQL y SQL Server.” (Mojena Marín & Melo Pérez, 2014)

El proyecto mencionado contribuirá significativamente a fortalecer la seguridad de los sistemas de gestión de bases de datos, brindando tranquilidad tanto a los administradores como a los usuarios finales.

TÉCNICAS DE TRANSPARENCIA Y ENCRIPCIÓN DE INFORMACIÓN, este proyecto fue desarrollado por el Licenciado Andrés Julián Sáenz Landazábal, en el año 2014 en la Universidad Libre, en la Facultad de Ingeniería cuyo objetivo fue: “Construir el documento de referencia que describe el proceso de encriptación como mecanismo de protección de una base de datos.” (Sáenz Landazábal, 2015)

El proyecto mencionado será una guía completa que proporcionará a los usuarios un entendimiento claro y práctico de cómo implementar y gestionar eficazmente el enmascaramiento para proteger la integridad y confidencialidad de los datos almacenados, al proporcionar este recurso se busca fortalecer la seguridad de las bases de datos, mitigar riesgos de acceso no autorizado y contribuir a la protección de la información sensible.

1.2.2. Antecedentes institucionales

La Academia, antes era conocida como el Palacio de Goytia, se encuentra ubicado en la Calle Bolívar 578, de la ciudad de La Paz, su construcción data de principios del siglo XX, es una edificación de estilo neoclásico perteneciente a la familia Goytia, familia de militares y de mucho reconocimiento en la sociedad de la época, dicha construcción, que con el transcurrir del tiempo se convirtió en el “Círculo Militar” y que en la actualidad es el repositorio histórico militar más importante del país, pues desde hace más de treinta años pasó a denominarse como “Academia Boliviana de la Historia Militar”.

Acerca de su propietario e impulsor Don Benedicto Goytia, fue un político nacido en la localidad de Camargo del Departamento de Chuquisaca un 12 de enero de 1851, quien había comenzado sus estudios en el Colegio Mercantil de Tupiza, posteriormente pasó a la ciudad de Sebastián Pagador en Oruro donde comenzó a trabajar y luego se trasladó a Nuestra Señora de La Paz en el año 1867.

Don Benedicto Goytia, fue el constructor y edificador de esta casona que hoy se muestra como un Tesoro Republicano en la plaza principal del centro político de Bolivia, lugar en el que se discurren los temas más importantes de la Historia Patria por destacados miembros de la Academia Boliviana de la Historia Militar.

Ya desde mucho antes que se levantará la actual edificación los predios donde se encuentra eran de especial trascendencia, precisamente por su estratégica ubicación, reuniendo así un fuerte historial que debe enorgullecer a todo aquel que visite sus ambientes. Es precisamente con el advenimiento de la República que la ciudad de La Paz va en crecimiento y el antiguo cuartel patriota se convirtió en un local de tiendas de comercio y posteriormente pasaría a ser el Hotel Guibert, seguramente de origen francés, que contaba con un portal labrado en piedra a su ingreso y sin duda durante esa época recibió a una gran cantidad de notables pasajeros.

El Palacio Goytia, hoy en día Academia Boliviana de la Historia Militar se constituye en una pieza arquitectónica de gran valor para la ciudad de La Paz, este repositorio histórico, en su terreno inicial ocupa parte de lo que fue el cuartel de la Guardia real española y con la revolución del 16 de julio de 1809 pasó a ser cuartel patriota y en plena época republicana se convierte en el hotel central Guibert.

Como mencionamos, originalmente la edificación fue utilizada como vivienda, propiedad de Don Benedicto Goytia, posteriormente Círculo Militar, para después acondicionar sus espacios y convertirse en la Academia Boliviana de la Historia Militar, este inmueble se constituye en un importante ejemplo de arquitectura civil, privada, puesto que presenta las características estilísticas constructivas y materiales usados en su correspondiente momento histórico.

1.2.2.1. Misión de la institución

La Academia Boliviana de Historia Militar deberá coordinar y fomentar la investigación historiográfica en las FF.AA., con rigor documental y bases científicas, durante todo el proceso de formación, capacitación y especialización militar, para preservar y promover la cultura histórica, cívica y patriótica en el personal militar, a fin de formar ciudadanos con identificación y conciencia nacional y alto compromiso con la Patria, aptos para defenderla y apoyar activa y decididamente a su desarrollo. (Academia Boliviana Historia Militar, 2024)

1.2.2.2. Visión de la institución

La Academia Boliviana de Historia Militar, deberá constituirse en el plazo inmediato y en el futuro Centro Cultural de las FF.AA., único y con responsabilidad en el establecimiento y manejo de los objetivos, políticas, estrategias y acciones en los campos culturales e históricos. Siendo además asesor y coordinador con la Universidad de las FF.AA. en materia de historia, geografía, sociología, filosofía y etnografía en los diferentes niveles de formación, capacitación y especialización, asimismo el ente rector en la implementación manejo y conservación de los Museos de FF.AA. (Academia Boliviana Historia Militar, 2024)

1.3. PLANTEAMIENTO DEL PROBLEMA

Para estructurar y definir de manera precisa los problemas que serán abordados en el proyecto, se utilizará el método del árbol de problemas; a través de este enfoque, se analizarán y desglosarán detalladamente los desafíos identificados, permitiendo una comprensión exhaustiva de las áreas que requieren atención y de esta forma, se establecerán bases sólidas para la formulación de soluciones efectivas que contribuyan al éxito del presente trabajo. Ver el anexo A.

1.3.1. Identificación del problema

La Academia Boliviana de Historia Militar, es una institución que se encarga de preservar y promover la historia militar, ostenta un valioso patrimonio de documentos, archivos y recursos históricos que constituyen un legado invaluable para la nación, englobando una diversidad de registros que abarcan desde manuscritos antiguos hasta fotografías, mapas, uniformes y objetos militares históricos, cada uno de estos elementos atesora una gran parte fundamental de Bolivia.

Uno de los desafíos clave en el desarrollo de este sistema es garantizar la seguridad de los contenidos históricos militares, la información sensible y estratégica relacionada con la historia militar de Bolivia necesita ser protegida contra posibles amenazas cibernéticas, filtraciones de datos o acceso no autorizado, para evitar cambios que puedan afectar directamente a la institución. Es esencial garantizar que el sistema pueda manejar grandes volúmenes de datos y proporcionar acceso rápido a los usuarios, la gestión de una gran cantidad de documentos, imágenes, videos y otros recursos, lo que plantea desafíos en términos de capacidad y rendimiento.

La implementación de técnicas de enmascaramiento seguras y eficientes es un aspecto técnico crítico del proyecto deben ser seleccionadas y configuradas de manera adecuada para proteger la información sensible sin comprometer su accesibilidad para los usuarios autorizados.

1.3.2. Formulación del problema

Actualmente la Academia Boliviana de Historia Militar no cuenta con un espacio para la difusión del contenido histórico que sea accesible para estudiantes e investigadores lo que provoca limitaciones en el cumplimiento de su misión.

1.4. OBJETIVOS

Para establecer una dirección clara y alcanzar resultados significativos, los objetivos del presente proyecto se delinearán de manera precisa y detallada, asegurándonos que cada etapa contribuya de manera efectiva hacia su cumplimiento.

1.4.1. Objetivo general

Desarrollar un sistema de gestión de contenidos empleando una técnica de enmascaramiento de manera segura y eficiente para la Academia Boliviana de Historia Militar, que coadyuvará a la misión de preservar y promover datos históricos.

1.4.2. Objetivos específicos

- Realizar un análisis de las condiciones en las que se desarrollarán las actividades de la Academia Boliviana de Historia Militar para determinar los requerimientos funcionales y no funcionales del sistema.
- Diseñar la arquitectura y estructura del sistema de gestión de contenidos, que sirva de apoyo para cumplir las actividades de la Academia Boliviana de Historia Militar.
- Implementar una técnica de enmascaramiento en base a un análisis comparativo de las existentes, para lograr un almacenamiento seguro en la base de datos.
- Desarrollar e integrar las técnicas de enmascaramiento con el sistema de gestión de contenidos incorporando elementos visuales claros y colores amigables para facilitar la navegación y comprensión de los usuarios.
- Validar el sistema de gestión de contenidos realizando pruebas unitarias y funcionales para determinar el correcto funcionamiento.

1.5. JUSTIFICACIONES

Las justificaciones del presente proyecto se definirán de la siguiente manera:

1.5.1. Justificación social

Al desarrollar un sistema de gestión de contenidos que sea seguro y eficiente, compensa el acceso a esta valiosa información, llegando a un público más amplio, incluyendo investigadores, estudiantes y entusiastas de la historia militar, esto fomenta el conocimiento y el aprecio por la historia del país, fortaleciendo la identidad nacional y la comprensión de eventos históricos cruciales, además, al garantizar la seguridad de los datos, se protege la confidencialidad y la integridad de la información, generando confianza entre los usuarios y promoviendo la transparencia en la gestión de contenidos.

1.5.2. Justificación económica

Se reducirán los costos asociados con la preservación física de documentos históricos y la gestión de archivos en papel, además, al emplear la técnica de enmascaramiento de datos, se garantiza la seguridad de la información sin necesidad de inversiones costosas en infraestructura de seguridad, la eficiencia operativa mejorada también reduce los gastos relacionados con la administración de archivos y permite un acceso más rápido y eficiente a los contenidos, lo que ahorra tiempo y recursos para la Academia Boliviana de Historia Militar.

1.5.3. Justificación técnica

La implementación de un sistema de gestión de contenidos en línea aprovecha las ventajas de la tecnología digital, permitiendo un acceso ágil y global a los recursos históricos, además, de usar el empleo de técnicas de enmascaramiento de datos garantiza la seguridad de la información, cumpliendo con estándares de protección de datos, esta solución técnica no solo moderniza la gestión de contenidos históricos, sino que también asegura su integridad y accesibilidad a largo plazo, alineándose con los avances tecnológicos y las mejores prácticas en preservación digital. La implementación se hará desde cero para adaptarlo completamente a las necesidades específicas del proyecto, esto nos brindará un control total sobre el rendimiento, seguridad y personalización, evitando la complejidad innecesaria de plugins y temas preexistentes, lo que garantiza una solución más eficiente y escalable.

1.6. ALCANCES

Los alcances del presente proyecto se dividen de la siguiente manera:

1.6.1. Alcance temático

- a) **Área general:** Ingeniería de sistemas.
- b) **Áreas específicas:** Auditoría de Sistemas, Bases de datos, Ingeniería de Software, Proyectos Web y Cloud Computing.
- c) **Área temática:** Criptografía y técnicas de enmascaramiento de datos.

1.6.2. Alcance geográfico

El presente trabajo será desarrollado para la Academia Boliviana de Historia Militar, que se encuentra ubicada en la Calle Bolívar 578, de la ciudad de La Paz. Ver el anexo B.

1.6.3. Alcance temporal

El presente proyecto de grado se realizará en los dos semestres académicos según las políticas de la Universidad La Salle Bolivia.

1.7. LIMITES

Los límites del sistema de gestión de contenidos para la Academia Boliviana de Historia Militar serán los siguientes casos:

- El proyecto no abarca otros aspectos institucionales de la Academia Boliviana de Historia Militar que no estén directamente relacionados con la gestión de contenidos en línea, como la administración interna o cuestiones financieras.
- No se aborda el manejo de recursos físicos, ni el mantenimiento físico de documentos históricos y archivos fuera del ámbito digital.
- El sistema se enfoca exclusivamente en la historia militar de Bolivia y no incluye contenidos relacionados con otras áreas históricas o temas.



CAPÍTULO 2.

MARCO TEÓRICO

2.1. INTRODUCCIÓN

El proyecto tiene un marco teórico que cubre diversos temas fundamentales que serán la base conceptual y metodológica para analizar y solucionar problemas específicos que se han seleccionado cuidadosamente debido a su importancia y pertinencia para el alcance y los objetivos.

En el ámbito académico y laboral, es crucial consolidar el conocimiento adquirido y aplicar las teorías en la práctica, centrándonos en la exploración y el fortalecimiento de un marco teórico sólido que apoye y mejore la investigación realizada.

En este marco teórico se estudiarán conceptos para brindar una comprensión completa y contextualizada del tema de investigación, donde se analizarán métodos y teorías pertinentes que orientarán la creación y la ejecución propuesta.

Durante este proyecto, se enfatizará la integración de teorías y prácticas, promoviendo un enfoque interdisciplinario que facilite el abordaje de desafíos desde múltiples perspectivas y enriquezca el análisis con distintos puntos de vista, esperando que esta combinación de conocimientos teóricos y metodológicos que contribuyan de manera significativa a la generación de resoluciones de los problemas planteados.

En resumen, el actual proyecto se basa en un sólido marco teórico que cubre varios temas relevantes que proporcionan una base conceptual y metodológica fuerte para la investigación propuesta, que busca contribuir al avance del conocimiento, así como a la identificación de soluciones innovadoras y efectivas para los desafíos planteados mediante la exploración y análisis crítico de estas teorías y enfoques.

2.2. TÉCNICAS DE RECOLECCIÓN DE DATOS

Cuando se realiza un trabajo de investigación, es necesario considerar los métodos, las técnicas e instrumentos como aquellos elementos que aseguran el hecho empírico de la investigación, donde el método representa el camino a seguir la investigación, las técnicas constituyen el conjunto de instrumentos en el cual se efectúa el método, mientras que el

instrumento incorpora el recurso o medio que ayuda a realizar la investigación, además el uso de técnicas de recolección de información es una etapa donde se inspecciona y se transforman los datos con el objetivo de resaltar información útil, lo que sugiere conclusiones y apoyo a la toma de decisiones. (Hernández Mendoza & Duana Avila, 2020)

2.2.1. La técnica de entrevista

Según el libro de Técnicas para investigar Recursos Metodológicos para la preparación de Proyectos de Investigación de (Yuni & Urbano, 2024) las investigaciones basadas en entrevistas son relevantes para la investigación social y cultural porque permiten obtener información provista por los propios sujetos, y con ello se obtiene un acceso más directo a los significados que éstos le otorgan a su realidad. Por lo tanto, a través de las entrevistas se obtiene información sobre ideas, creencias y concepciones de las personas entrevistadas, estas producciones de los sujetos pueden referirse a hechos o sucesos ocurridos en el pasado, la posibilidad que da la entrevista como herramienta para acceder a información de fenómenos de estudio ubicados en distintos contextos temporales es muy amplia, ya que permite indagar sobre el pasado, sobre el presente y también sobre las anticipaciones acerca del futuro.

2.2.1.1. Tipos de entrevistas

Existen diferentes tipos de entrevistas, según el grado de estructuración y el nivel de flexibilidad en la interacción con el entrevistado:

- **Según la situación de la interacción:** El entrevistado y el entrevistador conversan en un lugar previamente fijado, aquí el entrevistador tiene la oportunidad de obtener información verbal en la entrevista ya que se van planteando los objetivos, permitiéndose que se formulen preguntas claves para el desarrollo de la entrevista que ayuden a llegar al objetivo final que se requiere. (Mata Solís, 2020)
- **Según el número de participantes:** Un grupo de personas es entrevistado simultáneamente por un entrevistador que generalmente está apoyado por uno o varios observadores, en las entrevistas grupales se obtiene en poco tiempo una gran cantidad de información, dado que la interacción entre los sujetos entrevistados permite que emerjan las diferentes perspectivas sobre un tema, las coincidencias y divergencias respecto a hechos o a interpretaciones de los mismos. (Amezcuza, 2003)

2.2.2. Investigación documental

Según (Yuni & Urbano, 2024) la investigación documental, se basa en la observación y el análisis de la “documentación” lo que nos permite volver la mirada hacia un tiempo pasado para de este modo comprender e interpretar una realidad actual de los acontecimientos pasados que han sido los antecedentes que han derivado en los consecuentes de situaciones, acontecimientos y procesos de una realidad determinada, esto posibilita una mirada retrospectiva (hacia atrás), una mirada actual, y otra prospectiva (hacia delante) de la realidad que es objeto de indagación. De este modo, el análisis documental le permite al investigador ampliar el campo de observación y enmarcar la realidad objeto de investigación dentro del acontecer histórico; lo cual amplía la captación de los significados que nos permiten mirar esa realidad desde una perspectiva más global y holística.

2.2.2.1. Clasificación de Documentos

- **Documentos escritos:** Son aquellos que emplean un sistema de símbolos que suponen narraciones a través de sistemas de signos convencionales (términos) que expresan situaciones, acontecimientos y procesos. (Hernández Sampieri, Fernández Collado, & Baptista Lucio, 2014)
- **Documentos visuales:** Son aquellos en los que se expresan ideas a través de elementos aprehensibles mediante la observación directa; la cual supone la utilización de sentidos como la vista, el olfato y el tacto. (Uwe Flick, 2007)
- **Documentos acerca de hechos reales:** Son aquellos documentos que registran acontecimientos, situaciones y procesos, presentes o pasados, que se producen o se han producido espontáneamente. (Yin, 2007)
- **Documentos públicos:** Son aquellos materiales que se han producido con la intencionalidad de ser difundidos, dentro de este tipo de documentos es posible encontrar: informes y estadísticas oficiales en general, obras literarias, anuarios, discursos, memorias, biografías, etc. (Yuni & Urbano, 2024)
- **Documentos que hacen referencia a restos físicos:** Son aquellos documentos que están constituidos en las pinturas, esculturas y arquitectura que nos permite la aproximación a la historia de un pueblo a partir de restos físicos de su cultura. (Hodder, 1994)

2.3. INGENIERÍA DE SOFTWARE

La ingeniería de software según (Sommerville, 2005) es una disciplina de la ingeniería que comprende todos los aspectos de la producción de software desde las etapas iniciales de la especificación del sistema, hasta el mantenimiento de éste después de que se utiliza. En esta definición, existen dos frases clave:

- **Disciplina de la ingeniería.** - Los ingenieros hacen que las cosas funcionen, aplican teorías, métodos y herramientas donde sean convenientes, pero las utilizan de forma selectiva y siempre tratando de descubrir soluciones a los problemas, aun cuando no existan teorías y métodos aplicativos. (Sommerville, Ingeniería del software, 2005)
- **Todos los aspectos de producción de software.** – La ingeniería del software no sólo comprende los procesos técnicos del desarrollo de software, sino que también con actividades tales como la gestión de proyectos de software y el desarrollo de herramientas. (Sommerville, Ingeniería del software, 2005)

2.3.1. Actividades y procesos

Para (Sommerville, 2005) las cuatro actividades básicas del proceso son la especificación, el desarrollo, la validación y la evolución que se organizan de forma distinta en diferentes procesos del desarrollo. En el enfoque en cascada, están organizadas en secuencia, mientras que en el desarrollo evolutivo se entrelazan, cómo se llevan a cabo estas actividades depende del tipo de software, de las personas y de la estructura organizacional implicadas, no hay una forma correcta o incorrecta de organizar estas actividades.

2.3.2. Especificación del software

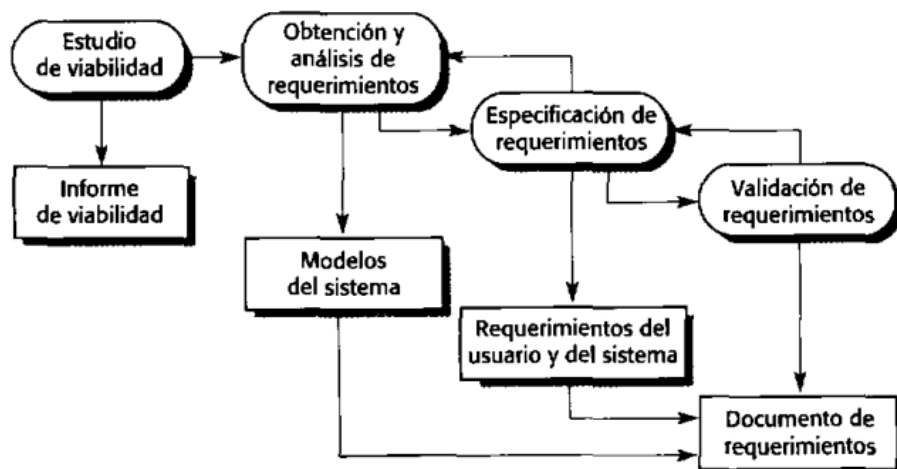
La especificación del software es el proceso de comprensión y definición de qué servicios se requieren para el sistema y de la identificación de las restricciones de funcionamiento y desarrollo del mismo. Es una etapa particularmente crítica en el proceso del software ya que los errores en esta etapa originan inevitablemente problemas posteriores en el diseño e implementación del sistema, los usuarios finales y los clientes necesitan una declaración de alto nivel de los requerimientos, mientras que los desarrolladores del sistema necesitan una especificación más detallada de éste. Para (Sommerville, 2005) existen cuatro fases principales en el proceso de ingeniería de requerimientos:

- **Estudio de viabilidad.** – Se analiza si el sistema propuesto será rentable desde un punto de vista de negocios y si se puede desarrollar dentro de las restricciones de presupuesto existentes. (Bridges, 2025)
- **Obtención y análisis de requerimientos.** - Obtener los requerimientos del sistema por medio de la observación de los sistemas existentes, discusiones con los usuarios potenciales y proveedores, el análisis de tareas, etc. (Espacios Media, 2016)
- **Especificación de requerimientos.** - Es la actividad de traducir la información recopilada durante la actividad de análisis en un documento que define un conjunto de requerimientos. (Sommerville, Ingeniería del software, 2005)
- **Validación de requerimientos.** - En este proceso, inevitablemente se descubren errores en el documento de requerimientos y se debe modificar entonces para corregir estos problemas. (Visure, 2023)

En la Figura 1 se muestra el proceso de ingeniería de requerimientos.

Figura 1:

El proceso de la ingeniería de requerimientos.



Nota: En la figura se muestra el proceso de requerimientos que utiliza en la parte de especificación del software, que nos sirve para la comprensión y definición de servicios que requiere el sistema. Fuente: (Sommerville, 2005)

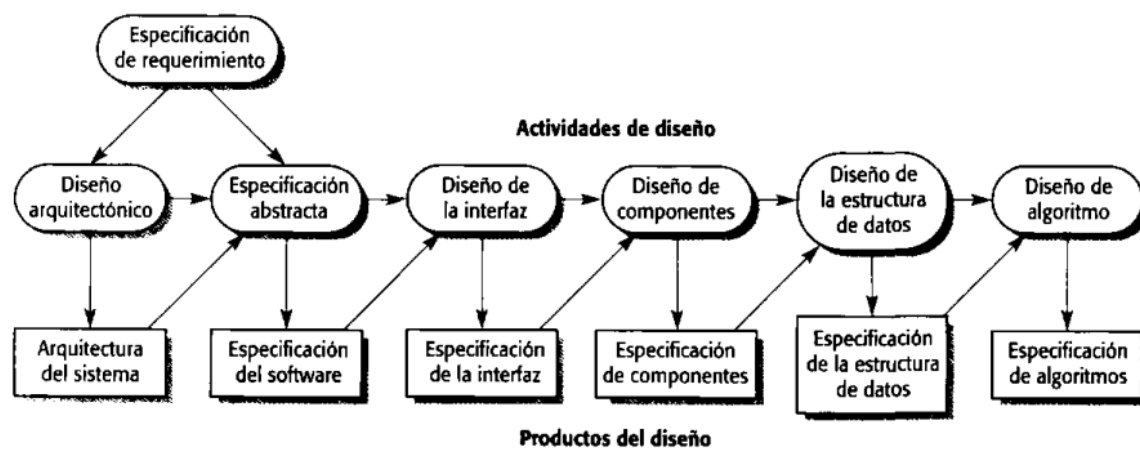
2.3.3. Diseño e implementación del software

Según (Sommerville, Ingeniería del software, 2005) un diseño de software es una descripción de la estructura del software que se va a implementar, los datos que son parte del sistema, las interfaces entre los componentes del sistema y, algunas veces, los

algoritmos utilizados, el proceso de diseño puede implicar el desarrollo de varios modelos del sistema con diferentes niveles de abstracción, mientras se descompone un diseño, se descubren errores de las etapas previas, esta retroalimentación permite mejorar los modelos de diseño previos, ver la Figura 2 que es un modelo de este proceso que muestra las descripciones de diseño que pueden producirse en varias etapas del diseño.

Figura 2:

El modelo general del proceso de diseño



Nota: En la figura se muestra el modelo de procesos del diseño del software, que nos sirve para la descubrir los errores y mediante esto nos permite hacer una retroalimentación para mejorar los modelos previos. Fuente: (Sommerville, 2005)

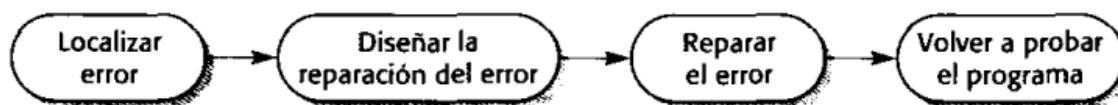
Las actividades específicas del proceso de diseño son:

- **Diseño arquitectónico.** - Los subsistemas que forman el sistema y sus relaciones se identifican y documentan. (Egaña, 2022)
- **Especificación abstracta.** - Para cada subsistema se produce una especificación abstracta de sus servicios y las restricciones. (Sommerville, 2005)
- **Diseño de la interfaz.** - Para cada subsistema se diseña y documenta su interfaz con otros subsistemas. (Sommerville, 2005)
- **Diseño de componentes.** - Se asignan servicios a los componentes y se diseñan sus interfaces. (Pressman, 2021)
- **Diseño de la estructura de datos.** - Se diseña en detalle y especifica la estructura de datos utilizada en la implementación del sistema. (Sommerville, 2005)
- **Diseño de algoritmos.** - Se diseñan en detalle y especifican los algoritmos utilizados para proporcionar los servicios. (Pressman, 2021)

Según (Sommerville, 2005) la programación es una actividad personal y no existe un proceso general que se siga comúnmente, algunos programadores empiezan con los componentes que comprenden, los desarrollan y después continúan con los que comprenden menos, otros toman el enfoque opuesto dejando los componentes que son más familiares hasta el final debido a que saben cómo desarrollarlos. Normalmente los programadores llevan a cabo algunas pruebas del código que han desarrollado, a menudo esto muestra defectos en el programa que se deben eliminar del mismo, y esto se denomina depuración, las pruebas y la depuración de defectos son procesos diferentes, las pruebas establecen la existencia de defectos y la depuración comprende la localización y corrección de estos defectos. Para comprender mejor ver la Figura 3.

Figura 3:

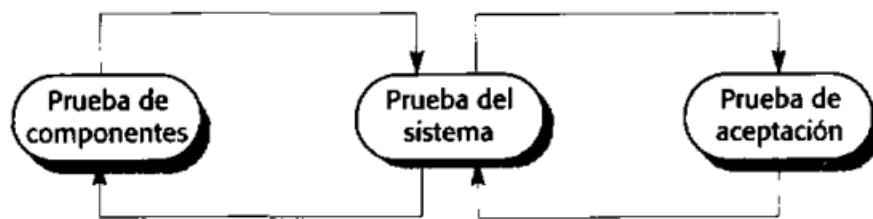
El proceso de depuración



Nota: En la figura se muestra el proceso de depuración que utilizan los programadores, esto les ayuda a localizar los errores que deben solucionarse para no tener fallas en el sistema. Fuente: (Sommerville, 2005)

2.3.4. Validación del software

La validación del software o de forma general, la verificación y validación se utiliza para mostrar que el sistema se ajusta a su especificación y que cumple las expectativas del usuario que lo comprará, esto implica procesos de comprobación. como las inspecciones y revisiones, en cada etapa del proceso del software desde la definición de requerimientos hasta el desarrollo del programa, la mayoría de los costos de validación aparecen después de la implementación, cuando se prueba el funcionamiento del sistema, en el mejor de los casos, los defectos se descubren en las etapas iniciales del proceso y los problemas con la interfaz, cuando el sistema se integra, sin embargo, cuando se descubren defectos el programa debe depurarse y esto puede requerir la repetición de otras etapas del proceso de pruebas y los errores en los componentes del programa pueden descubrirse durante las pruebas del sistema, por lo tanto, el proceso es iterativo y se retroalimenta tanto de las últimas etapas como de la primera parte del proceso. La Figura 4 según (Sommerville, 2005) se ve un proceso de pruebas de tres etapas en el cual se prueban los componentes del sistema, la integración del sistema y, el sistema con los datos del cliente.

Figura 4:*El proceso de pruebas*

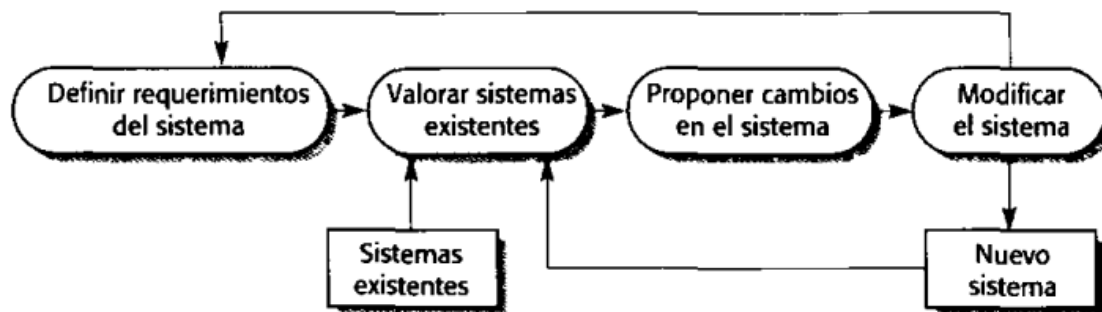
Nota: En la figura se muestra un proceso de pruebas de tres etapas en el cual se prueban los componentes del sistema, la integración del sistema y finalmente el sistema con los datos del cliente. Fuente: (Sommerville, 2005)

Las etapas del proceso de pruebas son:

- **Prueba de componentes.** - Se prueban los componentes individuales para asegurarse de que funcionan correctamente. (Pressman, 2021)
- **Prueba del sistema.** - Este proceso comprende encontrar errores que son el resultado de interacciones no previstas entre los componentes. (Sommerville, 2005)
- **Prueba de aceptación.** - Se prueban con los datos proporcionados por el cliente más que con datos de prueba simulados. (Pressman, 2021)

2.3.5. Evolución del software

(Sommerville, 2005) muestra la flexibilidad de los sistemas software es una de las principales razones por la que más y más software se incorpora a los sistemas grandes y complejos, una vez que se decide adquirir hardware, es muy costoso hacer cambios en su diseño, sin embargo, se pueden hacer cambios al software en cualquier momento durante o después del desarrollo del sistema, llegando a ser cambios importantes que son todavía mucho más económicos que los correspondientes de los sistemas hardware, la gente considera el desarrollo de software como una actividad creativa en la cual un sistema software se desarrolla desde un concepto inicial hasta que se pone en funcionamiento, a veces consideran el mantenimiento del software como algo aburrido y sin interés, pocos sistemas software son completamente nuevos, lo que implica que tiene más sentido ver el desarrollo y el mantenimiento como actividades continuas, más que dos procesos separados, es más realista considerar a la ingeniería del software como un proceso evolutivo ver la Figura 5 en el cual el software se cambia continuamente durante su periodo de vida como respuesta a los requerimientos cambiantes y necesidades del usuario.

Figura 5:*Evolución del sistema*

Nota: En la figura se muestra cómo se cambia el software continuamente durante su periodo de vida como respuesta a los requerimientos cambiantes y necesidades del usuario. Fuente: (Sommerville, 2005)

2.4. METODOLOGÍA EXTREME PROGRAMMING (XP)

Según (Letelier & Penadés, 2006) la metodología ágil XP está centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo, esto se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios, XP también se adecua para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

2.4.1. Herramientas de la metodología XP

2.4.1.1. Historias de usuario

Las historias de usuario representan una breve descripción del comportamiento del sistema, se realizan por cada característica principal del sistema y son utilizadas para cumplir estimaciones de tiempo y el plan de lanzamientos, así mismo reemplazan un gran documento de requisitos y presiden la creación de las pruebas de aceptación, cada historia de usuario debe ser lo suficientemente comprensible y delimitada para que los programadores puedan implementarlas en unas semanas. Para la elaboración de las historias de usuario se tendrá como ejemplo la siguiente Tabla 1.

Tabla 1:*Plantilla para las historias de usuario*

HISTORIA DE USUARIO	
Número: Permite identificar a una historia de usuario.	Usuario: Persona que utilizará la funcionalidad del sistema descrita en la historia de usuario.
Nombre historia: Describe de manera general a una historia de usuario.	
Prioridad en negocio: Grado de importancia que el cliente asigna a una historia de usuario.	Riesgo en desarrollo: Valor de complejidad que una historia de usuario representa al equipo de desarrollo.
Puntos estimados: Número de semanas que se necesitará para el desarrollo de una historia de usuario.	Iteración asignada: Número de iteración, en que el cliente desea que se implemente una historia de usuario.
Programador responsable: Persona encargada de programar cada historia de usuario.	
Descripción: Información detallada de una historia de usuario.	
Observaciones: Campo opcional utilizado para aclarar, si es necesario, el requerimiento descrito de una historia de usuario.	

Nota: En la tabla se muestra la elaboración de las historias de usuario y la explicación de cada uno de sus componentes. Fuente: (Meléndez Valladarez, Gaitan, & Pérez Reyes, 2016)

2.4.1.2. Tareas de ingenierías

Según (Cohn, 2004) Una historia de usuario puede dividirse en múltiples tareas técnicas que permiten una mejor planificación, asignación y seguimiento del trabajo, estas tareas de ingeniería especifican las acciones concretas que deben ejecutarse para implementar adecuadamente la funcionalidad descrita en la historia, entre ellas pueden incluirse actividades como el diseño de interfaces, la codificación, la configuración de bases de datos, las pruebas unitarias y de integración, entre otras. A diferencia de las historias de usuario, que están orientadas a describir el valor desde el punto de vista del usuario, las tareas de ingeniería se relacionan directamente con las actividades técnicas y suelen ser ejecutadas por desarrolladores o ingenieros del equipo, esta vinculación permite una mayor precisión en la estimación del esfuerzo, facilita la asignación de responsabilidades y contribuye al monitoreo efectivo del avance del proyecto, esta descomposición asegura que los objetivos definidos a nivel funcional se traduzcan en pasos técnicos concretos, manteniendo la trazabilidad entre las necesidades del cliente y el producto final, lo que mejora la calidad del proceso y promueve la entrega continua de valor. Para la elaboración de las tareas de ingeniería se tendrá como ejemplo la siguiente Tabla 2.

Tabla 2:*Plantilla para las tareas de ingeniería*

TAREAS DE INGENIERÍA	
Número de tarea: Permite identificar a una tarea de ingeniería.	Número de historia: Número asignado de la historia correspondiente.
Nombre de tarea: Describe de manera general a una tarea de ingeniería.	
Tipo de tarea: Tipo al que corresponde la tarea de ingeniería.	Puntos estimados: Número de días que se necesitará para el desarrollo de una tarea ingeniería.
Fecha inicio: Fecha inicial de la creación de la tarea de ingeniería.	Fecha fin: Fecha final concluida de la tarea de ingeniería.
Programador responsable: Persona encargada de programar la tarea de ingeniería.	
Descripción: Información detallada de la tarea de ingeniería.	

Nota: En la tabla se muestra la elaboración de las tareas de ingeniería y la explicación de cada uno de sus componentes. Fuente: (Meléndez Valladarez, Gaitan, & Pérez Reyes, 2016)

2.4.1.3. Pruebas de aceptación

Según (Crispin & Gregory, 2009) Las pruebas de aceptación, también denominadas pruebas del cliente, son esenciales no solo para verificar el correcto funcionamiento del sistema desde la perspectiva del usuario, sino también para asegurar que el software entregado realmente aporta valor al negocio, esta práctica garantiza que el producto cumpla con los requerimientos funcionales establecidos en las historias de usuario, que se consideran completas únicamente cuando superan satisfactoriamente estas pruebas, una de las principales ventajas de las pruebas de aceptación es que permiten al cliente conocer el estado actual del desarrollo del sistema, facilitando la visibilidad del progreso alcanzado y estableciendo un punto de control claro antes de iniciar una nueva iteración, para el equipo de desarrollo, estas pruebas proporcionan una retroalimentación inmediata sobre el trabajo realizado, así como información útil sobre posibles ajustes, correcciones o cambios que deban considerarse en el diseño de nuevas historias de usuario, la naturaleza colaborativa de estas pruebas promueve una comunicación continua entre el cliente y el equipo de desarrollo, reduciendo la ambigüedad en los requerimientos y fomentando una cultura de mejora continua, desde una perspectiva metodológica, su implementación sistemática fortalece la calidad del producto, reduce los riesgos de malinterpretación de los requerimientos y contribuye a mantener la alineación entre las expectativas del cliente y el resultado final. Para la elaboración de las pruebas de aceptación se tendrá como ejemplo la siguiente Tabla 3.

Tabla 3:*Plantilla para las pruebas de aceptación*

PRUEBAS DE ACEPTACIÓN	
Código: Número único, permite identificar la prueba de aceptación.	Número de historia de usuario: Número único que identifica a la historia de usuario.
Historia de usuario: Nombre que indica de manera general la descripción de la historia de usuario.	
Condiciones de ejecución: Condiciones previas que deben cumplirse para realizar la prueba de ejecución.	
Entrada/Pasos de ejecución: Pasos que siguen los usuarios para probar la funcionalidad de la historia de usuario.	
Resultado esperado: Respuesta del sistema que él cliente espera, después de haber ejecutado una funcionalidad.	
Evaluación de la prueba: Nivel de satisfacción del cliente sobre la respuesta del sistema. Los niveles son Aprobada y No Aprobada.	

Nota: En la tabla se muestra la elaboración de las pruebas de aceptación y la explicación de cada uno de sus componentes. Fuente: (Meléndez Valladarez, Gaitan, & Pérez Reyes, 2016)

2.4.1.4. Tarjetas CRC (Clase – Responsabilidades - Colaboradores)

Según (Beck & Cunningham, 1989) Las tarjetas CRC son una herramienta didáctica y de modelado empleada principalmente en el diseño orientado a objetos, el propósito principal es facilitar el análisis y diseño de software mediante la simulación de escenarios de interacción entre objetos, esta técnica permite distribuir de manera coherente las responsabilidades entre las clases, su uso es valioso en las etapas iniciales del desarrollo, cuando se define la arquitectura lógica del sistema. Para la elaboración de la Tarjetas CRC se tendrá como ejemplo la siguiente Tabla 4.

Tabla 4:*Plantillas para las tarjetas CRC*

PRUEBAS DE ACEPTACIÓN	
Nombre de la clase: Nombre de la clase al cual se hace referencia la tarjeta.	
Responsabilidades: Atributos y operaciones de la clase.	Colaboradores: Clases que colaboran con la clase citada en la tarjeta.

Nota: En la tabla se muestra la elaboración de las tarjetas CRC y la explicación de cada uno de sus componentes. Fuente: (Meléndez Valladarez, Gaitan, & Pérez Reyes, 2016)

2.4.2. Roles de la metodología XP

La Propuesta original según (Meléndez Valladarez, Gaitan, & Pérez Reyes, 2016) incluye los siguientes roles:

- **Programador**, el programador es responsable de implementar las historias de usuario por el cliente, además de diseñar y ejecutar los test de unidad del código que ha implementado o modificado. (López Mendoza, 2020)
- **Cliente**, determina la funcionalidad en cada iteración y define las prioridades de implementación según el valor de negocio. (Beck & Cunningham, 1989)
- **Encargado de pruebas**, es el encargado de ejecutar las pruebas regularmente, difunde los resultados dentro del equipo y es también el responsable de las herramientas de soporte para pruebas. (López Mendoza, 2020)

2.4.3. Fases de la metodología XP

Para (Meléndez Valladarez, Gaitan, & Pérez Reyes, 2016) la Programación Extrema consta de 4 fases, las cuales son:

2.4.3.1. Planificación

La planificación como un diálogo continuo entre las partes involucradas en el proyecto, incluyendo al cliente, a los programadores y a los coordinadores, el proyecto comienza recopilando las historias de usuarios, las que constituyen a los tradicionales casos de uso. Según (Meléndez Valladarez, Gaitan, & Pérez Reyes, 2016) los Conceptos básicos de la planificación son:

- **Historias de usuarios**, las cuales son descritas por el cliente, en su propio lenguaje, como descripciones cortas de lo que el sistema debe realizar. (López Mendoza, 2020)
- **El plan de entregas**, establece que las historias de usuarios serán agrupadas para conformar una entrega y el orden de las mismas. (Beck & Cunningham, 1989)
- **Plan de iteraciones**, las historias de usuarios seleccionadas para cada entrega son desarrolladas y probadas en un ciclo de iteración, de acuerdo al orden preestablecido.
- **Reuniones de seguimiento**, el objetivo es mantener la comunicación entre el equipo y compartir problemas y soluciones. (López Mendoza, 2020)

2.4.3.2. Diseño

La Metodología XP hace especial énfasis en los diseños simples y claros. Los conceptos más importantes para (Meléndez Valladarez, Gaitan, & Pérez Reyes, 2016) son el diseño para esta metodología y son los siguientes:

- **Simplicidad**, un diseño simple se implementa más rápidamente que uno complejo, por ello se propone implementar un diseño que funcione. (López Mendoza, 2020)
- **Soluciones**, cuando aparecen problemas técnicos, o cuando es difícil de estimar el tiempo para implementar una historia de usuario, pueden utilizarse pequeños programas de prueba. (Beck & Cunningham, 1989)
- **Recodificación**, consiste en escribir nuevamente parte del código de un programa, sin cambiar su funcionalidad, a los efectos de crearlo más simple, conciso y entendible. (Meléndez Valladarez, Gaitan, & Pérez Reyes, 2016)

2.4.3.3. Codificación

Los conceptos más importantes de la codificación según (Meléndez Valladarez, Gaitan, & Pérez Reyes, 2016) en esta metodología son los siguientes:

- **Disponibilidad del cliente**, tener al cliente disponible durante todo el proyecto, no solamente como apoyo a los desarrolladores, sino formando parte del grupo, el cliente es fundamental para el desarrollo del proyecto. (López Mendoza, 2020)
- **Uso de estándares**, la programación basada en estándares, de manera que sea fácilmente entendible por todo el equipo. (Beck & Cunningham, 1989)
- **Programación dirigida por las pruebas**, se propone un modelo inverso, primero se escribe los test que el sistema debe pasar, luego el desarrollo debe ser el mínimo necesario para pasar las pruebas previamente definidas y las pruebas a los que se refiere esta práctica, son las pruebas unitarias. (López Mendoza, 2020)
- **Integraciones permanentes**, todos los desarrolladores necesitan trabajar siempre con la “última versión”, publicar lo antes posible las nuevas versiones, aunque no sean las últimas, siempre que estén libres de errores. (Beck & Cunningham, 1989)
- **Propiedad colectiva de código**, todo el equipo puede contribuir con nuevas ideas que apliquen a partes del proyecto, se puede cambiar el código que sea necesario para corregir problemas, agregar funciones o recodificar. (López Mendoza, 2020)

2.4.3.4. Pruebas

Los conceptos más importantes de las pruebas según (Meléndez Valladarez, Gaitan, & Pérez Reyes, 2016) en esta metodología son los siguientes:

- **Pruebas unitarias**, todos los módulos deben de pasar las pruebas unitarias antes de ser liberados o publicados, las pruebas deben ser definidas antes de realizar el código. (Meléndez Valladarez, Gaitan, & Pérez Reyes, 2016)
- **Detección y corrección de errores**, cuando se encuentra un error, éste debe ser corregido inmediatamente, y se deben tener precauciones para que errores similares no vuelvan a ocurrir. (Beck & Cunningham, 1989)
- **Pruebas de aceptación**, son creadas en base a las historias de usuarios, en cada ciclo de la iteración del desarrollo, el cliente debe especificar uno o diversos escenarios para comprobar que una historia de usuario ha sido correctamente implementada. (López Mendoza, 2020)

2.5. ANÁLISIS Y DISEÑO DE SISTEMAS

Según (Senn, 1992) el análisis y diseño de sistemas constituye un pilar metodológico esencial en el desarrollo del presente proyecto, esta disciplina proporciona las herramientas necesarias para planificar, estructurar y documentar de forma sistemática el funcionamiento del sistema propuesto. Desde una perspectiva técnica, el análisis de sistemas permite identificar los requerimientos funcionales y no funcionales del sistema, a partir de un estudio detallado de las necesidades de los usuarios, las condiciones de seguridad de los datos y la naturaleza del entorno institucional, es importante tratar con la información sensible, como la que maneja una academia de historia militar, ya que permite prever el uso de técnicas como el enmascaramiento de datos desde la arquitectura misma del sistema.

El diseño de sistemas, por su parte, posibilita estructurar la solución técnica mediante diagramas de flujo, diagramas de casos de uso, diagramas de clases, entre otros modelos que permiten representar de forma precisa la interacción entre los distintos componentes del sistema, asimismo un buen diseño facilita la comunicación entre desarrolladores, usuarios y otras partes interesadas, permitiendo una mejor comprensión de los objetivos del sistema.

2.5.1. Tipos de sistemas

El libro de (Garfinkel, 2015) nos menciona las relaciones que se tienen con el presente proyecto que hace uso de varios tipos de sistemas, los cuales se integran de manera funcional para alcanzar los objetivos planteados:

- **Sistema de gestión de contenidos (CMS):** El sistema de gestión de contenidos nos permiten crear, modificar, organizar y publicar contenido digital mediante interfaces intuitivas, su principal ventaja es que posibilitan la gestión descentralizada de la información, sin necesidad de conocimientos técnicos avanzados por parte del usuario final, además, estos sistemas promueven la accesibilidad y la estructuración coherente del contenido digital. (Laudon & Laudon, 2021)
- **Sistemas de seguridad informática:** El sistema de seguridad informática comprende mecanismos y políticas orientadas a proteger la información y los recursos tecnológicos frente a accesos no autorizados, alteraciones o pérdidas, en entornos donde se manejan datos sensibles, es crucial y necesario aplicar estrategias como el enmascaramiento de datos, algoritmos de seguridad, entre otras técnicas que buscan anonimizar la información personal del sistema central para evitar su identificación indebida y accesibilidad no autorizada. (Garfinkel, 2015)
- **Sistemas cliente – servidor:** La arquitectura cliente - servidor representa una de las estructuras más utilizadas en el diseño de sistemas distribuidos, consiste en uno o más servidores que brindan servicios y múltiples clientes que los consumen, esta organización mejora la escalabilidad del sistema a futuro, el control de datos que se maneja y la eficiencia en el uso de recursos. (Senn, 1992)
- **Sistemas de control de accesos:** El sistema de control de accesos del sistema administra los derechos de entrada de los usuarios a los recursos de una plataforma tecnológica, que se basa en la definición de roles de usuario y niveles de permisos para garantizar la confidencialidad, integridad y disponibilidad de la información, se considera una herramienta fundamental para limitar el acceso de usuarios a datos críticos muy relevantes. (Laudon & Laudon, 2021)
- **Sistemas de gestión de base de datos:** El sistema de gestión bases de datos está encargado de almacenar, organizar y administrar grandes volúmenes de información, la función principal es asegurar la consistencia, integridad y seguridad de los datos, asimismo permite realizar consultas, actualizaciones y análisis que apoyan la toma de decisiones. (Senn, 1992)

2.5.2. Integración de las tecnologías en sistemas

Según (Pressman, 2021), La integración de tecnologías en sistemas es un proceso estratégico que permite unir diversos componentes de software y hardware para trabajar de manera coordinada, esta práctica asegura que los diferentes subsistemas involucrados en una solución tecnológica operen como una unidad cohesiva, lo cual es fundamental en entornos donde se manejan grandes volúmenes de información y donde la seguridad, la eficiencia y la interoperabilidad son requisitos indispensables, en el desarrollo de sistemas no solo mejora la funcionalidad general del producto, sino que también facilita su escalabilidad, mantenimiento y evolución a lo largo del tiempo. Es importante destacar dos tipos de integración tecnológica que permiten que los sistemas funcionen de forma coordinada y eficaz:

- **Integración de datos:** Consiste en unificar datos procedentes de diversas fuentes en un repositorio común (una base de datos central). Esta integración permite que distintos módulos, como el CMS y el sistema de control de accesos, consulten y actualicen datos en tiempo real, manteniendo la coherencia e integridad de la información. (Laudon & Laudon, 2021)
- **Integración de aplicaciones:** Implica la conexión entre aplicaciones o sistemas de software que necesitan compartir procesos o funcionalidades, por ejemplo, se puede integrar el sistema de gestión de contenidos con módulos de autenticación para que el acceso a ciertos documentos esté restringido según el perfil del usuario. (Sommerville, Software Engineering, 2016)

2.5.3. Roles del analista de sistemas

El analista de sistemas cumple un rol fundamental en todo el ciclo de vida del desarrollo del sistema: desde el levantamiento de requerimientos, pasando por el diseño y evaluación de alternativas, hasta el acompañamiento de la implementación, el analista actúa como un vínculo clave entre las necesidades del usuario y las soluciones técnicas, es un profesional en el proceso de desarrollo de software, encargado de identificar, documentar y traducir las necesidades del usuario en soluciones técnicas viables, su participación garantiza que el sistema propuesto no solo sea técnicamente correcto, sino también funcional, seguro y alineado con los objetivos institucionales. Según (Kendall & Kendall, 2011), los roles principales del analista de sistemas incluyen:

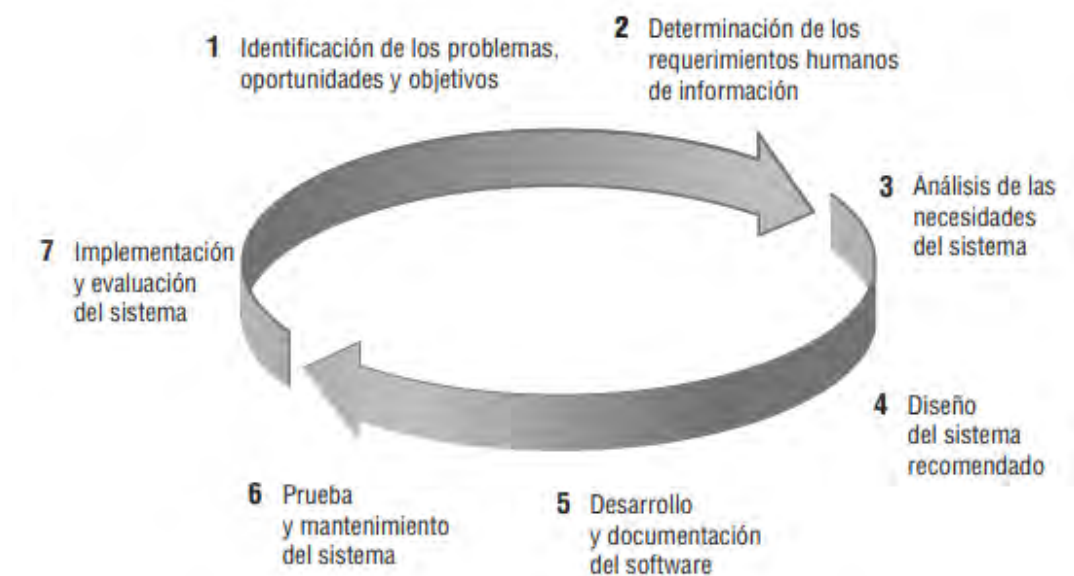
- **Investigador:** Recoge información sobre la organización, sus procesos y los problemas existentes. (Pressman, 2021)
- **Diseñador:** Propone soluciones técnicas que resuelvan los problemas identificados, mediante modelos y especificaciones detalladas. (Laudon & Laudon, 2021)
- **Consultor:** Asesora a los usuarios y toma decisiones sobre tecnologías, procesos y metodologías adecuadas. (Sommerville, Software Engineering, 2016)
- **Agente de cambio:** Participa en la gestión del cambio organizacional que implica la implementación de nuevos sistemas. (Laudon & Laudon, 2021)
- **Comunicador:** Actúa como puente entre los usuarios finales, desarrolladores y otras partes interesadas. (Sommerville, Software Engineering, 2016)
- **Gestor de proyectos:** Coordina actividades, tiempos y recursos involucrados en el desarrollo del sistema. (Pressman, 2021)

2.5.4. El ciclo de vida del desarrollo de sistemas

El Ciclo de Vida del Desarrollo de Sistemas, es una metodología estructurada que guía las fases necesarias para el desarrollo exitoso de un sistema de información. Según (Kendall & Kendall, 2011) el ciclo se divide en siete fases, como se muestra en la Figura 6.

Figura 6:

Las siete fases del ciclo de desarrollo de sistemas



Nota: En la figura se muestra el funcionamiento del ciclo de la vida del desarrollo de sistemas, los analistas mantienen firme que este ciclo es el más útil. Fuente: (Kendall & Kendall, 2011)

- **Identificación de los problemas, oportunidades y objetivos.** - En esta etapa inicial, se analiza el entorno organizacional para identificar necesidades, problemas o áreas de mejora que justifiquen el desarrollo de un sistema, también se definen los objetivos generales del proyecto. (Sommerville, Software Engineering, 2016)
- **Determinación de los requerimientos de información.** - Se recaba información precisa y completa sobre las necesidades del usuario mediante técnicas como entrevistas, cuestionarios, observación directa y análisis de documentos, el objetivo es definir qué debe hacer el sistema. (Laudon & Laudon, 2021)
- **Análisis de las necesidades del sistema.** - En esta fase se modelan los procesos actuales y se identifican las deficiencias del sistema existente, esto permite establecer los requerimientos funcionales y no funcionales del sistema. (Pressman, 2021)
- **Diseño del sistema recomendado.** - Se propone una solución detallada mediante diagramas, especificaciones técnicas y modelos visuales, incluyendo el diseño de la base de datos, interfaces de usuario, flujos de procesos. (Laudon & Laudon, 2021)
- **Desarrollo y documentación del software.** - Se procede a codificar el sistema conforme al diseño propuesto, utilizando herramientas y lenguajes de programación adecuados, paralelamente, se genera documentación técnica. (Pressman, 2021)
- **Prueba y mantenimiento del sistema.** - Se realizan pruebas para verificar que el sistema funcione correctamente, de acuerdo con los requerimientos establecidos, una vez implementado, se da paso al mantenimiento para corregir errores, realizar mejoras y adaptarlo a las necesidades. (Sommerville, Software Engineering, 2016)
- **Implementación y evaluación del sistema.** - Una vez que el sistema ha sido implementado y utilizado, se evalúa su desempeño para determinar si cumple con los objetivos organizacionales, esta fase también permite identificar mejoras o nuevas necesidades que podrían motivar futuras versiones o desarrollos. (Pressman, 2021)

2.6. DISEÑO DE PROCESOS

Según (Kendall & Kendall, 2011), el diseño de procesos constituye una herramienta fundamental en el análisis y desarrollo de sistemas de información, ya que permite modelar gráficamente el flujo de actividades, decisiones, entradas y salidas dentro del sistema, este enfoque contribuye a visualizar con claridad cómo se ejecutan los procesos organizacionales, facilitando la identificación de redundancias y oportunidades de mejora.

El diseño de procesos implica la especificación detallada de las actividades que componen un sistema y cómo interactúan entre sí, siendo una guía para estructurar la solución tecnológica y alinear los requerimientos del sistema con los objetivos institucionales.

2.6.1. Pasos para un diseño de procesos

Según (Kendall & Kendall, 2011), para realizar un diseño de procesos correctos se deben seguir los siguientes pasos:

- **Identificación de los procesos clave:** Se define qué procesos forman parte esencial del sistema a desarrollar. (Negocios Atalaya, 2023)
- **Levantamiento de información:** Se recogen datos sobre cómo se realizan actualmente los procesos o cómo se espera que se hagan, mediante observaciones o revisión documental. (Falcon, 2022)
- **Análisis de los procesos existentes:** Se estudian los procesos identificados para detectar problemas, ineficiencias, redundancias o mejoras. (Kendall & Kendall, 2011)
- **Modelado de procesos:** Se crean diagramas o representaciones gráficas, diagramas de flujo, o mapas de procesos para visualizar cada paso. (Belcic & Stryker, 2024)
- **Rediseño o mejora del proceso:** Se proponen mejoras o automatizaciones, buscando mayor eficiencia, seguridad y alineación con el sistema. (Kuuse, 2024)
- **Validación del proceso:** Se revisa el diseño con los usuarios o responsables para confirmar los cambios con sus expectativas y necesidades. (Kendall & Kendall, 2011)
- **Documentación del proceso:** Se elaboran descripciones claras de cada proceso, sus entradas, salidas, responsables y reglas de negocio, para facilitar su implementación y mantenimiento. (Falcon, 2022)

2.7. LENGUAJE UNIFICADO DE MODELADO (UML)

2.7.1. Definición de UML

El Lenguaje Unificado de Modelado (UML) es utilizado para visualizar, especificar, construir y documentar los componentes de un sistema de software, especialmente en entornos orientados a objetos, se aplica como herramienta metodológica clave para representar de forma estructurada y comprensible el diseño del sistema propuesto.

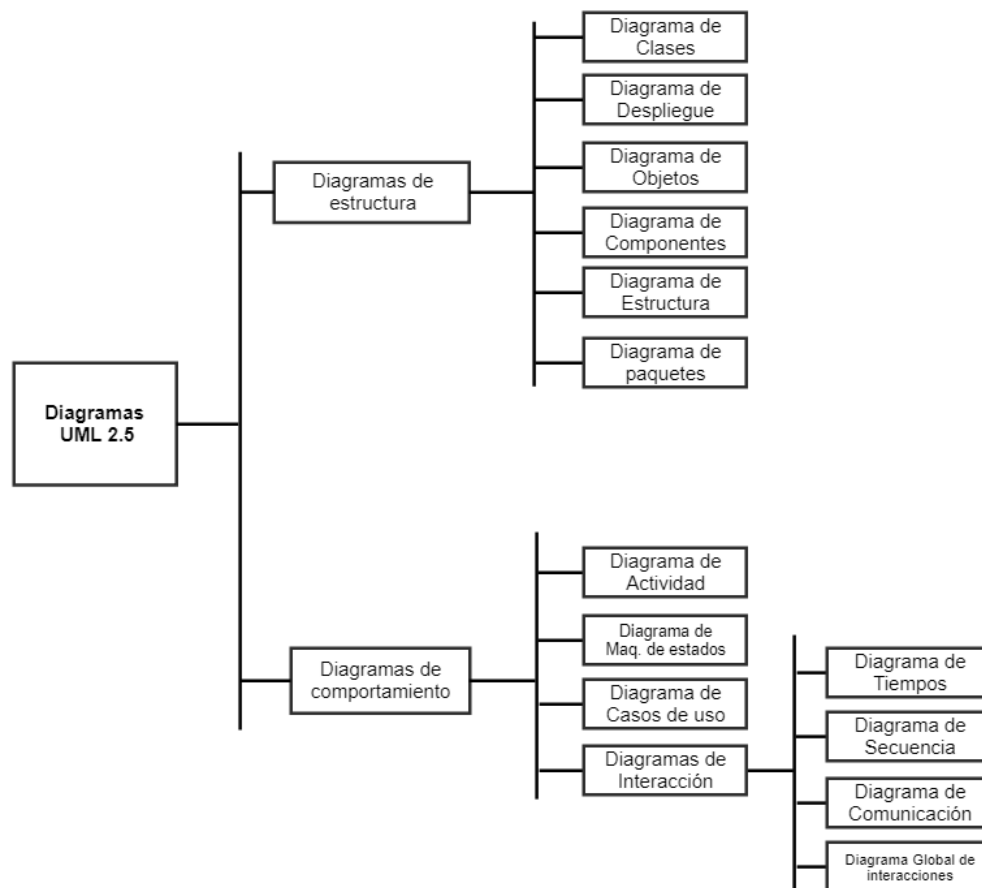
El uso de UML contribuye a mejorar la comunicación entre los miembros del equipo de desarrollo y los usuarios finales, facilita la toma de decisiones durante el diseño técnico, y proporciona una documentación estandarizada que favorece la escalabilidad y el mantenimiento futuro del sistema, mediante diagramas como los de casos de uso, clases, secuencia y actividades, se logra una representación integral y coherente de la arquitectura del sistema. (Booch, Rumbaugh, & Jacobson, 1998)

2.7.2. Tipos de diagramas UML

Existen dos clasificaciones de diagramas: Los diagramas estructurales y los diagramas de comportamiento. Así como se ve en la Figura 7 (Lopez, 2013)

Figura 7:

Clasificación de los diagramas UML



Nota: En la figura se muestran los diagramas UML dividido en dos tipos de diagramas que son los estructurados que contienen seis diagramas diferentes y los diagramas de comportamiento que se dividen en cuatro diagramas diferentes, estos diagramas realizan acciones diferentes y que se enfocan de distinta manera cuando se los realiza. Fuente: (Lopez, 2013)

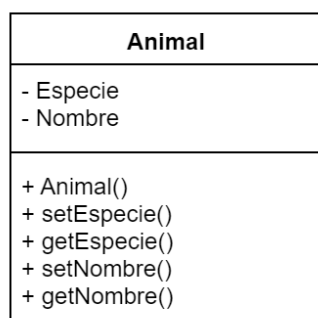
2.7.3. Diagrama de clases

El diagrama de clases según (López, Diagrama de clases: Diagramas UML, 2020) es clasificado dentro de los diagramas de estructura y, como tal, se utiliza para representar los elementos que componen un sistema de información desde un punto de vista estático. Este diagrama está formado por dos elementos: clases y relaciones.

Clases, este elemento normalmente se utiliza para representar conceptos o entidades del negocio, define un grupo de objetos que comparten características, condiciones y significado, es la manera más rápida para encontrar clases sobre un enunciado. Una clase está compuesta por tres elementos: nombre de la clase, atributos y funciones, el cual se muestra en la Figura 8. Los atributos como las funciones pueden ser (+) públicas, (-) privadas y (#) protegidas:

Figura 8:

Ejemplo de una clase



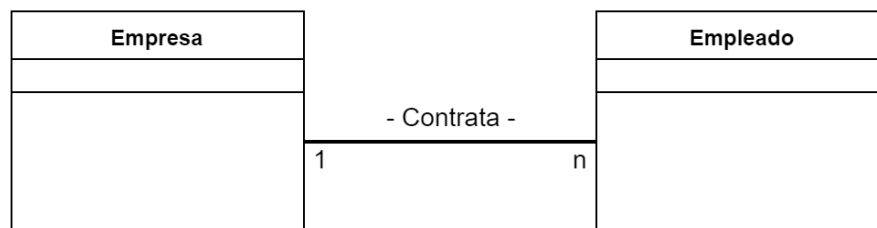
Nota: En la figura se muestra la estructura de una clase y como se componen en tres partes que es el nombre de la clase, los atributos y las funciones. Fuente: (López, Diagrama de clases: Diagramas UML, 2020)

Relaciones, una relación identifica una dependencia, esta dependencia puede ser entre dos o más clases o una clase, así como se ve en la Figura 9, las relaciones se representan con una línea que une las clases, esta línea variará dependiendo del tipo de relación. Estas propiedades son las siguientes:

- **Multiplicidad**, es el número de elementos de una clase que participan en una relación.
- **Nombre de la asociación**, es una indicación de la asociación que ayuda a la relación que tienen dos clases.

Figura 9:

Ejemplo de relación empresa - empleado

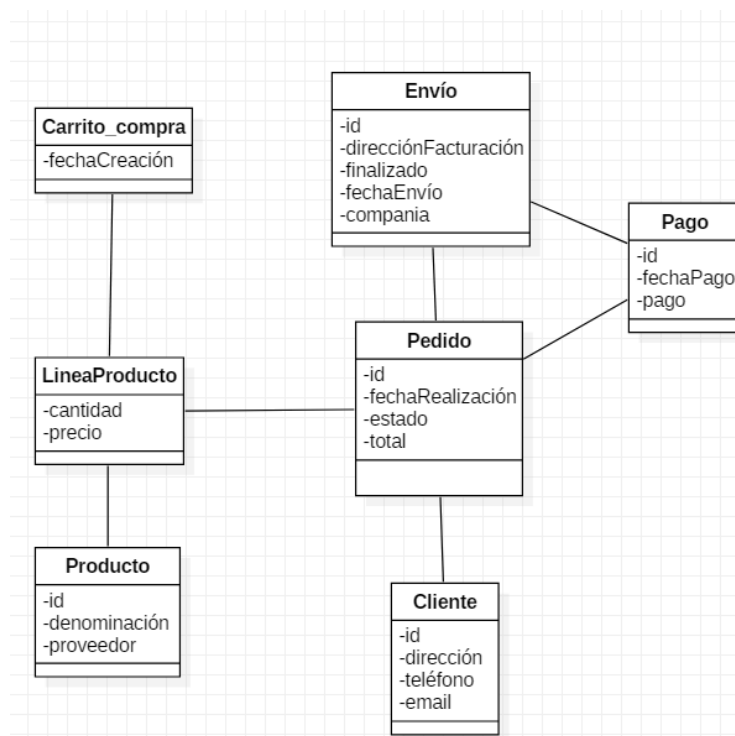


Nota: En la figura se muestra un ejemplo del funcionamiento de las relaciones de como una empresa se relaciona con varios empleados por medio del contrato. Fuente: (López, Diagrama de clases: Diagramas UML, 2020)

Para entender mejor la estructura de un diagrama de clases completo y funcional, se puede ver la Figura 10 donde se ve un ejemplo de un diagrama de clase para una tienda.

Figura 10:

Ejemplo de diagrama de clases



Nota: En la figura se muestra un ejemplo de una tienda virtual para comprar un producto en línea, realizando un pedido mediante un envío por parte del cliente y realizar un pago. Fuente: (López, Diagrama de clases: Diagramas UML, 2020)

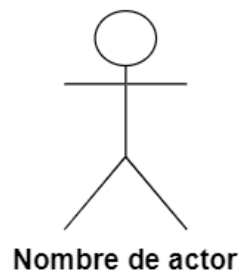
2.7.4. Diagrama de casos de uso

El diagrama de casos de uso según (López, Diagrama de casos de uso: Diagramas UML, 2020) es más conocido y es utilizado para representar los actores externos que interactúan con el sistema de información y a través de qué funcionalidades se relacionan, dicho de otra manera, muestra de manera visual las distintas funciones que puede realizar uno o más usuarios de un sistema de información. El diagrama de casos de uso, dependiendo de la profundidad que le demos, puede ser utilizado para muchos fines, un diagrama de casos de uso está compuesto, principalmente, de tres elementos: Actores, Casos de uso y Relaciones.

Actores, como ya hemos comentado en la presentación, un actor es algo o alguien externo al sistema que interactúa de forma directa con el sistema, cuando decimos que interactúa nos referimos a que aporta información, recibe información, inicia una acción y se representan con una imagen de un “muñeco de palo” con el nombre del actor debajo, como se muestra en la Figura 11.

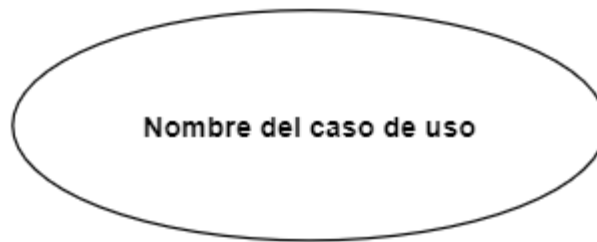
Figura 11:

Representación de un actor



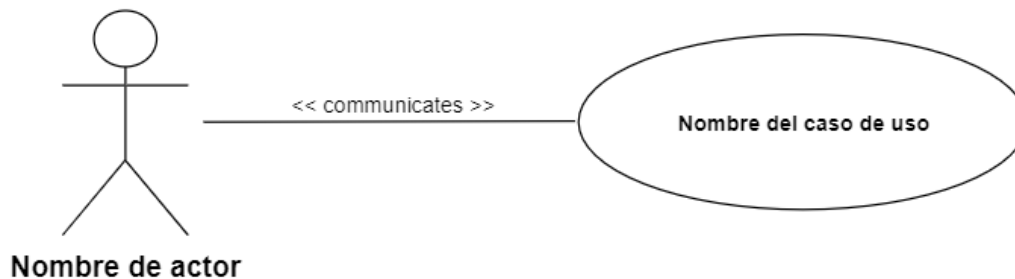
Nota: En la figura se muestra la representación de un actor que aporta y recibe información además de iniciar una acción y se pone el nombre por debajo. Fuente: (López, Diagrama de casos de uso: Diagramas UML, 2020)

Casos de uso, se utiliza para representar una de las funcionalidades que realiza el sistema, es lo que indica (López, 2020) es una secuencia de acciones que hace el sistema, un caso de uso es una clasificación de comportamiento que especifica una unidad de funcionalidad completa y que está realizada por uno o más sujetos que se relacionan con el caso de uso colaborando para ello con uno o más actores y que produce un resultado que tiene alguna utilidad para cualquier de esos actores como se muestra en la Figura 12.

Figura 12:*Representación de un caso de uso*

Nota: En la figura se muestra la representación del caso de uso y se utiliza para representar las funcionalidades que realiza el sistema. Fuente: (López, Diagrama de casos de uso: Diagramas UML, 2020)

Relaciones, cuando se conectan un actor con un caso de uso representa que ese actor interactúa de alguna manera con ese caso de uso y se representa con una línea continua con la identificación. (López, Diagrama de casos de uso: Diagramas UML, 2020)

Figura 13:*Ejemplo de relación actor - caso de uso*

Nota: En la figura se muestra la representación de la conexión entre el actor y el caso de uso. Fuente: (López, Diagrama de casos de uso: Diagramas UML, 2020)

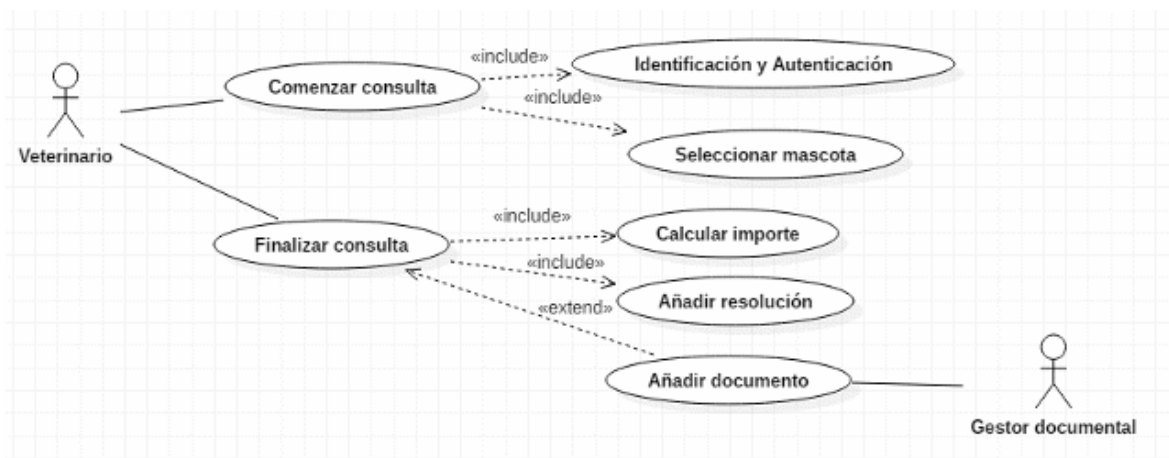
Cuando conectan casos de uso entre sí se pueden diferenciar dos tipos de relaciones: `<<include>>` y `<<extends>>`.

- **`<<include>>`**: Se utiliza para representar que un caso de uso utiliza siempre a otro caso de uso o depende de otro caso de uso, es decir, un caso de uso se ejecutará obligatoriamente ya que lo requiere necesariamente. (Megino Barquinero, 2013)
- **`<<extends>>`**: Se utilizan cuando un caso de uso tiene un comportamiento opcional, reflejado en otro caso de uso, es decir, un caso de uso puede ejecutar, normalmente dependiendo de alguna condición o flujo del programa. (Paradigm, 2022)

Para entender mejor la estructura de un diagrama de casos de uso, ver la Figura 14 donde se ve un ejemplo de un diagrama de casos de uso donde el actor es un Veterinario y Gestor documental.

Figura 14:

Ejemplo de diagrama de casos de uso



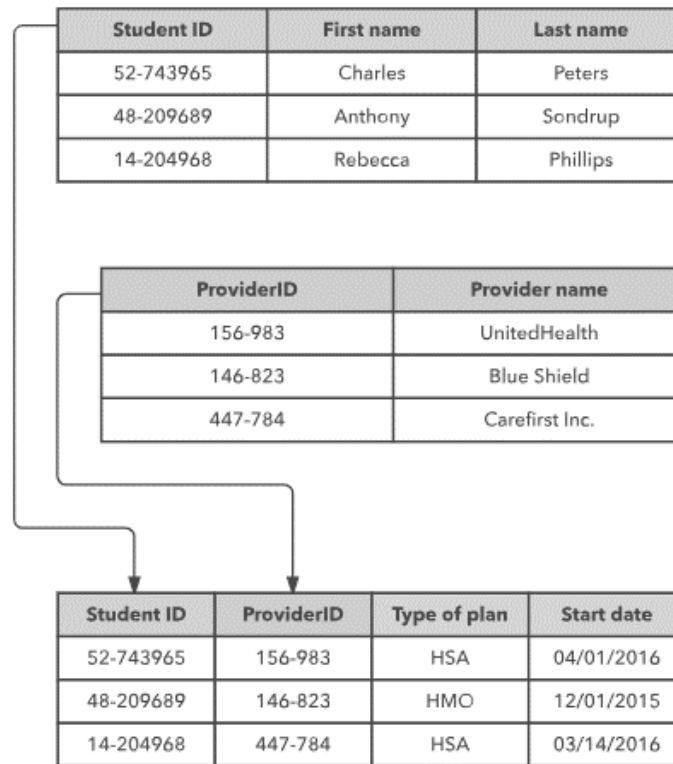
Nota: En la figura se muestra un ejemplo con dos actores que son veterinario que realiza consultas y revisión de mascotas y un gestor documental para verificar los importes y el registro de la consulta. Fuente: (López, Diagrama de casos de uso: Diagramas UML, 2020)

2.8. MODELOS DE BASES DE DATOS

Una base de datos se define como un conjunto organizado de datos que se almacenan y gestionan electrónicamente, permitiendo su recuperación, actualización y administración de manera eficiente, su implementación es esencial para garantizar que la información estructurada que maneja el sistema (como usuarios, contenido, niveles de acceso y logs de actividad) esté almacenada de forma segura, coherente y accesible. (Coronel & Morris, 2019)

2.8.1. Modelo Relacional

El modelo más común se ven en la Figura 15, donde se ordena los datos en tablas, también conocidas como relaciones, cada una de las cuales se compone de columnas y filas además cada columna enumera un atributo como clave primaria, a la cual se puede hacer referencia en otras tablas, en donde se la denomina clave externa. (Lucidchart, 2023)

Figura 15:*Ejemplo de un modelo relacional*

Nota: En la figura se muestra un ejemplo del modelo relacional y se ordenan las claves primarias de estudiantes y proveedor en la tercera, pero pasando como llaves foráneas. Fuente: (Lucidchart, 2023)

2.9. INTERFAZ DE USUARIO

2.9.1. Definición de la interfaz de usuario (UI)

La interfaz de usuario según (Lenis, 2023) es el medio por el cual una persona controla una aplicación de software o dispositivo de hardware, es decir que el programa incluye controles gráficos que optimizan la experiencia de usuario al emplear un mouse o teclado, lo que posibilita la interacción con los procesadores para realizar un trabajo.

2.9.2. Características de la interfaz de usuario (UI)

El balance de estos elementos resulta en una interfaz que ayuda al usuario a realizar el trabajo para el cual fue diseñada una aplicación o un programa de cómputo, esto nos indica (Lenis, 2023).

- **Atractivo visual**, la apariencia debe ser una prioridad de los equipos de desarrollo de interfaz del usuario, ya que permite que el usuario se sienta identificado y cómodo con el programa. (The Bridge, 2024)
- **Claridad**, es la manera de transmitir la información al usuario debe ser clara y concisa para evitar errores al momento de interactuar el software. (Corrales, 2019)
- **Coherencia**, los usuarios pueden crear patrones de uso de forma intuitiva, sin la necesidad de aprender procesos muy complejos que podrían desinhibir su uso de tecnología. (Rodríguez Mira, 2025)
- **Flexibilidad**, es una herramienta que puede adaptarse a las necesidades del usuario favoreciendo su éxito en el futuro. (Lara Galicia, 2024)

2.9.3. Tipos de interfaces de usuario

Se tomarán en cuenta las interfaces más utilizadas entre la interacción del usuario con las interfaces, para esto (Lenis, 2023) menciona las siguientes:

- **Interfaz de lenguaje natural**, es una transacción de la computadora hacia la persona, y el lenguaje de acción, que se caracteriza por ser la interacción de la persona con la computadora. (Lenis, 2023)
- **Interfaz de preguntas y respuestas**, es una de las interfaces más utilizadas por los consumidores: esta les muestra una pregunta en la pantalla y, a través del teclado o haciendo clic con el mouse, la aplicación les permite responder, según la respuesta recibida, la computadora actúa de una manera preprogramada. (Lenis, 2023)
- **Interfaz gráfica de usuario**, se utilizan imágenes, iconos y menús para mostrar las acciones disponibles en un dispositivo, entre las que un usuario puede escoger, las cuales reemplazan comandos de texto y código binario por elementos gráficos, hoy utilizados en la mayoría de equipos. (Lenis, 2023)

2.10. FRAMEWORK DE DESARROLLO

Un framework de desarrollo es una estructura de trabajo predefinida, compuesta por herramientas, bibliotecas, buenas prácticas y convenciones que facilitan y agilizan la construcción de aplicaciones de software. Según (Fowler, 2003) esto nos ayuda a los desarrolladores a mantener la calidad, coherencia y escalabilidad de los proyectos.

2.10.1. Laravel

El creador (Taylor, 2015) dice que Laravel es un framework de desarrollo web en PHP que sigue el patrón arquitectónico Modelo – Vista - Controlador (MVC). Se caracteriza por su sintaxis elegante y fácil de entender, así como por la extensa gama de herramientas que ofrece para el desarrollo de aplicaciones web de alta calidad. Entre sus características más destacadas se encuentran:

- **Eloquent ORM:** Laravel facilita el manejo de bases de datos mediante su potente ORM, permitiendo realizar consultas de bases de datos. (Taylor, 2015)
- **Autenticación y control de acceso:** Laravel proporciona mecanismos preconfigurados para la autenticación de usuarios, registros, restablecimiento de contraseñas y control de accesos. (Cosmin, 2025)
- **Migraciones de base de datos:** Laravel incluye un sistema de migraciones, lo que permite realizar cambios en la estructura de la base de datos de manera fácil, lo que es esencial en un proyecto de largo plazo donde se pueden hacer actualizaciones periódicas a la base de datos. (Arana Flores, 2023)
- **Rutas y controladores:** Facilita la creación de rutas y la asignación de funciones a estas rutas mediante controladores. (Taylor, 2015)
- **Testing y herramientas de depuración:** Laravel incluye un conjunto de herramientas para realizar pruebas de la aplicación y facilitar la depuración. (Taylor, 2015)

2.11. LENGUAJES Y HERRAMIENTAS DE PROGRAMACIÓN

En este punto estaremos mencionando todos los lenguajes de programación y herramientas que utilizaremos para nuestro proyecto, haciendo conocer sus conceptos necesarios y como se utilizan.

2.11.1. HTML

HTML es el lenguaje estándar de marcado utilizado para la creación y estructuración de contenido en la web, no es un lenguaje de programación, sino un sistema de etiquetado que define la disposición y semántica del contenido en una página web, como títulos, párrafos, listas, imágenes, formularios y enlaces. (Duckett, 2011)

2.11.2. CSS

CSS es un lenguaje de hojas de estilo que se usa para definir la apariencia visual de los elementos HTML, permite separar el contenido de la presentación, facilitando el diseño responsive y la reutilización de estilos, controla propiedades como colores, tamaños, márgenes, tipografías y posicionamiento en pantalla, lo que lo convierte en un componente fundamental para el diseño web moderno. (Meyer, 2011)

2.11.3. JAVASCRIPT

JavaScript es un lenguaje de programación interpretado, orientado a objetos y basado en eventos, que se ejecuta principalmente del lado del cliente (aunque puede funcionar en el servidor con tecnologías como Node.js). Permite agregar interactividad, validar formularios, manipular el DOM, realizar peticiones asíncronas y construir interfaces dinámicas, es esencial para mejorar la experiencia del usuario en aplicaciones web. (Flanagan, 2020)

2.11.4. PHP

PHP es un lenguaje de programación del lado del servidor, ampliamente utilizado para el desarrollo de aplicaciones web dinámicas, que permite la conexión con bases de datos, la gestión de sesiones, el envío de formularios y la generación de contenido dinámico, además que se integra fácilmente con HTML y bases de datos como MySQL, lo que lo hace ideal para proyectos de tipo CMS (Content Management System). (Welling & Thomsom, 2009)

2.11.5. MySQL

Según (Robledano, 2019) MySQL es el sistema de gestión de bases de datos relacional más extendido en la actualidad al estar basado en código abierto, que cuenta con una doble licencia. Por una parte, es de código abierto, pero por otra, cuenta con una versión comercial gestionada por la compañía Oracle.

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:

- **Arquitectura Cliente y Servidor:** Su funcionamiento es un modelo cliente y servidor, cada cliente puede hacer consultas a través del sistema de registro para obtener datos, modificarlos, guardar estos cambios en nuevas tablas. (Cardona, 2024)
- **Compatibilidad con 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. (Robledano, 2019)
- **Vistas:** Ofrece una compatibilidad para poder configurar vistas personalizadas del mismo modo que podemos hacerlo en otras bases de datos de gran tamaño y las vistas se hacen un recurso imprescindible. (Cardona, 2024)
- **Procedimientos almacenados:** Posee la característica de no procesar las tablas directamente, sino que a través de procedimientos almacenados es posible incrementar la eficacia de nuestra implementación. (Robledano, 2019)
- **Desencadenantes:** Permite automatizar ciertas tareas dentro de nuestra base de datos, en el momento que se produce un evento otro es lanzado para actualizar registros u optimizar su funcionalidad. (Oluseye, 2024)
- **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, el sistema opta por preservar la integridad de la base de datos. (Cardona, 2024)

También (Robledano, 2019) dice MySQL comparte las sentencias del lenguaje SQL. Lo que en la práctica se traduce en una plena compatibilidad. Al ser un elemento muy ligado al backend de cualquier aplicación web, normalmente desarrollada sobre algún lenguaje como PHP, puede ser un poco difícil de ejemplificar pues sería necesario abarcar todo un artículo, pero se puede realizar la siguiente relación de sentencias básicas:

- **SELECT** es usada para consultar datos.
- **DISTINCT** sirve para eliminar los duplicados de las consultas de datos.
- **WHERE** es usada para incluir las condiciones de los datos que queremos consultar.

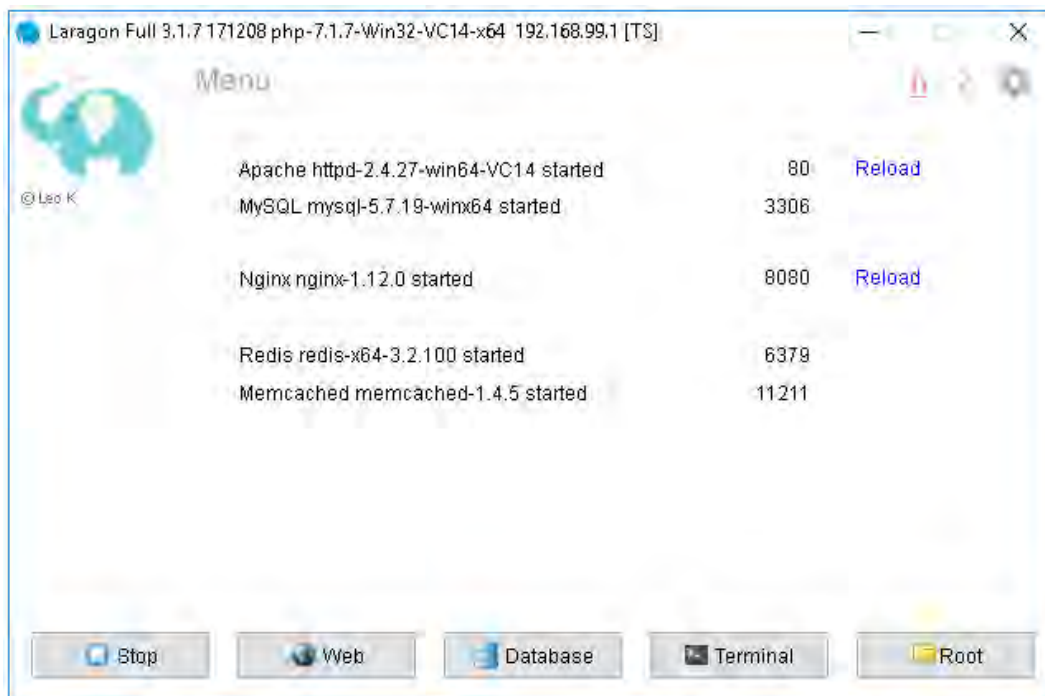
- **AND y OR** es usada para incluir 2 o más condiciones a una consulta.
- **ORDER BY** es usada para ordenar los resultados de una consulta.
- **INSERT** es usada para insertar datos.
- **UPDATE** es usada para actualizar o modificar datos ya existentes.
- **DELETE** Es usada para borrar datos.

2.11.6. Laragon

(Laragon, 2018) Laragon es un entorno de desarrollo universal portátil, aislado, rápido y potente para PHP, Node.js, Python, Java, Go, Ruby. Es rápido, liviano, fácil de usar y fácil de ampliar. Es excelente para crear y administrar aplicaciones web modernas que se centra en el rendimiento y está diseñado en torno a la estabilidad, la simplicidad, la flexibilidad y la libertad, es muy liviano y se mantendrá lo más delgado posible, el binario central en sí tiene menos de 2 MB y utiliza menos de 4 MB de RAM cuando se ejecuta. Laragon no utiliza los servicios de Windows, ya que administra sus propios servicios de forma asincrónica y sin bloqueos, por lo que encontrará que todo funciona rápido y sin problemas. En la Figura 17.

Figura 16:

Interfaz de inicio de Laragon



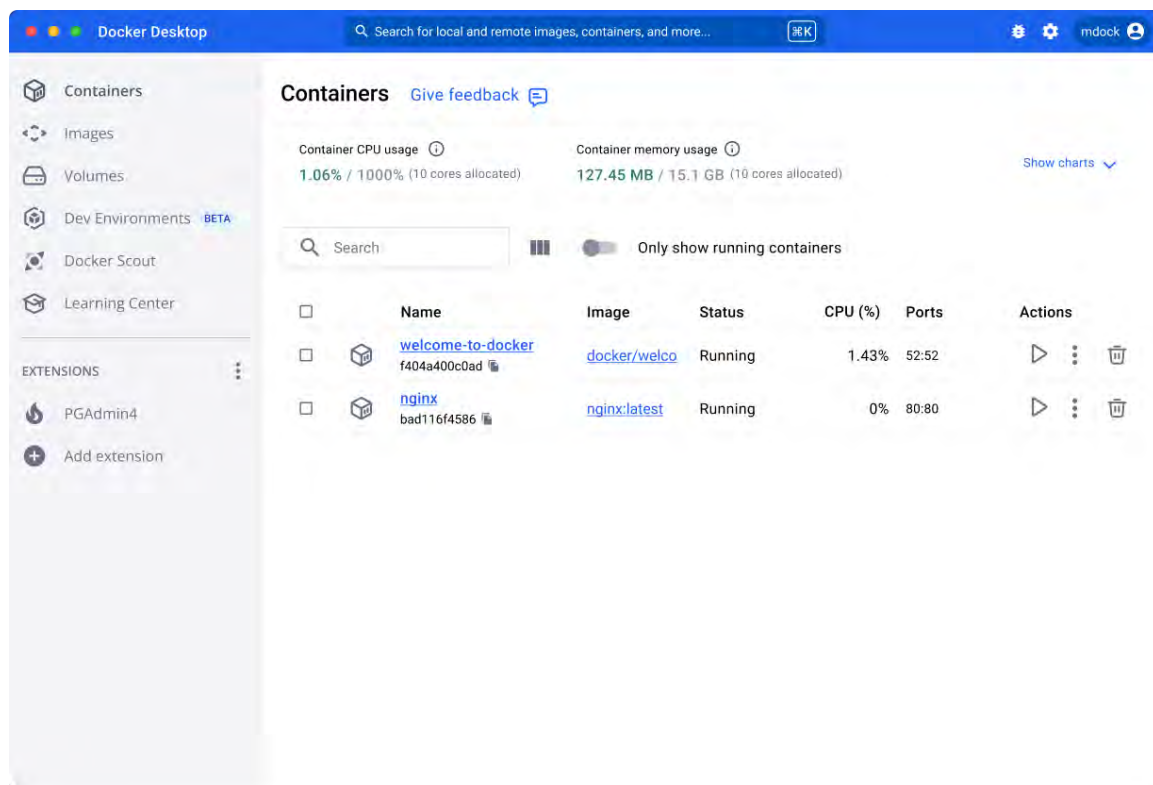
Nota: En la figura se muestra la interfaz de Laragon que está encendida y como se puede ver el Apache inicia en el puerto 80 y MySQL en el puerto 3306. Fuente: (Laragon, 2018)

2.11.7. Docker

Docker es una plataforma de código abierto que permite empaquetar aplicaciones y sus dependencias en contenedores aislados, a diferencia de las máquinas virtuales tradicionales, Docker permite ejecutar múltiples contenedores en el mismo sistema operativo, optimizando recursos y su uso garantiza que una aplicación se comporte igual en todos los entornos, desde desarrollo hasta producción, esto es especialmente útil para mantener coherencia en equipos de trabajo y para despliegues automatizados. Los contenedores son una unidad de software estandarizada que permite a los desarrolladores aislar su aplicación de su entorno, solucionando así el problema de "funciona en mi equipo", para millones de desarrolladores hoy en día, Docker es el estándar de facto para crear y compartir aplicaciones en contenedores, desde el escritorio hasta la nube, estamos desarrollando nuestra experiencia única conectada, desde el código hasta la nube, para desarrolladores y equipos de desarrollo. (Merkel, 2014)

Figura 17:

Interfaz de Inicio de Docker



Nota: En la figura se muestra la interfaz de Docker que está encendida y como se puede ver los Container de inicio que muestra. Fuente: (Merkel, 2014)

2.12. SEGURIDAD INFORMÁTICA

Según (Garfinkel, 2015), la seguridad informática es un campo integral que cubre diversos aspectos y capas de protección dentro de los sistemas tecnológicos, esto implica una combinación de procesos, herramientas, protocolos y políticas que se utilizan para resguardar la información digital de amenazas y ataques externos e internos, estas amenazas pueden surgir de diversas fuentes, tales como hackers, malwares, errores humanos, desastres naturales, o incluso vulnerabilidades en el software. Los componentes clave de la seguridad informática incluyen:

- **Confidencialidad:** Garantiza que la información solo sea accesible para aquellas personas o entidades que están autorizadas, esto se logra mediante técnicas como encriptación, control de acceso y políticas de privacidad. (ESG Innova, 2018)
- **Integridad:** Se refiere a la precisión y fiabilidad de los datos que asegura que la información no sea modificada, alterada o destruida sin la debida autorización, cuyas técnicas son utilizadas para garantizar la integridad de verificación, firmas digitales y protocolos de hash. (Garfinkel, 2015)
- **Disponibilidad:** Asegura que la información y los sistemas estén accesibles cuando se necesiten, para ello se utilizan sistemas de respaldo, planes de recuperación ante desastres, y tecnologías de alta disponibilidad. (ESG Innova, 2018)
- **Autenticación:** El proceso de verificar la identidad de un usuario o sistema antes de otorgar acceso. (Garfinkel, 2015)
- **Control de acceso:** Determina qué usuarios o sistemas pueden acceder a ciertos recursos con listas de control de acceso, roles de usuarios. (ESG Innova, 2018)
- **Monitoreo y auditoría:** Implica el seguimiento continuo de los sistemas y redes para detectar actividades inusuales o sospechosas. Las auditorías también son necesarias para verificar el cumplimiento de las políticas de seguridad. (Garfinkel, 2015)
- **Protección contra amenazas:** Implica el uso de diversas herramientas de seguridad, como firewalls, antivirus, sistemas de detección y prevención de intrusiones (IDS/IPS), y actualizaciones de seguridad para prevenir ataques. (ESG Innova, 2018)
- **Encriptación:** Es un proceso clave para proteger los datos durante su transmisión o almacenamiento, haciendo que la información solo sea legible para aquellos con la clave de desencriptación. (Garfinkel, 2015)
- **Prevención y respuesta a incidentes:** Incluye estrategias y procedimientos para prevenir ataques y mitigar su impacto en caso de que ocurran. (ESG Innova, 2018)

2.12.1. Enmascaramiento de datos

El enmascaramiento de datos es una técnica de seguridad avanzada que transforma la información sensible de manera que no pueda ser utilizada de forma indebida, pero que sigue siendo útil para procesos como pruebas de desarrollo, análisis de datos y simulaciones de datos que implica la modificación de los datos originales para generar valores ficticios que se asemejan a los datos reales, este proceso asegura que los datos sensibles no sean revelados a personas no autorizadas, y se puede hacer de forma reversible o irreversible, dependiendo de la necesidad del sistema. (Garfinkel, 2015)

Una de las aplicaciones más comunes del enmascaramiento es en el ámbito de desarrollo y pruebas de software, donde los desarrolladores necesitan trabajar con datos similares a los reales, pero sin tener acceso a la información personal, este proceso se realiza mediante algoritmos que sustituyen datos reales con valores generados que mantienen el formato y la consistencia del tipo de dato, pero que no contienen información sensible.

2.12.1.1. Tipos de enmascaramiento de datos

Según (Garfinkel, 2015), existen tres tipos de enmascaramiento de datos que son los que se han utilizados con más frecuencia y son más factibles al momento de implementarlos en el desarrollo y son los siguientes:

- **Enmascaramiento Estático de Datos (SDM):** En este enfoque, los datos sensibles son reemplazados por datos ficticios y almacenados en una base de datos separada o sobrecargada, esta versión modificada es utilizada para pruebas y desarrollo, y los datos reales permanecen protegidos. (Garfinkel, 2015)
- **Enmascaramiento Dinámico de Datos (DDM):** A diferencia del enmascaramiento estático, el enmascaramiento dinámico modifica los datos en tiempo real cuando se consultan o acceden, sin alterar los datos almacenados, este enfoque es adecuado para situaciones en las que se necesita que los datos sigan siendo consistentes y utilizables en su formato original. (Amazon AWS, 2023)
- **Enmascaramiento Reversible de Datos:** Algunos métodos de enmascaramiento permiten que los datos sean restaurados a su forma original utilizando una clave o proceso específico, este tipo de enmascaramiento es útil cuando se requiere un acceso temporal o limitado a los datos originales. (Garfinkel, 2015)



CAPÍTULO 3.

MARCO APLICATIVO

3.1. INTRODUCCIÓN

La realización exitosa de cualquier proyecto de grado implica la elección cuidadosa de una metodología de desarrollo que no solo se adecúe a los requisitos del sistema, sino que también permita una adaptabilidad eficiente a lo largo de todo el proceso, en este capítulo, se optó por la metodología de Programación Extrema (XP), reconocida por su agilidad y habilidad para abordar los desafíos del desarrollo de software, la elección de XP se sustenta en su enfoque iterativo y colaborativo, que se alinea de manera óptima con los objetivos del presente trabajo de grado.

El desarrollo del proyecto se estructuró en un ciclo de vida compuesto por cinco fases fundamentales: planificación, diseño, codificación, pruebas y validación; cada una de estas etapas desempeñó un papel esencial en la construcción y evolución continua del sistema, brindando un sólido marco para la toma de decisiones y la gestión eficaz de los recursos. Estas fases nos permitieron realizar un progreso ordenado del proyecto, también nos facilitaron la identificación temprana de oportunidades de mejora, contribuyendo así a la optimización constante del desarrollo.

En la fase de planificación se estableció las bases para el proyecto, definiendo la situación actual del caso de estudio, los objetivos, los alcances y los recursos necesarios; en la fase de diseño se creó una arquitectura sólida y escalable, garantizando la coherencia entre las diferentes partes del sistema; la codificación se llevó a cabo con un enfoque iterativo, facilitando la adaptación a cambios y mejoras constantes, continuando con la fase de pruebas se desempeñó un papel esencial en la identificación y corrección temprana de posibles errores, asegurando la calidad del software y culminando con la etapa de validación se basó en un exhaustivo caso de estudio, donde los problemas identificados y las soluciones propuestas fueron sometidos a una evaluación rigurosa.

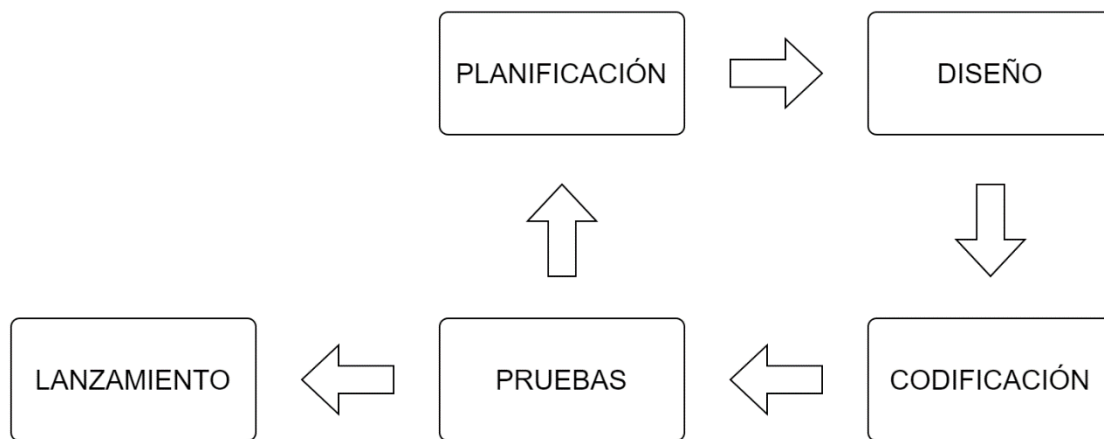
Los resultados de este proceso fueron validados mediante un análisis detallado del caso de estudio, que puso de manera concreta la eficacia de la metodología, en la resolución de los problemas planteados, en las fases de desarrollo que fueron destacando los límites, los alcances y las soluciones propuestas en el transcurso de la elaboración del producto final, ofreciendo una mejor comprensión del sistema ofreciendo funciones efectivas y aplicables en el contexto específico de este proyecto, consolidando así su contribución al producto final plenamente funcional.

3.2. PLANIFICACIÓN DEL MARCO APLICATIVO

Para el desarrollo del sistema de gestión de contenidos se debe tomar en cuenta la facilidad que se necesita al momento de utilizar el sistema. Utilizando la metodología XP debemos entender cómo funcionan las diferentes fases que se presenta en la siguientes Figura 18.

Figura 18:

Funcionamiento y fases de la metodología XP



Nota: En la figura se muestra como son las fases de la metodología XP y el correcto funcionamiento. Fuente: Elaboración Propia

3.2.1. Entrevistas con el cliente

3.2.1.1. Entrevista general

En la entrevista tuvimos el agrado de conocer al Coronel Gary Mercado director general y al Subteniente Iván Salazar encargado de recursos humanos actualmente al mando de la Academia Boliviana de Historia Militar, donde nos brindaron información sobre el estado de la academia, indicándonos cuales eran las funciones que desarrollaban, las variedades de actividades que realizaban como institución, entre otras cosas más, pero una vez terminada la breve presentación que nos brindaron, nos comentaron sus inquietudes con respecto a que actualmente la academia estaba teniendo un crecimiento de visitas, por lo que algunos visitantes entre ellos jóvenes estudiantes, historiadores les habían sugerido tener un sitio web en el cual muchas personas interesadas en la historia militar puedan tener ese libre acceso, de poder entrar a revisar documentos históricos avalados por la academia.

Por lo cual nos pusimos a ver qué tipo de sistema estaban buscando, entre la charla surgieron varias ideas, pero los encargados de la academia nos dieron una información donde nos indicaban que anteriormente ya habían trabajado con un sistema el cual realizaba publicaciones de blogs y algunas noticias que ellos compartían sobre la academia y sus filiales pero en el transcurso del uso de dicho sistema surgieron errores como la más principal que fue inseguridad en la información, debido a que un artículo mal escrito publicado llego a países vecinos en donde la redacción no les pareció muy adecuada, en ese sentido hubo llamadas de atención pero al ver que el sistema era vulnerable al fácil acceso de que terceras personas ingresaran a modificar la información se optó por dejar de usar dicho sistema, con esa información brindada empezamos a buscar soluciones en base a las palabras de nuestros encargados de la Academia y nos pusimos a ver las diferentes opciones de desarrollo. Para ver las preguntas realizadas en la entrevista ver Anexo E.

3.2.1.2. Descripción de necesidades

- Tener un sitio web en el cual puedan mostrar las actividades que se realizan dentro de la academia, la fotografías que se tienen actualmente, consultas sobre libros en la biblioteca, y entre otras reliquias antiguas.
- Tener un formulario en el cual puedan agregar a usuarios con roles y funciones específicas, además de una tabla para administrarlos,
- Tener un formulario en el cual puedan agregar colecciones de fotos presidenciales, además de una tabla para administrarlos.
- Tener un formulario en el cual puedan agregar blogs denominados “30 segundos con la historia”, además de una tabla para administrarlos.
- Tener un formulario en el cual puedan agregar publicaciones sobre las visitas estudiantiles académicas, además de una tabla para administrarlos.
- Tener un formulario en el cual puedan agregar publicaciones sobre eventos de exposición, además de una tabla para administrarlos.
- Tener un formulario en el cual puedan agregar publicaciones sobre noticias que de la centra y sus filiales, además de una tabla para administrarlos.
- Tener un formulario en el cual puedan agregar libros disponibles en la academia y sus filiales, además de una tabla administrarlos.
- Que el sistema pueda contar con la seguridad correspondiente para no sufrir los cambios de la información o alterar las publicaciones maliciosamente y asegurar la integridad de los datos.

3.2.1.3. Propuesta del programador

- Se realizará un sistema de gestión de contenidos personalizado para la Academia diseñado para administrar y difundir eficazmente su valioso contenido histórico y educativo, para llegar a más personas.
- Se realizará los siguientes módulos para el sistema de gestión de contenidos: módulo usuarios, módulo colección de presidentes, módulo blog (30 segundos con la historia), módulo visitas, módulo eventos, módulo noticias y módulo libros disponibles.
- Cada módulo se desarrollará de manera personalizada según las recomendaciones de la academia para que se tenga un buen manejo del sistema, presentando los módulos correspondientes en las fechas indicadas.
- Se implementará un algoritmo para la seguridad del sistema empleando el enmascaramiento de datos, para garantizar que la información publicada no sea alterada y sea protegida de manera segura, esto para tener la conformidad de la academia y que no corra ningún riesgo al momento de realizar sus publicaciones correspondientes.

3.2.2. Requerimientos funcionales y no funcionales

Tabla 5:

Descripción de los requerimientos funcionales

Requerimientos Funcionales	Descripción	Prioridad
Gestión de autenticación y permisos de usuarios.	Permite a los usuarios autenticarse en el sistema, gestionar usuarios y asignar permisos según sus roles.	Alta
Gestión de contenidos.	Administrar diferentes tipos de contenidos: colección de presidentes, visitas, eventos, noticias, blogs y libros.	Media
Protección de información sensible mediante enmascaramiento.	Aplicar un algoritmo de enmascaramiento para proteger datos sensibles y garantizar la seguridad de la información.	Alta

Nota: En la presente tabla se muestra todos los requerimientos funcionales del sistema que son los más importantes para el desarrollo del presente proyecto. Fuente: Elaboración Propia

Tabla 6:*Descripción de los requerimientos no funcionales*

Requerimientos No Funcionales	Descripción	Prioridad
Arquitectura escalable.	El sistema debe poder ampliarse y adaptarse a futuros cambios o aumento de usuarios sin perder rendimiento.	Alta
Experiencia de usuario optimizada.	Garantizar una navegación ágil, fluida y una interfaz intuitiva para mejorar la interacción del usuario.	Alta
Rendimiento del sistema.	Asegurar tiempos de respuesta rápidos y procesamiento eficiente en todas las operaciones.	Alta

Nota: En la presente tabla se muestra todos los requerimientos no funcionales del sistema, son los estándares que se deberá realizar y cumplir para que el proyecto funcione correctamente. Fuente: Elaboración Propia

3.2.3. Asignación de roles del proyecto

En la Tabla 7 se muestra la asignación de los roles para el presente proyecto.

Tabla 7:*Asignación de roles del proyecto.*

Roles	Miembros
Programador	Dev. José Antonio Aguilar Fernández
Cliente	Coronel Gary Mercado
Encargado de pruebas	Subteniente Iván Salazar
Encargado de seguimiento	General José Rodrigo García

Nota: En la tabla se muestra la asignación de roles que se contemplaran dentro del proyecto, mostrando a los encargados de hacer pruebas y seguimiento, junto con el cliente y el programador a cargo del desarrollo del sistema.

3.2.4. Historias de usuario

Las Historias de Usuario han sido descritas en un lenguaje común, para que puedan ser entendidas por todos los miembros del grupo de trabajo, representando los requerimientos con los que debe cumplir el sistema como se representan en las siguientes tablas descritas a continuación:

Tabla 8:*Historia de usuario para el acceso al sistema*

HISTORIA DE USUARIO	
Número: 1	Usuario: Administrador, Editor y Afiliado
Nombre Historia: Acceso al sistema.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 2	Iteración asignada: 1
Programador responsable: José Antonio Aguilar Fernández	
Descripción: Los usuarios administrador, editor y afiliado tendrán un correo y una contraseña única para poder ingresar al sistema.	
Observaciones: El usuario por defecto será el administrador, el será el encargado de crear nuevos usuarios, como el editor y el afiliado cada uno con sus credenciales correspondientes.	

Nota: En la tabla se muestra la historia de usuario para el acceso al sistema. Fuente: Elaboración Propia.

Tabla 9:*Historia de usuario creación de permisos*

HISTORIA DE USUARIO	
Número: 2	Usuario: Administrador
Nombre Historia: Creación de permisos.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 2	Iteración asignada: 1
Programador responsable: José Antonio Aguilar Fernández	
Descripción: El administrador le asignará el permiso que tendrá dentro del sistema.	
Observaciones: Solamente el usuario administrador podrá asignar permisos a los usuarios.	

Nota: En la tabla se muestra la historia de usuario para la creación de permisos. Fuente: Elaboración Propia.

Tabla 10:*Historia de usuario gestión de usuarios*

HISTORIA DE USUARIO	
Número: 3	Usuario: Administrador, Editor y Afiliado
Nombre Historia: Gestión de usuarios.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 2	Iteración asignada: 1
Programador responsable: José Antonio Aguilar Fernández	
Descripción: Por defecto el sistema mostrará al usuario administrador, pero el sistema le mostrará a cada usuario un entorno de trabajo diferente según su permiso asignado	
Observaciones: Solo el administrador tendrá el total control de la funcionalidad del sistema.	

Nota: En la tabla se muestra la historia de usuario para la gestión de usuario. Fuente: Elaboración Propia.

Tabla 11:*Historia de usuario gestión colección de presidentes*

HISTORIA DE USUARIO	
Número: 4	Usuario: Administrador, Editor
Nombre Historia: Gestión de colección de presidentes.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 2	Iteración asignada: 1
Programador responsable: José Antonio Aguilar Fernández	
Descripción: La información requerida de la colección de presidentes será cargada al sistema, almacenada en la base de datos y publicado en el sitio web.	
Observaciones: Solo los usuarios asignados podrán realizar estas funciones.	

Nota: En la tabla se muestra la historia de usuario para la gestión de colección de presidentes. Fuente: Elaboración Propia.

Tabla 12:*Historia de usuario gestión de visitas*

HISTORIA DE USUARIO	
Número: 5	Usuario: Administrador, Editor, Afiliado
Nombre Historia: Gestión de visitas.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 2	Iteración asignada: 1
Programador responsable: José Antonio Aguilar Fernández	
Descripción: La información requerida de las visitas será cargada al sistema, almacenada en la base de datos y publicado en el sitio web.	
Observaciones: Solo los usuarios asignados podrán realizar estas funciones.	

Nota: En la tabla se muestra la historia de usuario para la gestión de colección de visitas. Fuente: Elaboración Propia.

Tabla 13:*Historia de usuario gestión de eventos*

HISTORIA DE USUARIO	
Número: 6	Usuario: Administrador, Editor, Afiliado
Nombre Historia: Gestión de eventos.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 2	Iteración asignada: 1
Programador responsable: José Antonio Aguilar Fernández	
Descripción: La información requerida de los eventos será cargada al sistema, almacenada en la base de datos y publicado en el sitio web.	
Observaciones: Solo los usuarios asignados podrán realizar estas funciones.	

Nota: En la tabla se muestra la historia de usuario para la gestión de eventos. Fuente: Elaboración Propia.

Tabla 14:*Historia de usuario gestión de noticias*

HISTORIA DE USUARIO	
Número: 7	Usuario: Administrador, Editor, Afiliado
Nombre Historia: Gestión de noticias.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 2	Iteración asignada: 1
Programador responsable: José Antonio Aguilar Fernández	
Descripción: La información requerida de las noticias será cargada al sistema, almacenada en la base de datos y publicado en el sitio web.	
Observaciones: Solo los usuarios asignados podrán realizar estas funciones.	

Nota: En la tabla se muestra la historia de usuario para la gestión de noticias. Fuente: Elaboración Propia.

Tabla 15:*Historia de usuario gestión de blogs*

HISTORIA DE USUARIO	
Número: 8	Usuario: Administrador, Editor
Nombre Historia: Gestión de blogs.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 2	Iteración asignada: 1
Programador responsable: José Antonio Aguilar Fernández	
Descripción: La información requerida de los blogs será cargada al sistema, almacenada en la base de datos y publicado en el sitio web.	
Observaciones: Solo los usuarios asignados podrán realizar estas funciones.	

Nota: En la tabla se muestra la historia de usuario para la gestión de blogs. Fuente: Elaboración Propia.

Tabla 16:*Historia de usuario análisis comparativo*

HISTORIA DE USUARIO	
Número: 9	Usuario: Administrador
Nombre Historia: Análisis comparativo.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 3	Iteración asignada: 1
Programador responsable: José Antonio Aguilar Fernández	
Descripción: Se realizará un análisis comparativo de las diferentes técnicas de enmascaramiento y se seleccionará una técnica, para trabajar e implementarla.	
Observaciones: El usuario administrador estará a cargo de hacer el análisis comparativo.	

Nota: En la tabla se muestra la historia de usuario para el análisis comparativo. Fuente: Elaboración Propia.

Tabla 17:*Historia de usuario algoritmo de enmascaramiento*

HISTORIA DE USUARIO	
Número: 10	Usuario: Administrador
Nombre Historia: Algoritmo de enmascaramiento.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 3	Iteración asignada: 1
Programador responsable: José Antonio Aguilar Fernández	
Descripción: Se desarrollará el algoritmo correspondiente para realizar las pruebas correspondientes en los datos a enmascarar.	
Observaciones: El administrador estará realizando las pruebas del algoritmo.	

Nota: En la tabla se muestra la historia de usuario para el algoritmo de enmascaramiento. Fuente: Elaboración Propia.

Tabla 18:*Historia de usuario implementación de algoritmo*

HISTORIA DE USUARIO	
Número: 11	Usuario: Administrador
Nombre Historia: Implementación del algoritmo.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 3	Iteración asignada: 1
Programador responsable: José Antonio Aguilar Fernández	
Descripción: Se realizará la implementación del algoritmo desarrollado para ponerlo en producción y hacer las pruebas correspondientes.	
Observaciones: Se harán pruebas en la producción para evitar errores en el producto final.	

Nota: En la tabla se muestra la historia de usuario para la implementación del algoritmo. Fuente: Elaboración Propia.

3.2.5. Plan de entregas del proyecto

Para la elaboración del plan de entrega, se han tomado en cuenta criterios clave como la prioridad asignada a cada historia de usuario y el nivel de esfuerzo estimado para su implementación, estos factores permitieron establecer una planificación realista y coherente con los objetivos del proyecto, asegurando que las funcionalidades más críticas y de mayor impacto fueran abordadas con anticipación, la distribución estratégica de las tareas, basada en estos parámetros, se detalla en la Tabla 19, la cual proporciona una visión clara del enfoque adoptado para la gestión y desarrollo incremental del sistema.

Tabla 19:*Plan de entregas del proyecto*

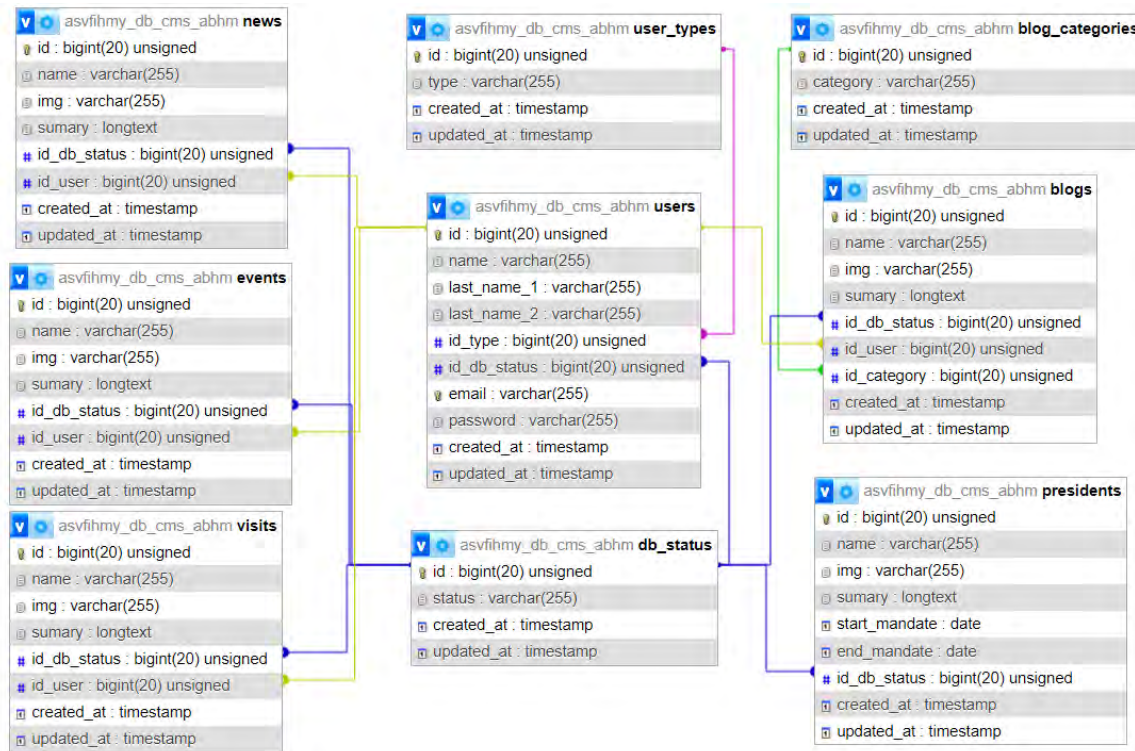
Historias	Iteración	Prioridad	Esfuerzo	Fecha Inicio	Fecha Final
Historia 1	1	Alta	2	06/10/2024	13/10/2024
Historia 2	1	Alta	2	06/10/2024	13/10/2024
Historia 3	1	Alta	2	13/10/2024	20/10/2024
Historia 4	1	Alta	2	13/10/2024	20/10/2024
Historia 5	1	Alta	2	20/10/2024	27/10/2024
Historia 6	1	Alta	2	20/10/2024	27/10/2024
Historia 7	1	Alta	2	27/10/2024	03/11/2024
Historia 8	1	Alta	2	27/10/2024	03/11/2024
Historia 9	1	Alta	2	03/11/2024	10/11/2024
Historia 10	1	Alta	3	03/11/2024	10/11/2024
Historia 11	1	Alta	3	10/11/2024	17/11/2024

Nota: En la tabla se muestra el plan de entregas utilizando cada una de las historias de usuario, mostrando la iteración, la prioridad, el esfuerzo, fechas de inicio y final. Fuente: Elaboración Propia.

3.3. DISEÑO DEL MARCO APLICATIVO

3.3.1. Diagrama de bases de datos

En la siguiente Figura 19 se muestra como está estructurada la base de datos para la academia boliviana de historia militar, en donde se aplicó todas las bases fundamentales mencionadas en el marco teórico para que el sistema de gestión de contenidos sea flexible y fácil de manejar para los encargados de la academia, además de no tener ninguna complicación con los nombres ni con los formularios, teniendo un total de diez tablas en las bases de datos.

Figura 19:**Diagrama de bases datos del sistema**

Nota: En la presente figura el modelo entidad relación en su tercera forma normal de la base de datos, muestra como estará estructurada el sistema de gestión de contenidos. Fuente: Elaboración Propia

3.3.2. Diagrama de clases del sistema

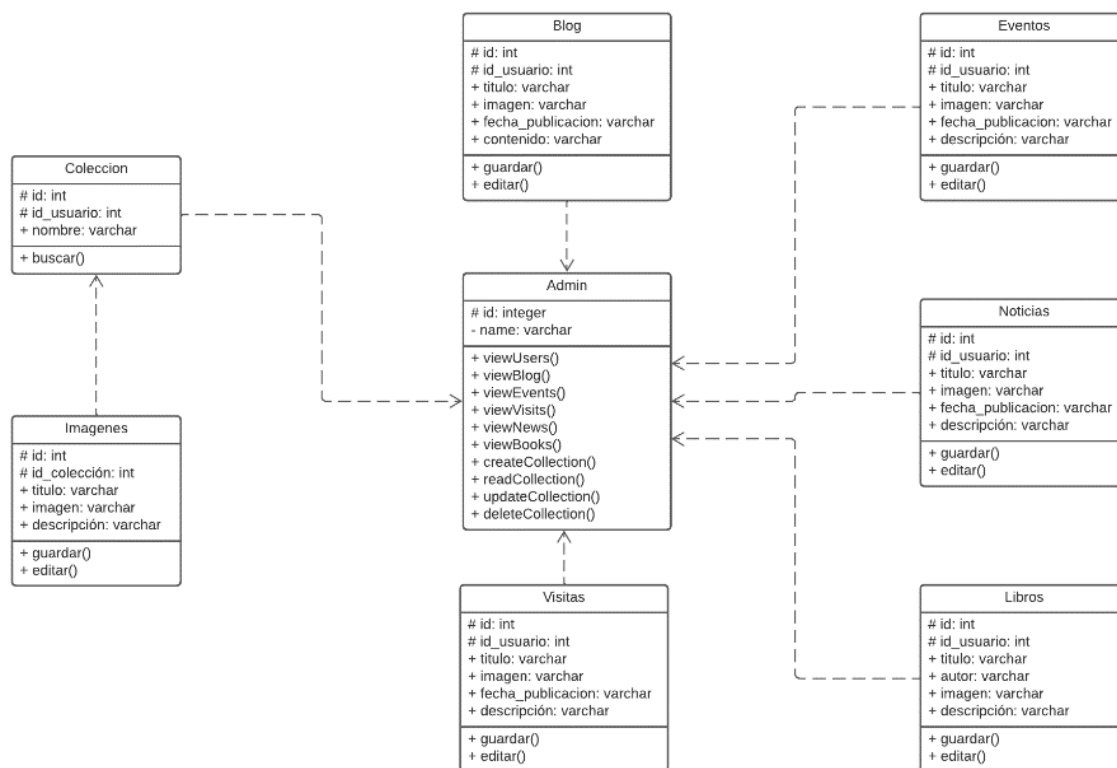
En este apartado se presentarán tres diagramas de clases que permitirán comprender con mayor claridad los distintos roles de usuario definidos dentro del sistema, así como las funciones específicas que desempeña cada uno en el contexto de la plataforma, el uso de estos diagramas facilita la representación visual de las entidades del sistema, sus atributos y los métodos que cada clase implementa, permitiendo una mejor organización y comprensión del modelo orientado a objetos que se ha utilizado en el desarrollo del proyecto.

A continuación, en la Figura 20, se presenta el diagrama de clases correspondiente al usuario administrador, quien desempeña un papel central en la gestión del sistema, este rol tiene acceso a diversas funcionalidades que le permiten visualizar, modificar y administrar contenidos como blogs, eventos, noticias, visitas, libros e imágenes. Asimismo, el puede realizar operaciones CRUD (crear, leer, actualizar y eliminar) sobre distintas entidades, y

tiene la responsabilidad de mantener el correcto funcionamiento de los recursos internos del sistema, la estructura reflejada en este diagrama también evidencia las relaciones que existen entre las clases que interactúan con los roles administrativos, proporcionando una vista general del alcance y control que este rol tiene sobre la plataforma.

Figura 20:

Diagrama de clases para el usuario administrador



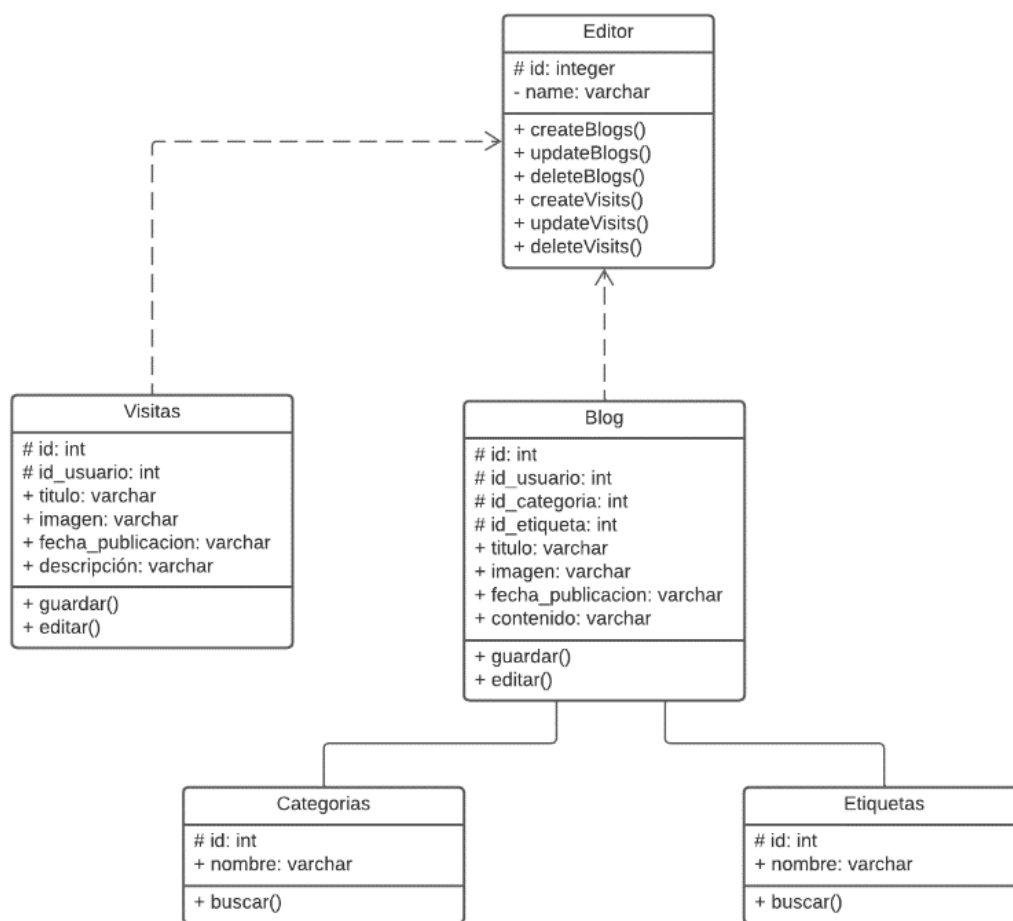
Nota: En la presente figura se muestra el diagrama de clases para el usuario administrador y las funciones que realizaría dentro del sistema. Fuente: Elaboración Propia

Continuando con la presentación de diagramas, en la Figura 21 se expone el diagrama de clases correspondiente al usuario editor, cuya función principal dentro del sistema es colaborar en la gestión y publicación de contenidos, este rol tiene permisos limitados en comparación con el administrador, pero sigue siendo fundamental para el mantenimiento actualizado de la información y la generación de contenido relevante en la plataforma. El usuario editor tiene acceso a funcionalidades específicas como la creación y edición de entradas en módulos tales como noticias, eventos, blogs o libros, dependiendo de las restricciones establecidas por el administrador del sistema.

A través del diagrama de clases se puede observar la interacción de este rol con distintas entidades del sistema, así como los métodos y atributos que le permiten realizar sus tareas operativas, este enfoque modular garantiza que el editor actúe dentro de un entorno seguro y controlado, evitando la alteración de componentes críticos del sistema, el uso de este tipo de representaciones visuales refuerza la comprensión del comportamiento de los usuarios y su impacto en la arquitectura general del sistema.

Figura 21:

Diagrama de clases para el usuario editor



Nota: En la presente figura se muestra el diagrama de clases para el usuario editor y las funciones que realizaría dentro del sistema. Fuente: Elaboración Propia

En la Figura 22, se presenta el diagrama de clases correspondiente al usuario afiliado, quien representa a un perfil con funciones y accesos específicos dentro del sistema.

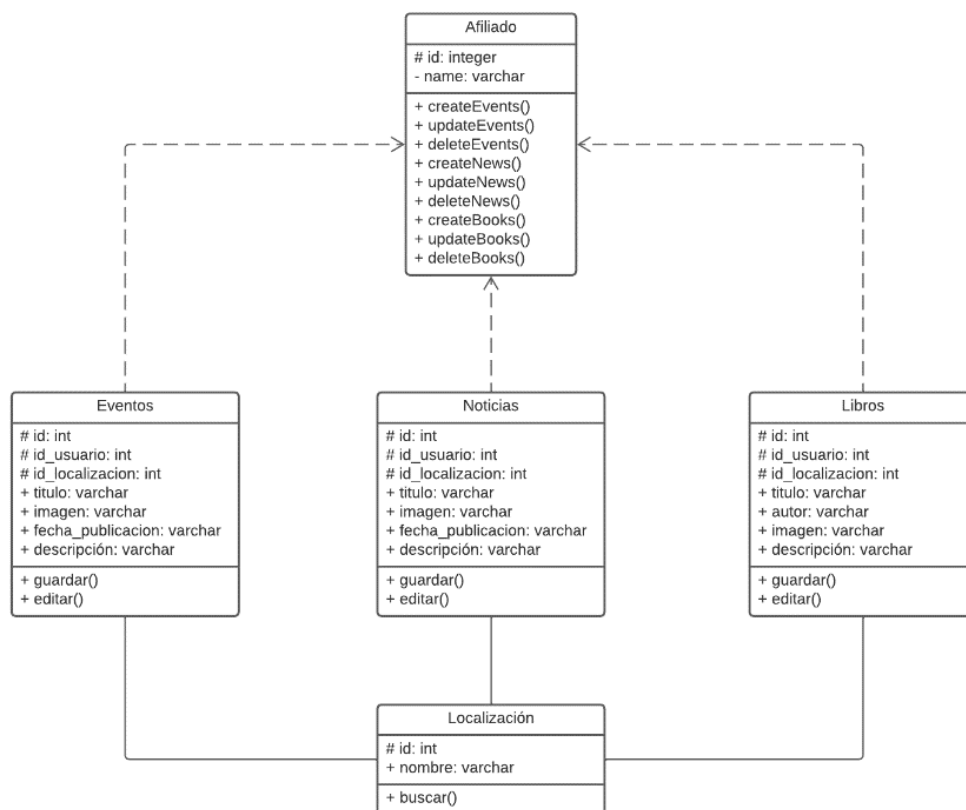
A diferencia de los roles de administrador o editor, el afiliado tiene permisos más restringidos, orientados principalmente al consumo de contenido y a la participación en actividades académicas o culturales promovidas por la plataforma.

Este tipo de usuario puede acceder a información relevante, visualizar publicaciones, eventos y material histórico, así como interactuar con ciertos módulos habilitados por el administrador, el diagrama de clases permite identificar con claridad los atributos y métodos asociados al perfil del afiliado, así como su relación con otras entidades del sistema, como las visitas o colecciones.

La inclusión de este rol en el modelo del sistema contribuye a una mayor personalización de la experiencia de usuario y a una mejor segmentación de los accesos, fortaleciendo así tanto la usabilidad como la seguridad del entorno digital.

Figura 22:

Diagrama de clases para el usuario afiliado



Nota: En la presente figura se muestra el diagrama de clases para el usuario afiliados y las funciones que realizaría dentro del sistema. Fuente: Elaboración Propia

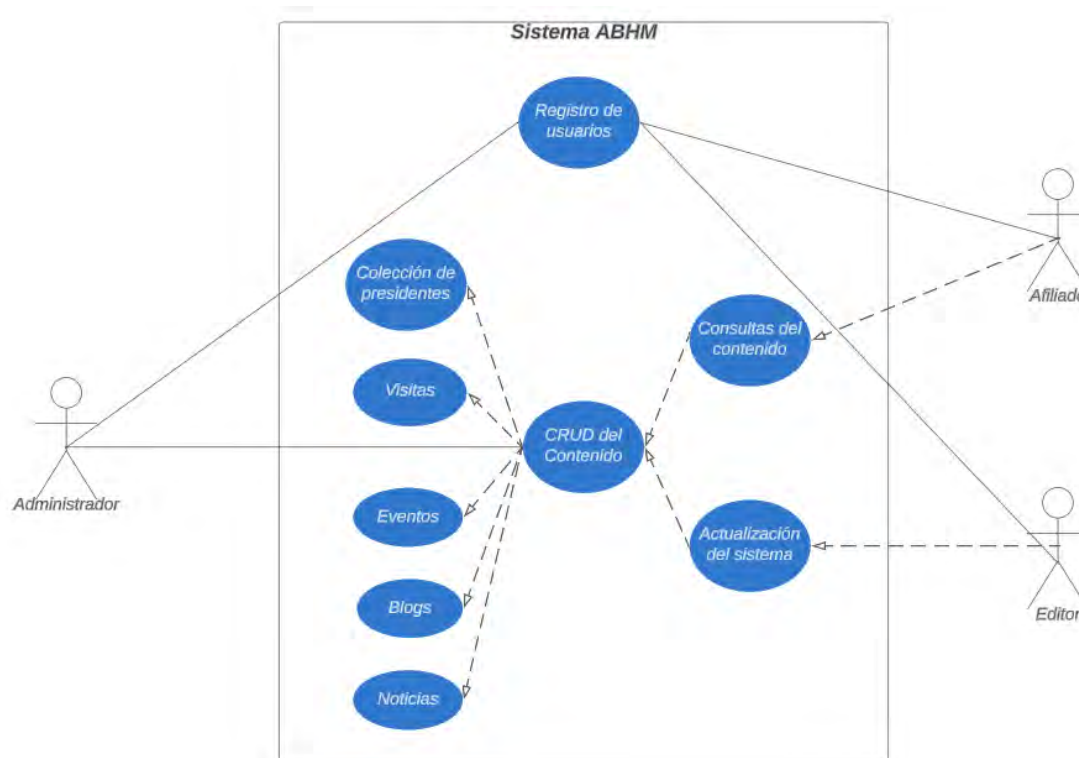
3.3.3. Diagrama de casos de uso del sistema

En este apartado se presentarán los diagramas de casos de uso, que nos ayudan comprender de forma clara y estructurada los diferentes roles de usuario dentro del sistema, así como las acciones que cada uno puede llevar a cabo, esta representación gráfica es fundamental para entender la interacción entre los usuarios y el sistema, además de facilitar la definición de requerimientos funcionales desde la perspectiva del usuario.

A continuación, en la Figura 23 se presenta el diagrama de casos de uso de nivel 0, que proporciona una visión general del sistema desarrollado, el diagrama muestra de manera simplificada la interacción principal entre los actores y el sistema, enfocándose únicamente en los procesos más generales y relevantes, el objetivo es representar de forma clara y concisa las funcionalidades esenciales, sirviendo como base para entender el alcance del proyecto y la relación entre los usuarios y las operaciones principales.

Figura 23:

Diagrama de casos de uso de nivel 0

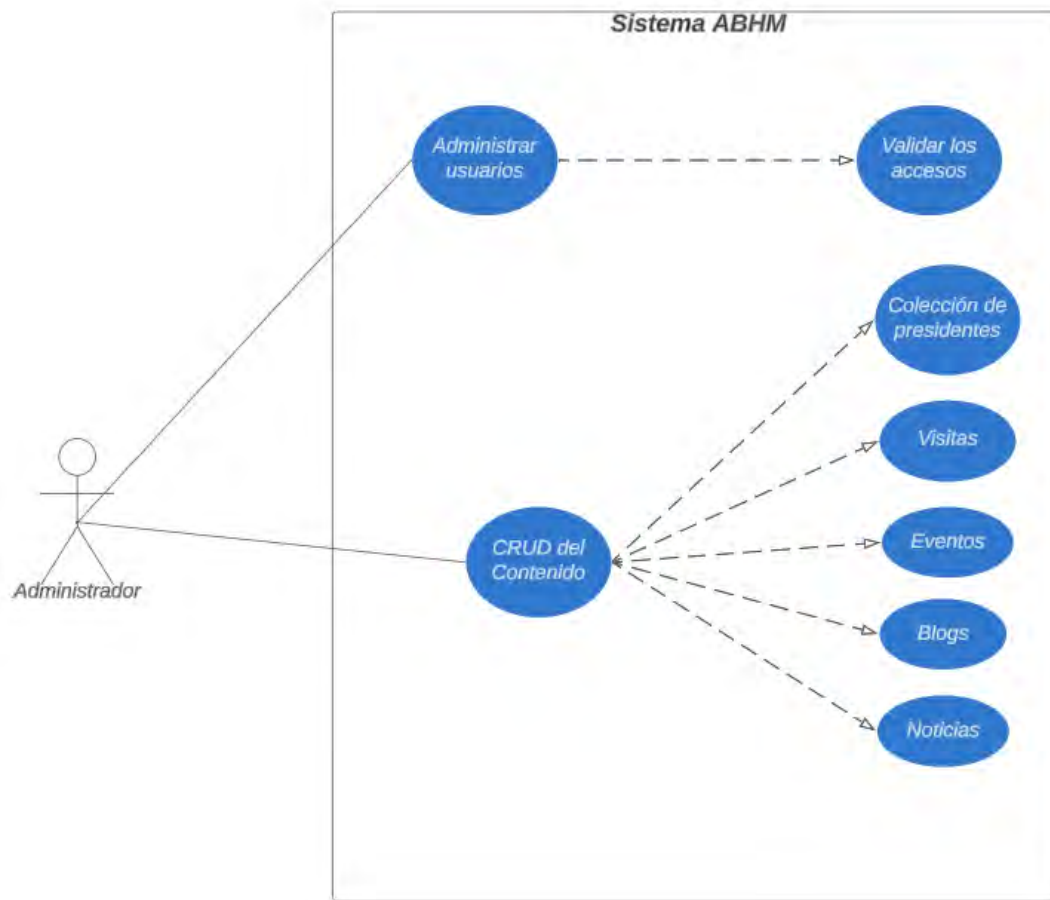


Nota: En la presente figura se muestra el diagrama de casos de uso de nivel 0 mostrando las funciones que realizara cada actor dentro del sistema. Fuente: Elaboración Propia

A continuación, en la Figura 24, se muestra el diagrama de casos de uso correspondiente al usuario administrador. En esta ilustración se detallan las principales funcionalidades que este perfil puede ejecutar dentro del sistema, tales como la gestión de contenidos, control de usuarios, edición de publicaciones y supervisión de otras actividades clave.

Figura 24:

Diagrama de casos de uso para el usuario administrador

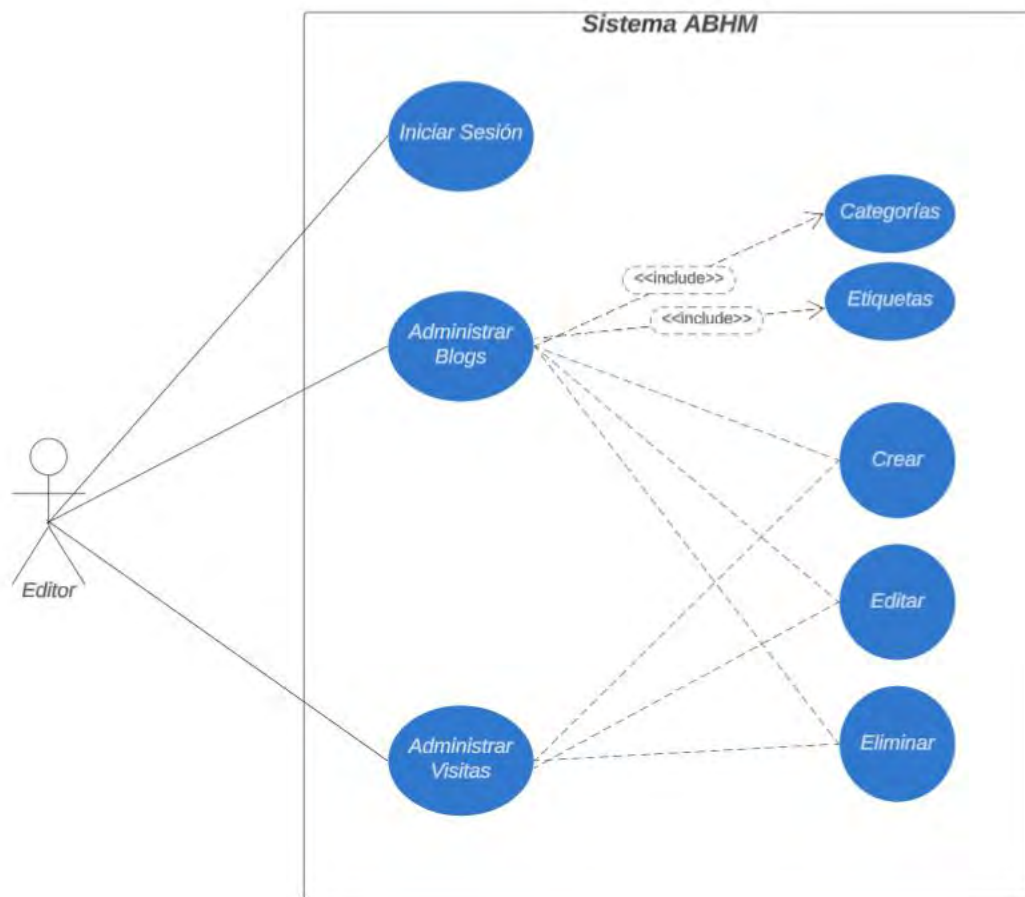


Nota: En la presente figura se muestra el diagrama de casos de uso para el usuario administrador y las funciones que realizara dentro del sistema. Fuente: Elaboración Propia

Continuando con la presentación de los diagramas, en la Figura 25 se presenta el diagrama de casos de uso correspondiente al usuario editor, en este diagrama permite visualizar de manera clara las acciones y responsabilidades que tiene este perfil dentro del sistema, el usuario editor posee permisos específicos orientados principalmente a la gestión y edición de contenidos, como la creación, modificación o eliminación de publicaciones, así como la administración de ciertos recursos documentales asignados. A través de este diagrama se busca representar gráficamente las interacciones clave de este rol en el sistema.

Figura 25:

Diagrama de casos de uso para el usuario editor

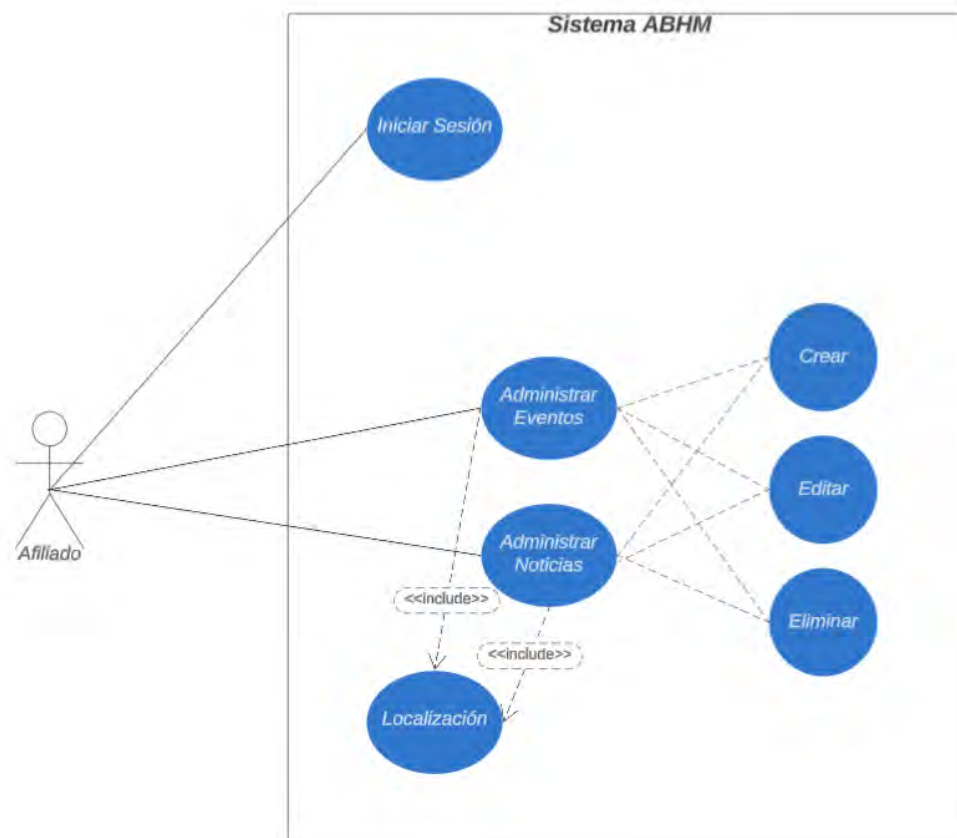


Nota: En la presente figura se muestra el diagrama de casos de uso para el usuario editor y las funciones que realizara dentro del sistema. Fuente: Elaboración Propia

Finalmente, en la Figura 26, se muestra el diagrama de casos de uso correspondiente al usuario afiliado, en este diagrama representa de manera esquemática las principales interacciones que puede realizar este tipo de usuario dentro del sistema, enfocándose en funciones de consulta, visualización y participación dentro de la plataforma, a diferencia del administrador y del editor, el usuario afiliado cuenta con permisos restringidos y orientados al acceso controlado a la información publicada, como la lectura de documentos, visualización de contenido histórico académico y posiblemente la generación de reportes personales, este tipo de representación es fundamental para comprender cómo los distintos tipos de usuarios utilizan el sistema según sus roles, permitiendo a su vez definir límites de seguridad, accesibilidad y funcionalidad.

Figura 26:

Diagrama de casos de uso para el usuario afiliado



Nota: En la presente figura se muestra el diagrama de casos de uso para el usuario afiliado y las funciones que realizara dentro del sistema. Fuente: Elaboración Propia

3.3.4. Descripción del apoyo a las actividades institucionales

La arquitectura y estructura del sistema de gestión de contenidos ha sido diseñada de manera modular, permitiendo una organización eficiente de las distintas funcionalidades requeridas por la Academia Boliviana de Historia Militar, este diseño permite apoyar directamente las actividades institucionales mediante la centralización de la información histórica, el registro de eventos, administración de colecciones, y publicaciones de contenido educativo, además de la implementación de formularios específicos para cada tipo de contenido, junto con la asignación de roles y permisos diferenciados, facilitando a que los usuarios responsables de área puedan gestionar sus secciones de manera autónoma y segura, optimizando los procesos internos y mejorando la difusión pública de sus responsabilidades institucionales.

A continuación, se presenta la Tabla 20, en la cual se detallan las principales actividades académicas que se desarrollan en la Academia Boliviana de Historia Militar, junto con la descripción del apoyo proporcionado por el diseño de la arquitectura del sistema de gestión de contenidos.

Tabla 20:

Apoyo a las actividades institucionales

ACTIVIDADES INSTITUCIONALES	
Actividades académicas	Apoyo del sistema
Publicación de contenido histórico como blogs, artículos, noticias.	Módulo de blogs, noticias y “30 segundos con la historia” con editor integrado.
Registro de visitas estudiantiles y académicas.	Módulo de visitas con formulario estructurado y tabla de administración por fechas.
Exposición de colecciones y documentos presidenciales.	Módulo de colecciones con galería multimedia organizada cronológicamente.
Administración interna de usuarios y roles.	Módulo de usuarios con jerarquías (administrador, editor, afiliado) y control de accesos.
Seguridad en la integridad de la información publicada.	Un algoritmo de enmascaramiento para proteger datos sensibles y evitar modificaciones no autorizadas.
Difusión de eventos institucionales y culturales.	Módulo de eventos con formulario de carga rápida y galería integrada.

Nota: En la tabla se muestran las actividades académicas y como el sistema apoya mediante el diseño y la estructura. Fuente: Elaboración Propia.

3.4. CODIFICACIÓN DEL MARCO APLICATIVO

3.4.1. Primera Iteración

Para la primera iteración del desarrollo del sistema, se han implementado diversos módulos fundamentales, los cuales incluyen: el módulo de acceso al sistema, creación de permisos y gestión de usuario.

En la Tabla 21 se presenta un resumen general de las historias de usuario correspondientes a esta etapa, las cuales constituyeron la base para la planificación, diseño y desarrollo incremental de los módulos implementados, permitiendo una evolución ordenada y coherente del sistema.

Tabla 21:
Historias de usuario primera iteración

Numero de historia	Nombre
1	Acceso al sistema.
2	Creación de permisos.
3	Gestión de usuario.

Nota: En la tabla se muestran las historias de usuario que se trabajaran dentro de la primera iteración del sistema. Fuente: Elaboración Propia.

3.4.1.1. Tareas de Ingeniería primera iteración

En la Tabla 22 se presenta una visión consolidada que permite identificar, de forma rápida, el alcance del trabajo técnico requerido, para un análisis más detallado y específico de cada una de estas tareas se puede observar las Tablas 23 – 28.

Tabla 22:
Tareas de ingeniería primera iteración

Número de tarea	Número de historias	Nombre de la tarea
1	1	Diseño de la interfaz para el acceso al sistema.
2	1	Validación de las credenciales de los usuarios.
3	2	Diseño de interfaz para la creación de permisos.
4	2	Registrar y guardar los usuarios con permisos.
5	3	Diseño de interfaz para la gestión de usuarios.
6	3	Realizar pruebas de CRUD con la gestión de usuarios.

Nota: En la tabla se muestran las tareas de ingeniería de la primera iteración, además de una breve descripción sobre cada una de las historias de usuario. Fuente: Elaboración Propia.

3.4.1.2. Descripción de tareas de ingeniería

En esta sección se detallan las tareas de ingeniería realizadas durante la primera iteración del proyecto, estas actividades abarcan desde el análisis de requisitos hasta el diseño, implementación y validación inicial de los módulos del sistema, cada tarea fue planificada y ejecutada siguiendo los principios de la metodología ágil adoptada, asegurando el cumplimiento de los objetivos establecidos para esta fase y sentando las bases para las siguientes etapas de desarrollo.

Tabla 23:*Tarea de ingeniería 1, historia de usuario 1*

TAREA DE INGENIERÍA	
Número de tarea: 1	Número de historia: 1
Nombre de tarea: Diseño de la interfaz para el acceso al sistema.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 06/10/2024	Fecha fin: 13/10/2024
Programador responsable: José Antonio Aguilar Fernández	
Descripción: Se realizará un formulario de acceso para que los usuarios puedan acceder con las credenciales correspondientes.	

Nota: En la tabla se muestra la tarea de ingeniería 1, para la historia de usuario 1. Fuente: Elaboración Propia.

Tabla 24:*Tarea de ingeniería 2, historia de usuario 1*

TAREA DE INGENIERÍA	
Número de tarea: 2	Número de historia: 1
Nombre de tarea: Validación de las credenciales de los usuarios.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 06/10/2024	Fecha fin: 13/10/2024
Programador responsable: José Antonio Aguilar Fernández	
Descripción: Solo los usuarios que tengan correo y contraseña podrán acceder al sistema, en caso de no contar con las credenciales no accederá al sistema.	

Nota: En la tabla se muestra la tarea de ingeniería 2, para la historia de usuario 1. Fuente: Elaboración Propia.

Tabla 25:*Tarea de ingeniería 3, historia de usuario 2*

TAREA DE INGENIERÍA	
Número de tarea: 3	Número de historia: 2
Nombre de tarea: Diseño de interfaz para la creación de permisos.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 06/10/2024	Fecha fin: 13/10/2024
Programador responsable: José Antonio Aguilar Fernández	
Descripción: Se realizará un formulario para la creación de permisos para que los usuarios puedan acceder con las credenciales correspondientes.	

Nota: En la tabla se muestra la tarea de ingeniería número 3, para la historia de usuarios número 2. Fuente: Elaboración Propia.

Tabla 26:*Tarea de ingeniería 4, historia de usuario 2*

TAREA DE INGENIERÍA	
Número de tarea: 4	Número de historia: 2
Nombre de tarea: Registrar y guardar los usuarios con permisos.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 06/10/2024	Fecha fin: 13/10/2024
Programador responsable: José Antonio Aguilar Fernández	
Descripción: Los usuarios registrados con los permisos asignados serán almacenados en la base de datos para estar registrados.	

Nota: En la tabla se muestra la tarea de ingeniería 4, para la historia de usuarios 2. Fuente: Elaboración Propia.

Tabla 27:*Tarea de ingeniería 5, historia de usuario 3*

TAREA DE INGENIERÍA	
Número de tarea: 5	Número de historia: 3
Nombre de tarea: Diseño de interfaz para la gestión de usuarios.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 13/10/2024	Fecha fin: 20/10/2024
Programador responsable: José Antonio Aguilar Fernández	
Descripción: Se realizará un interfaz para visualizar a todos los usuarios ingresados al sistema y con posibilidad de ver sus permisos otorgados.	

Nota: En la tabla se muestra la tarea de ingeniería 5, para la historia de usuario 3. Fuente: Elaboración Propia.

Tabla 28:*Tarea de ingeniería 6, historia de usuario 3*

TAREA DE INGENIERÍA	
Número de tarea: 6	Número de historia: 3
Nombre de tarea: Realizar pruebas de CRUD con la gestión de usuarios.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 13/10/2024	Fecha fin: 20/10/2024
Programador responsable: José Antonio Aguilar Fernández	
Descripción: Se podrán realizar pruebas en el cual se podrán realizar las siguientes acciones como: Ver, Editar, Actualizar y Eliminar datos de usuarios nuevos.	

Nota: En la tabla se muestra la tarea de ingeniería 6, para la historia de usuario 3. Fuente: Elaboración Propia.

3.4.1.3. Pruebas de aceptación

En la Tabla 29, se presenta una definición general de las pruebas de aceptación, proporcionando una visión estructurada de los alcances del sistema, para un mayor nivel de detalle se pueden ver las Tablas 30 – 32 donde describen de manera específica cada una de estas pruebas, incluyendo sus objetivos, procedimientos, condiciones de evaluación y resultados esperados.

Tabla 29:

Pruebas de aceptación primera iteración

Número de Prueba	Número de Historia	Nombre de la prueba
1	1	Acceso al sistema.
2	2	Creación de permisos.
3	3	Gestión de usuario.

Nota: En la tabla se muestran las pruebas de aceptación en general, mostrando el número de prueba y de la historia de usuario. Fuente: Elaboración Propia.

3.4.1.4. Descripción de pruebas de aceptación

A continuación, en la Tabla 30 se presentan las pruebas de aceptación definidas para el proyecto.

Tabla 30:

Caso de prueba, acceso al sistema

CASO DE PRUEBA	
Código: 1	Número de historia de usuario: 1
Historia de usuario: Acceso al sistema.	
Condiciones de ejecución: El usuario administrador ingresa primero por defecto se creó su correo y contraseña para verificar el funcionamiento.	
Entrada y pasos de ejecución: - Pulsar el botón de iniciar sesión que se encuentra en el menú de navegación. - Llenar el formulario introduciendo su nombre de usuario y contraseña. - Pulsar el botón iniciar sesión o login.	
Resultado esperado: El usuario administrador ingresa al inicio del sistema, visualizando todas las funciones que le corresponden.	
Evaluación de prueba: Las pruebas concluyeron satisfactoriamente.	

Nota: En la tabla se muestra el caso de prueba para el acceso al sistema. Fuente: Elaboración Propia.

Tabla 31:
Caso de prueba, creación de permisos

CASO DE PRUEBA	
Código: 2	Número de historia de usuario: 2
Historia de usuario: Creación de permisos.	
Condiciones de ejecución: El administrador tendrá que iniciar sesión en el sistema y posteriormente podrá crear un nuevo usuario y otorgar su permiso.	
Entrada y pasos de ejecución: - Pulsar el icono de usuario en el menú de dashboard, se abrirá una vista nueva. - Pulsar el botón de nuevo usuario. - Se abrirá un modal donde se encuentra un formulario de registro. - Al terminar de llenar el formulario, se pulsará el botón de guardar. - Si desea cancelar el registro, solo debe pulsar el botón de cancelar. - Al crear un usuario se cerrará el modal y podrá ver en una tabla al nuevo usuario.	
Resultado esperado: El usuario ha sido creado exitosamente, se verifica los permisos y se comprueba que puede acceder al sistema con sus nuevas credenciales de acceso.	
Evaluación de prueba: Las pruebas concluyeron satisfactoriamente.	

Nota: En la tabla se muestra el caso de prueba para la creación de permisos. Fuente: Elaboración Propia.

Tabla 32:
Caso de prueba, gestión de usuario

CASO DE PRUEBA	
Código: 3	Número de historia de usuario: 3
Historia de usuario: Gestión de usuario.	
Condiciones de ejecución: El administrador verificar que se han creado usuarios con permisos.	
Entrada y pasos de ejecución: - En la tabla se visualizará en una fila los el nombre del usuario con 2 botones. - Botón de eliminar y editar. - El botón eliminar le abrirá una ventana modal indicándole si quieres eliminar al usuario, si pone aceptar se eliminará de la tabla, pero no de la base de datos y si cancela se cierra la ventana modal. - El botón de editar le abrirá una ventana con los datos del usuario, donde se podrá editar la información y en esa ventana le aparecerá también un botón de actualizar. - El botón de actualizar le permitirá actualizar la información que haya editado y se visualizará los cambios en la tabla.	
Resultado esperado: Realizar un CRUD efectivamente bien.	
Evaluación de prueba: Las pruebas concluyeron satisfactoriamente.	

Nota: En la tabla se muestra el caso de prueba para asignar el tipo de usuario, donde se explican los pasos y resultados de la prueba. Fuente: Elaboración Propia.

A continuación, en las Figuras 27 – 28 se presentarán los mockups base diseñados para el sistema, estos prototipos permiten validar la usabilidad, identificar mejoras en la navegación y garantizar que el diseño cumpla con los requerimientos establecidos del sistema.

Figura 27:

Mockup de la pantalla de acceso al sistema

Nota: En la figura se muestra el mockup de la pantalla para el acceso al sistema. Fuente: Elaboración Propia.

Figura 28:

Mockup del formulario y dashboard de usuarios

ID	Nombres	Tipo usuario	Email	Acciones
1	José	Admin	jose@gmail.com	
2	Pedro	Editor	pedro@gmail.com	
3	Juan	Afiliado	juan@gmail.com	

Nota: En la figura se muestra el mockup del formulario y dashboard de usuarios. Fuente: Elaboración Propia.

A continuación, en las Figuras 29 – 32 se muestran una serie de capturas de pantalla que documentan visualmente el desarrollo del sistema, en base los mockups diseñados anteriormente así como de los distintos módulos que lo componen, estas imágenes corresponden a evidencias gráficas recolectadas durante el proceso de diseño y validación

funcional, en el cual se contó con la participación activa del cliente, cada una de las capturas refleja el resultado de un trabajo colaborativo, donde se definieron, ajustaron y validaron características clave de la interfaz y funcionalidades específicas del sistema.

3.4.1.5. Capturas de pantalla primera iteración

Figura 29:

Pantalla de acceso al sistema



Nota: En la figura se muestra el diseño de la pantalla para el acceso al sistema. Fuente: Elaboración Propia.

Figura 30:

Llenando el formulario con las credenciales



Nota: En la figura se muestran las credenciales de usuarios para el acceso al sistema. Fuente: Elaboración Propia.

Figura 31:*Pantalla de formulario y permisos de usuario*

Nuevo usuario

Nombre
Ingrese el nombre

Apellido paterno
Ingrese el apellido paterno

Apellido materno
Ingrese el apellido materno

Acceso
▼

Email
Ingrese el email

Contraseña
Ingrese una contraseña

Cancelar Guardar

Nota: En la figura se muestra el diseño de formulario para la creación de usuarios y el tipo de acceso que tendrá dentro del sistema. Fuente: Elaboración Propia.

Figura 32:*Pantalla de gestión de usuarios*

Usuarios

Nuevo usuario

Show 10 entries Search:

Id	Nombre	Ap. Paterno	Ap. Materno	Acceso	Email	Options
1	José	Aguilar	Fernandez	Admin	jaguiar@gmail.com	
2	ivan alfredo	salazar	aduviri	Admin	ivansalazar@gmail.com	

Showing 1 to 2 of 2 entries Previous 1 Next

Nota: En la figura se muestra el diseño de pantalla para los usuarios en donde se puede ver la tabla. Fuente: Elaboración Propia.

3.4.1.6. Descripción de resultados primera iteración

Como resultados de entrega de la primera iteración del sistema, el cliente manifestó su satisfacción general con las funcionalidades implementadas en los módulos desarrollados, las características presentadas cumplieron con los requerimientos funcionales iniciales y permitieron validar, en un entorno real, el enfoque adoptado para el diseño y desarrollo del sistema.

3.4.2. Segunda Iteración

Para la segunda iteración del desarrollo del sistema, se han implementado diversos módulos fundamentales, los cuales incluyen: el módulo de gestión de colección de presidentes, gestión de visitas y gestión de eventos.

En la Tabla 33 se presenta un resumen general de las historias de usuario correspondientes a esta etapa, las cuales constituyeron la base para la planificación, diseño y desarrollo incremental de los módulos implementados, permitiendo una evolución ordenada y coherente del sistema.

Tabla 33:
Historias de usuario segunda iteración

Numero de historia	Nombre
4	Gestión de colección de presidentes.
5	Gestión de Visitas.
6	Gestión de Eventos.

Nota: En la tabla se muestran las historias de usuario que se trabajaran dentro de la segunda iteración del sistema. Fuente: Elaboración Propia.

3.4.2.1. Tareas de ingeniería segunda iteración

En la Tabla 34 se presenta una visión consolidada que permite identificar, de forma rápida, el alcance del trabajo técnico requerido, para un análisis más detallado y específico de cada una de estas tareas se puede observar las Tablas 35 – 40.

Tabla 34:*Tareas de ingeniería segunda iteración*

Número de tarea	Número de historias	Nombre de la tarea
7	4	Diseño de la interfaz para la gestión colección de presidentes.
8	4	Realizar pruebas de CRUD en la colección de presidentes.
9	5	Diseño de la interfaz para la gestión visitas.
10	5	Realizar pruebas de CRUD en las visitas.
11	6	Diseño de la interfaz para la gestión de eventos.
12	6	Realizar pruebas de CRUD en los eventos.

Nota: En la tabla se muestran las tareas de ingeniería de la segunda iteración. Fuente: Elaboración Propia

3.4.2.2. Descripción de las tareas de ingeniería

En esta sección se detallan las tareas de ingeniería realizadas durante la segunda iteración del proyecto, desde el análisis de requisitos hasta el diseño, implementación y validación.

Tabla 35:*Tarea de ingeniería 7, historia de usuario 4*

TAREA DE INGENIERÍA	
Número de tarea: 7	Número de historia: 4
Nombre de tarea: Diseño de la interfaz para la gestión de colección de presidentes.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 13/10/2024	Fecha fin: 20/10/2024
Programador responsable: José Antonio Aguilar Fernández	
Descripción: Se realizará el diseño de interfaz y un formulario de registro, para cargar la información de la colección de presidentes.	

Nota: En la tabla se muestra la tarea de ingeniería 7, para la historia de usuario 4. Fuente: Elaboración Propia.

Tabla 36:*Tarea de ingeniería 8, historia de usuario 4*

TAREA DE INGENIERÍA	
Número de tarea: 8	Número de historia: 5
Nombre de tarea: Realizar pruebas de CRUD en la colección de presidentes.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 13/10/2024	Fecha fin: 20/10/2024
Programador responsable: José Antonio Aguilar Fernández	
Descripción: Se podrán realizar pruebas en el cual se podrán realizar las siguientes acciones como: Ver, Editar, Actualizar y Eliminar datos de usuarios nuevos.	

Nota: En la tabla se muestra la tarea de ingeniería 8, para la historia de usuario 4. Fuente: Elaboración Propia.

Tabla 37:*Tarea de ingeniería 9, historia de usuario 5*

TAREA DE INGENIERÍA	
Número de tarea: 9	Número de historia: 5
Nombre de tarea: Diseño de la interfaz para la gestión de visitas.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 20/10/2024	Fecha fin: 27/10/2024
Programador responsable: José Antonio Aguilar Fernández	
Descripción: Se realizará el diseño de interfaz y un formulario de registro, para cargar la información de las visitas.	

Nota: En la tabla se muestra la tarea de ingeniería 9, para la historia de usuario 5. Fuente: Elaboración Propia.

Tabla 38:*Tarea de ingeniería 10, historia de usuario 5*

TAREA DE INGENIERÍA	
Número de tarea: 10	Número de historia: 5
Nombre de tarea: Realizar pruebas de CRUD en las visitas.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 20/10/2024	Fecha fin: 27/10/2024
Programador responsable: José Antonio Aguilar Fernández	
Descripción: Se podrán realizar pruebas en el cual se podrán realizar las siguientes acciones como: Ver, Editar, Actualizar y Eliminar datos de usuarios nuevos.	

Nota: En la tabla se muestra la tarea de ingeniería 10, para la historia de usuario 5. Fuente: Elaboración Propia.

Tabla 39:*Tarea de ingeniería 11, historia de usuario 6*

TAREA DE INGENIERÍA	
Número de tarea: 11	Número de historia: 6
Nombre de tarea: Diseño de la interfaz para la gestión de visitas.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 20/10/2024	Fecha fin: 27/10/2024
Programador responsable: José Antonio Aguilar Fernández	
Descripción: Se realizará el diseño de interfaz y un formulario de registro, para cargar la información de los eventos	

Nota: En la tabla se muestra la tarea de ingeniería 11, para la historia de usuario 6. Fuente: Elaboración Propia.

Tabla 40:*Tarea de ingeniería 12, historia de usuario 6*

TAREA DE INGENIERÍA	
Número de tarea: 12	Número de historia: 6
Nombre de tarea: Realizar pruebas de CRUD en los eventos.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 20/10/2024	Fecha fin: 27/10/2024
Programador responsable: José Antonio Aguilar Fernández	
Descripción: Se podrán realizar pruebas en el cual se podrán realizar las siguientes acciones como: Ver, Editar, Actualizar y Eliminar datos de usuarios nuevos.	

Nota: En la tabla se muestra la tarea de ingeniería 12, para la historia de usuario 6. Fuente: Elaboración Propia.

3.4.2.3. Pruebas de aceptación

En la Tabla 41, se presenta una definición general de las pruebas de aceptación, proporcionando una visión estructurada de los alcances del sistema, para un mayor nivel de detalle se pueden ver las Tablas 42 – 44 donde describen de manera específica cada una de estas pruebas, incluyendo sus objetivos, procedimientos, condiciones de evaluación y resultados esperados.

Tabla 41:*Pruebas de aceptación segunda iteración*

Número de Prueba	Número de Historia	Nombre de la prueba
4	4	Gestión de colección de presidentes.
5	5	Gestión de visitas.
6	6	Gestión de eventos.

Nota: En la tabla se muestran las pruebas de aceptación en general, mostrando el número de prueba y de la historia de usuario. Fuente: Elaboración Propia.

3.4.2.4. Descripción de pruebas de aceptación

A continuación, se presentan las tablas correspondientes a las pruebas de aceptación definidas para el proyecto, las cuales han sido diseñadas con el objetivo de validar las funcionalidades del sistema cumpliendo con los requerimientos de los módulos establecidos, estas pruebas nos permiten verificar el comportamiento del sistema en condiciones controladas y asegurar los criterios funcionales como los no funcionales como usabilidad.

Tabla 42:*Caso de prueba gestión de colección de presidentes*

CASO DE PRUEBA	
Código: 4	Número de historia de usuario: 4
Historia de usuario: Gestión de colección de presidentes.	
Condiciones de ejecución: Los usuarios asignados tendrán acceso al módulo de la colección de presidentes para realizar las acciones necesarias.	
Entrada y pasos de ejecución: - Pulsar el botón Nuevo presidente y se le abrirá una venta modal donde podrá llenar el formulario para guardarlo. - En la ventana modal se ingresan los datos solicitados por el formulario de la colección de presidentes. - Pulsar el botón Guardar y se agregara a la tabla asignada de la colección de presidentes. - Si pulsa el botón Cancelar se cerrará la ventana modal y no se guardará nada.	
Resultado esperado: El formulario tiene una interfaz intuitiva y fácil de manejar, para que los puedan ingresar todos los datos de las publicaciones correctamente y sin complicaciones.	
Evaluación de prueba: Las pruebas concluyeron satisfactoriamente.	

Nota: En la tabla se muestra el caso de prueba para la gestión de colección de presidentes. Fuente: Elaboración Propia.

Tabla 43:*Caso de prueba gestión de visitas*

CASO DE PRUEBA	
Código: 5	Número de historia de usuario: 5
Historia de usuario: Gestión de visitas.	
Condiciones de ejecución: Los usuarios asignados tendrán acceso al módulo de las visitas para realizar las acciones necesarias.	
Entrada y pasos de ejecución: - Pulsar el botón Nueva Visita y se le abrirá una venta modal donde podrá llenar el formulario para guardarlo. - En la ventana modal se ingresan los datos solicitados por el formulario de las visitas. - Pulsar el botón Guardar y se agregara a la tabla asignada de la colección de presidentes. - Si pulsa el botón Cancelar se cerrará la ventana modal y no se guardará nada.	
Resultado esperado: El formulario tiene una interfaz intuitiva y fácil de manejar, para que los puedan ingresar todos los datos de las publicaciones correctamente y sin complicaciones.	
Evaluación de prueba: Las pruebas concluyeron satisfactoriamente.	

Nota: En la tabla se muestra el caso de prueba para la gestión de las visitas. Fuente: Elaboración Propia.

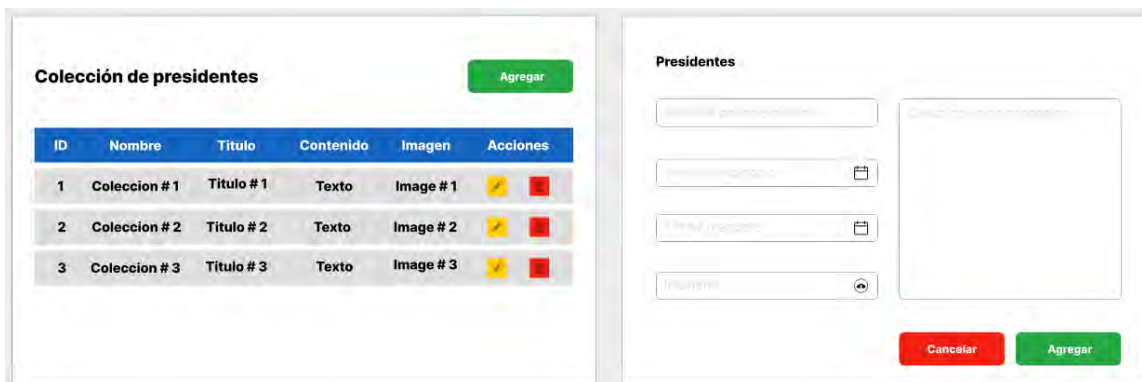
Tabla 44:
Caso de prueba gestión de eventos

CASO DE PRUEBA	
Código: 6	Número de historia de usuario: 6
Historia de usuario: Gestión de eventos.	
Condiciones de ejecución: Los usuarios asignados tendrán acceso al módulo de los eventos para realizar las acciones necesarias.	
Entrada y pasos de ejecución: - Pulsar el botón Nuevo Evento y se le abrirá una venta modal donde podrá llenar el formulario para guardarlo. - En la ventana modal se ingresan los datos solicitados por el formulario de los eventos. - Pulsar el botón Guardar y se agregara a la tabla asignada de la colección de presidentes. - Si pulsa el botón Cancelar se cerrará la ventana modal y no se guardará nada.	
Resultado esperado: El formulario tiene una interfaz intuitiva y fácil de manejar, para que los puedan ingresar todos los datos de las publicaciones correctamente y sin complicaciones.	
Evaluación de prueba: Las pruebas concluyeron satisfactoriamente.	

Nota: En la tabla se muestra el caso de prueba para la gestión de los eventos. Fuente: Elaboración Propia.

A continuación, en las Figuras 33 – 35 se presentarán los mockups base diseñados para el sistema, estos prototipos permiten validar la usabilidad, identificar mejoras en la navegación y garantizar que el diseño cumpla con los requerimientos establecidos del sistema.

Figura 33:
Mockup del formulario y dashboard de colección de presidentes



Nota: En la figura se muestra el mockup del formulario y dashboard de la colección de presidentes. Fuente: Elaboración Propia.

Figura 34:*Mockup del formulario y dashboard de visitas*

Visitas Recientes Agregar

ID	Título	Fecha	Contenido	Imagen	Acciones
1	Visita # 1	15-10-2023	Texto	Image # 1	
2	Visita # 2	19-10-2023	Texto	Image # 2	
3	Visita # 3	20-11-2023	Texto	Image # 3	

Visita

Formulario de visita con campos para: Titulo, Fecha, Contenido, Imagen, and Acciones. Botones: Cancelar, Agregar.

Nota: En la figura se muestra el mockup del formulario y dashboard de visitas. Fuente: Elaboración Propia.

Figura 35:*Mockup del formulario y dashboard de eventos*

Eventos Proximos Agregar

ID	Título	Fecha	Contenido	Imagen	Acciones
1	Evento # 1	15-10-2023	Texto	Image # 1	
2	Evento # 2	19-10-2023	Texto	Image # 2	
3	Evento # 3	20-11-2023	Texto	Image # 3	

Evento

Formulario de evento con campos para: Titulo, Fecha, Contenido, Imagen, and Acciones. Botones: Cancelar, Agregar.

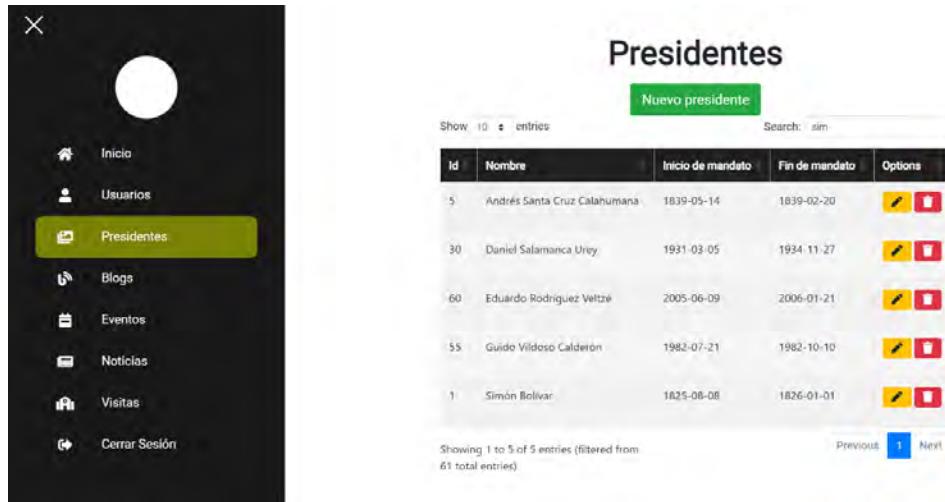
Nota: En la figura se muestra el mockup del formulario y dashboard de eventos. Fuente: Elaboración Propia.

A continuación, en las Figuras 36 – 41 se muestran una serie de capturas de pantalla que documentan visualmente el desarrollo del sistema, en base los mockups diseñados anteriormente así como de los distintos módulos que lo componen, estas imágenes corresponden a evidencias gráficas recolectadas durante el proceso de diseño y validación funcional, en el cual se contó con la participación activa del cliente, cada una de las capturas refleja el resultado de un trabajo colaborativo, donde se definieron, ajustaron y validaron características clave de la interfaz y funcionalidades específicas.

3.4.2.5. Capturas de pantalla segunda iteración

Figura 36:

Pantalla gestión de colección de presidentes



Nota: En la figura se muestra el diseño de la pantalla para la colección de presidentes. Fuente: Elaboración Propia.

Figura 37:

Pantalla de formulario de colección de presidentes

Nuevo presidente

Nombre:

Inicio de mandato:

Fin de mandato:

Imagen:

Resumen

Contenido

Lugar y fecha de nacimiento:

Títulos y honores:

Exaltación al mando:

Edad al asumir el mando:

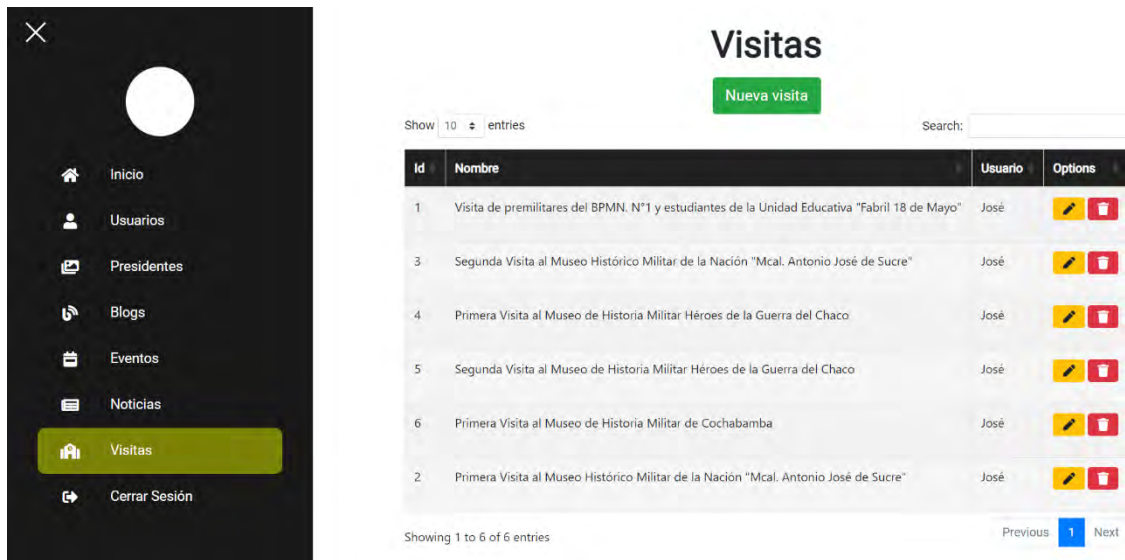
Conclusión de su mandato:

Tiempo de su mandato:

ACTOS ADMINISTRATIVOS

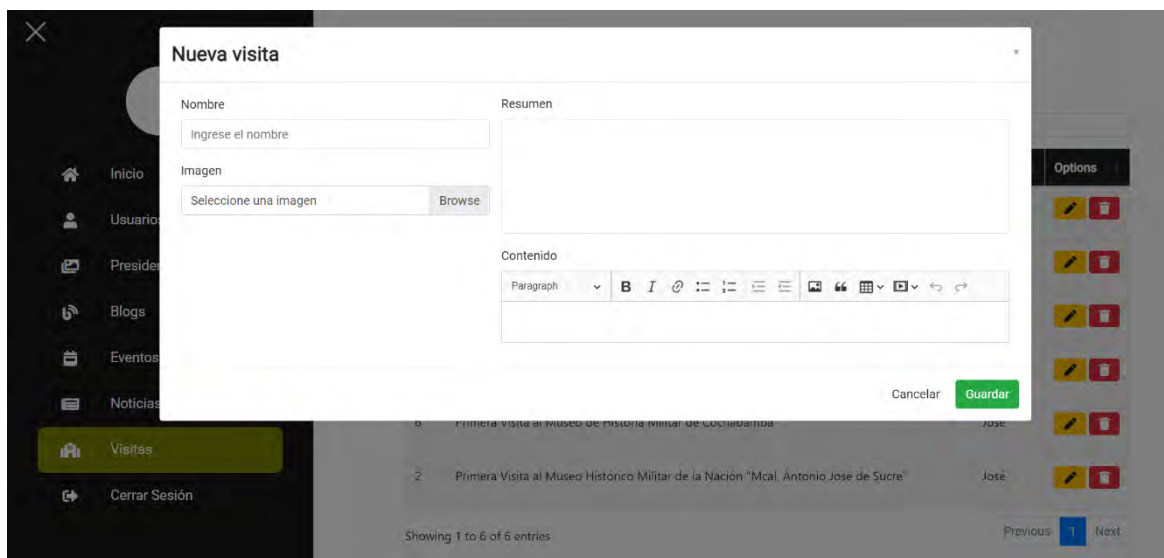
Nota: En la figura se muestra el diseño del formulario de la pantalla para crear una colección de presidentes. Fuente: Elaboración Propia.

Figura 38:
Pantalla para gestión de visitas

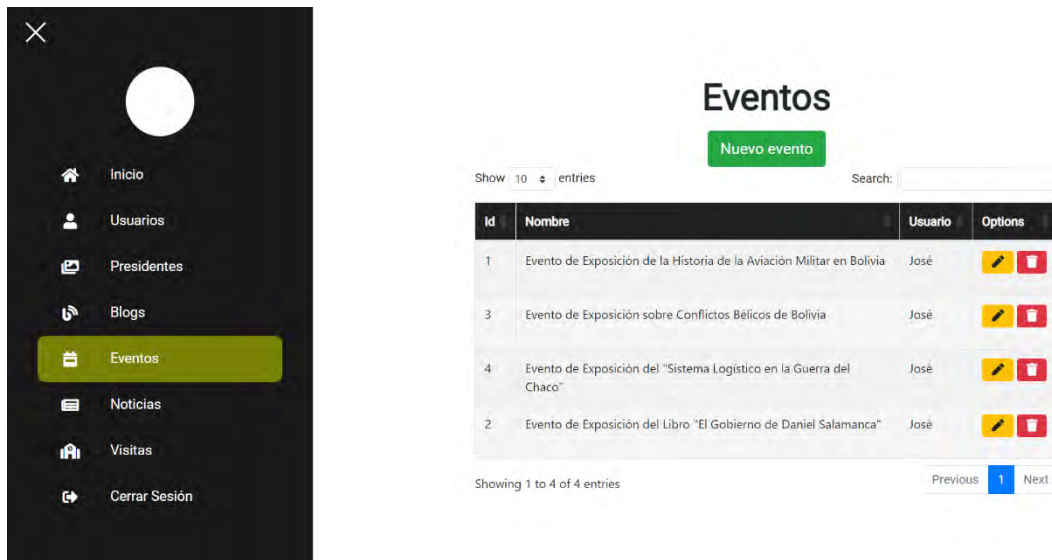


Nota: En la figura se muestra el diseño de la pantalla para la gestión visitas. Fuente: Elaboración Propia.

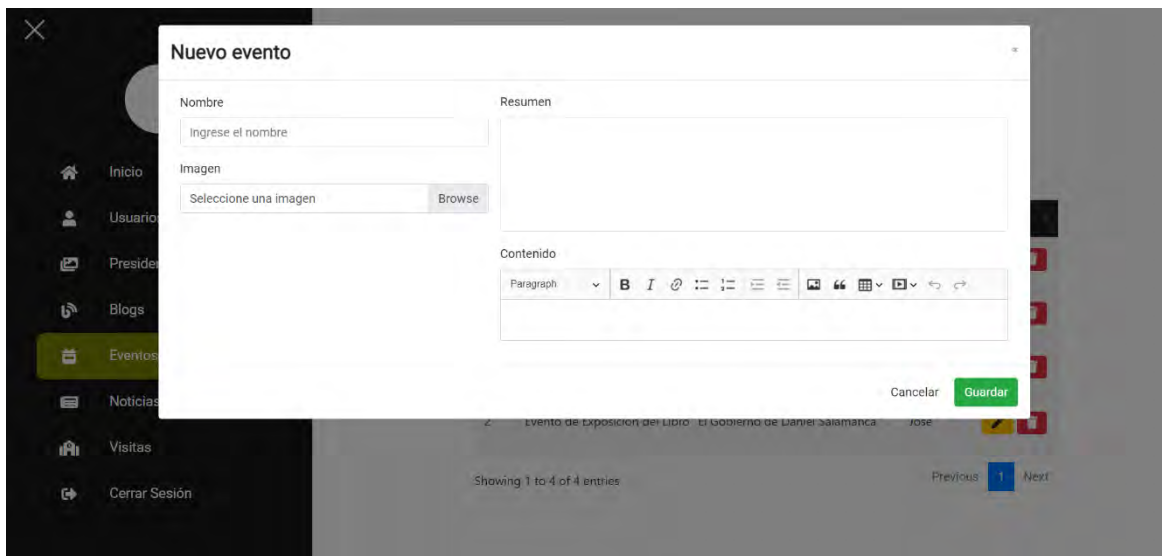
Figura 39:
Pantalla de formulario para visitas



Nota: En la figura se muestra el diseño del formulario de la pantalla para crear las visitas. Fuente: Elaboración Propia.

Figura 40:*Pantalla para gestión de eventos*

Nota: En la figura se muestra el diseño de la pantalla para la gestión de eventos. Fuente: Elaboración Propia.

Figura 41:*Pantalla de formulario para eventos*

Nota: En la figura se muestra el diseño del formulario de la pantalla para crear un evento. Fuente: Elaboración Propia.

3.4.2.6. Descripción de resultados segunda iteración

Como resultados de entrega de la segunda iteración del sistema, el cliente manifestó su satisfacción general con las funcionalidades implementadas en los módulos desarrollados, las características presentadas cumplieron con los requerimientos funcionales iniciales y permitieron validar, en un entorno real, el enfoque adoptado para el diseño y desarrollo del sistema.

3.4.3. Tercera Iteración

Para la tercera iteración del desarrollo del sistema, se han implementado diversos módulos fundamentales, los cuales incluyen: el módulo de gestión de noticias y gestión de blogs.

En la Tabla 45 se presenta un resumen general de las historias de usuario correspondientes a esta etapa, las cuales constituyeron la base para la planificación, diseño y desarrollo incremental de los módulos implementados, permitiendo una evolución ordenada y coherente del sistema.

Tabla 45:
Historias de usuario tercera iteración

Numero de historia	Nombre
7	Gestión de noticias.
8	Gestión de blogs.

Nota: En la tabla se muestran las historias de usuario que se trabajaran dentro de la tercera iteración del sistema.

Fuente: Elaboración Propia.

3.4.3.1. Tareas de Ingeniería tercera iteración

En la Tabla 46, se presenta una definición general de las pruebas de aceptación, proporcionando una visión estructurada de los alcances del sistema, para un mayor nivel de detalle se pueden ver las Tablas 47 – 50 donde describen de manera específica cada una de estas pruebas, incluyendo sus objetivos, procedimientos, condiciones de evaluación y resultados esperados.

Tabla 46:*Tareas de ingeniería segunda iteración*

Número de tarea	Número de historias	Nombre de la tarea
13	7	Diseño de la interfaz para la gestión de noticias.
14	7	Realizar pruebas de CRUD en las noticias.
15	8	Diseño de la interfaz para la gestión de blogs.
16	8	Realizar pruebas de CRUD en los blogs.

Nota: En la tabla se muestran las tareas de ingeniería de la segunda iteración. Fuente: Elaboración Propia

3.4.3.2. Descripción de las tareas de ingeniería

En esta sección se detallan las tareas de ingeniería realizadas durante la segunda iteración del proyecto, desde el análisis de requisitos hasta el diseño, implementación y validación.

Tabla 47:*Tarea de ingeniería 13, historia de usuario 7*

TAREA DE INGENIERÍA	
Número de tarea: 13	Número de historia: 7
Nombre de tarea: Diseño de la interfaz para la gestión de noticias.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 27/10/2024	Fecha fin: 03/11/2024
Programador responsable: José Antonio Aguilar Fernández	
Descripción: Se realizará el diseño de interfaz y un formulario de registro, para cargar la información de las noticias.	

Nota: En la tabla se muestra la tarea de ingeniería 13, para la historia de usuario 7. Fuente: Elaboración Propia.

Tabla 48:*Tarea de ingeniería 14, historia de usuario 7*

TAREA DE INGENIERÍA	
Número de tarea: 14	Número de historia: 7
Nombre de tarea: Realizar pruebas de CRUD en las noticias.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 27/10/2024	Fecha fin: 03/11/2024
Programador responsable: José Antonio Aguilar Fernández	
Descripción: Se podrán realizar pruebas en el cual se podrán realizar las siguientes acciones como: Ver, Editar, Actualizar y Eliminar datos de usuarios nuevos.	

Nota: En la tabla se muestra la tarea de ingeniería 8, para la historia de usuario 4. Fuente: Elaboración Propia.

Tabla 49:*Tarea de ingeniería 15, historia de usuario 8*

TAREA DE INGENIERÍA	
Número de tarea: 15	Número de historia: 8
Nombre de tarea: Diseño de la interfaz para la gestión de blogs.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 03/11/2024	Fecha fin: 10/11/2024
Programador responsable: José Antonio Aguilar Fernández	
Descripción: Se realizará el diseño de interfaz y un formulario de registro, para cargar la información de los blogs.	

Nota: En la tabla se muestra la tarea de ingeniería 15, para la historia de usuario 9. Fuente: Elaboración Propia.

Tabla 50:*Tarea de ingeniería 16, historia de usuario 8*

TAREA DE INGENIERÍA	
Número de tarea: 16	Número de historia: 8
Nombre de tarea: Realizar pruebas de CRUD en los blogs.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 03/11/2024	Fecha fin: 10/11/2024
Programador responsable: José Antonio Aguilar Fernández	
Descripción: Se podrán realizar pruebas en el cual se podrán realizar las siguientes acciones como: Ver, Editar, Actualizar y Eliminar datos de usuarios nuevos.	

Nota: En la tabla se muestra la tarea de ingeniería 16, para la historia de usuario 8. Fuente: Elaboración Propia.

3.4.3.3. Pruebas de aceptación

En la Tabla 51, se presenta una definición general de las pruebas de aceptación, proporcionando una visión estructurada de los alcances del sistema, para un mayor nivel de detalle se pueden ver las Tablas 52 – 53 donde describen de manera específica cada una de estas pruebas, incluyendo sus objetivos, procedimientos, condiciones de evaluación y resultados esperados.

Tabla 51:*Pruebas de aceptación segunda iteración*

Número de Prueba	Número de Historia	Nombre de la prueba
7	7	Gestión de noticias.
8	8	Gestión de blogs.

Nota: En la tabla se muestran las pruebas de aceptación en general, mostrando el número de prueba y de la historia de usuario. Fuente: Elaboración Propia.

3.4.3.4. Descripción de pruebas de aceptación

A continuación, se presentan las tablas correspondientes a las pruebas de aceptación definidas para el proyecto, las cuales han sido diseñadas con el objetivo de validar las funcionalidades del sistema cumpliendo con los requerimientos establecidos.

Tabla 52:

Caso de prueba gestión de noticias

CASO DE PRUEBA	
Código: 7	Número de historia de usuario: 7
Historia de usuario: Gestión de noticias.	
Condiciones de ejecución: Los usuarios asignados tendrán acceso al módulo de las noticias para realizar las acciones necesarias.	
Entrada y pasos de ejecución: - Pulsar el botón Nueva Noticias y se le abrirá una venta modal donde podrá llenar el formulario para guardarlo. - En la ventana modal se ingresan los datos solicitados por el formulario de la colección de presidentes. - Pulsar el botón Guardar y se agregara a la tabla asignada de la colección de presidentes. - Si pulsa el botón Cancelar se cerrará la ventana modal y no se guardará nada.	
Resultado esperado: El formulario tiene una interfaz intuitiva y fácil de manejar, para que los puedan ingresar todos los datos de las publicaciones correctamente y sin complicaciones.	
Evaluación de prueba: Las pruebas concluyeron satisfactoriamente.	

Nota: En la tabla se muestra el caso de prueba para la gestión de las noticias. Fuente: Elaboración Propia.

Tabla 53:

Caso de prueba gestión de blogs

CASO DE PRUEBA	
Código: 8	Número de historia de usuario: 8
Historia de usuario: Gestión de blogs.	
Condiciones de ejecución: Los usuarios asignados tendrán acceso al módulo de los blogs para realizar las acciones necesarias.	
Entrada y pasos de ejecución: - Pulsar el botón Nuevo Blog y se le abrirá una venta modal donde podrá llenar el formulario para guardarlo. - En la ventana modal se ingresan los datos solicitados por el formulario de las visitas. - Pulsar el botón Guardar y se agregara a la tabla asignada de la colección de presidentes. - Si pulsa el botón Cancelar se cerrará la ventana modal y no se guardará nada.	
Resultado esperado: El formulario tiene una interfaz intuitiva y fácil de manejar, para que los puedan ingresar todos los datos de las publicaciones correctamente y sin complicaciones.	
Evaluación de prueba: Las pruebas concluyeron satisfactoriamente.	

Nota: En la tabla se muestra el caso de prueba para la gestión de los blogs. Fuente: Elaboración Propia.

A continuación, en las Figuras 42 – 43 se presentarán los mockups base diseñados para el sistema, estos prototipos permiten validar la usabilidad, identificar mejoras en la navegación y garantizar que el diseño cumpla con los requerimientos establecidos del sistema.

Figura 42:

Mockup del formulario y dashboard de noticias

Noticias Actuales Agregar

ID	Título	Fecha	Contenido	Imagen	Acciones
1	Noticia # 1	15-10-2023	Texto	Image # 1	✓ ✗
2	Noticia # 2	19-10-2023	Texto	Image # 2	✓ ✗
3	Noticia # 3	20-11-2023	Texto	Image # 3	✓ ✗

Noticia

Título:

Imagen:

Cancelar Agregar

Nota: En la figura se muestra el mockup del formulario y dashboard de noticias. Fuente: Elaboración Propia.

Figura 43:

Mockup del formulario y dashboard de blogs

Publicación de Blogs Agregar

ID	Título	Fecha	Contenido	Imagen	Acciones
1	Blog # 1	15-10-2023	Texto	Image # 1	✓ ✗
2	Blog # 2	19-10-2023	Texto	Image # 2	✓ ✗
3	Blog # 3	20-11-2023	Texto	Image # 3	✓ ✗

Nuevo Blog

Título:

Categoría:

Contenido:

Cancelar Agregar

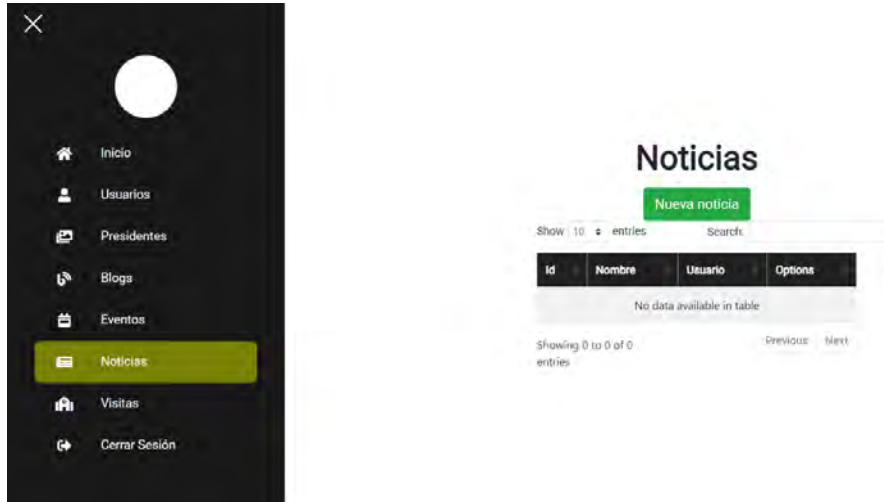
Nota: En la figura se muestra el mockup del formulario y dashboard de blogs. Fuente: Elaboración Propia.

A continuación, en las Figuras 44 – 47 se muestran una serie de capturas de pantalla que documentan visualmente el desarrollo del sistema, en base los mockups diseñados anteriormente, así como de los distintos módulos que lo componen, estas imágenes corresponden a evidencias gráficas recolectadas durante el proceso de diseño y validación funcional, en el cual se contó con la participación activa del cliente, cada una de las capturas refleja el resultado de un trabajo colaborativo.

3.4.3.5. Capturas de pantalla tercera iteración

Figura 44:

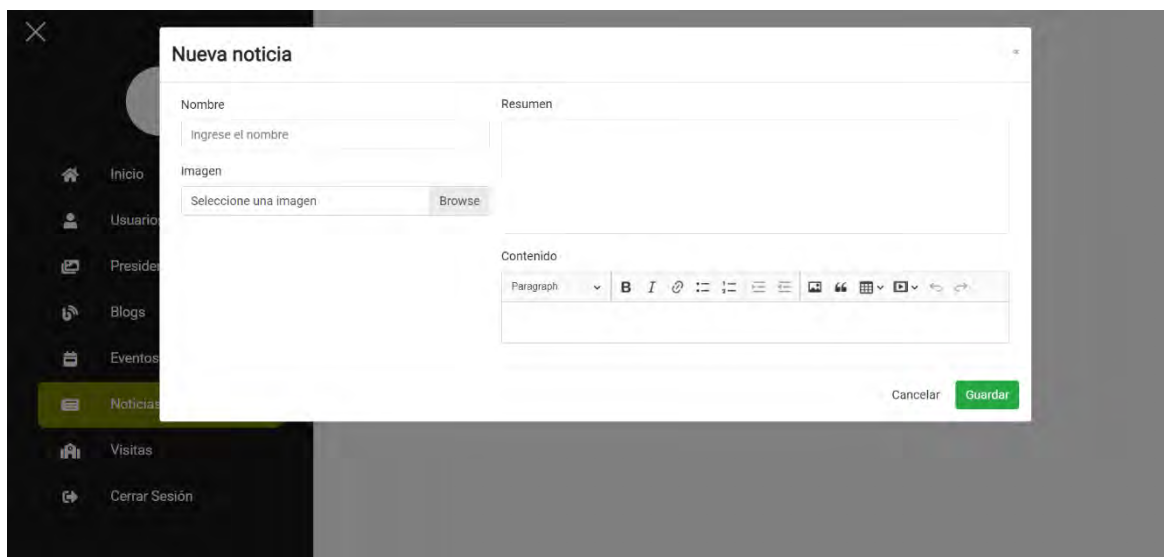
Pantalla para gestión de noticias



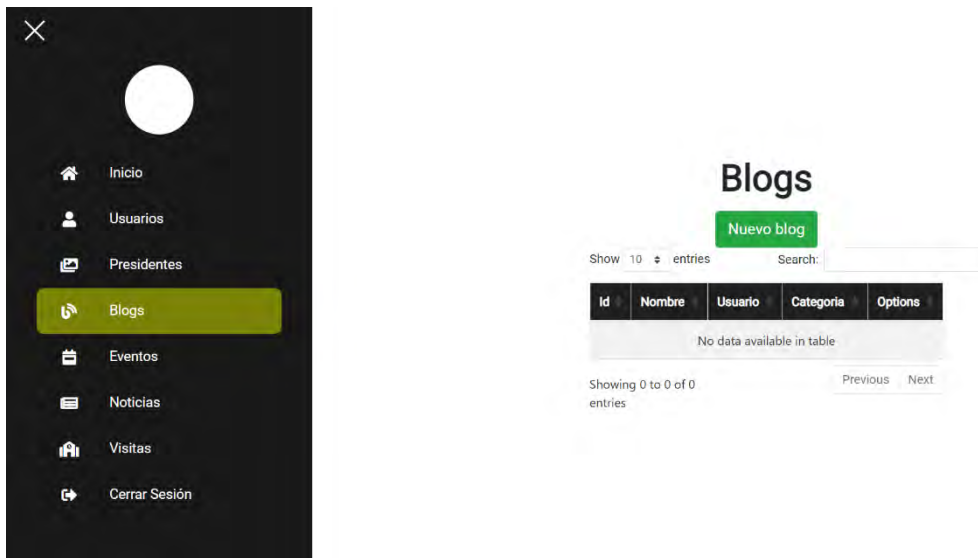
Nota: En la figura se muestra el diseño de la pantalla para las noticias y la tabla en donde se puede ver a todas las noticias ya agregadas. Fuente: Elaboración Propia.

Figura 45:

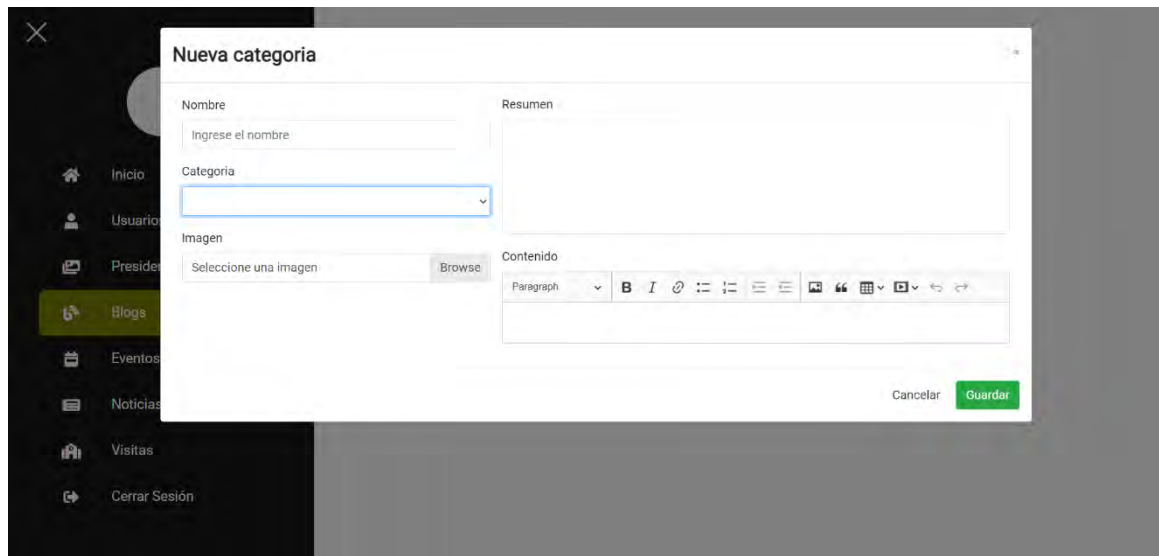
Pantalla de formulario para noticias



Nota: En la figura se muestra el diseño del formulario de la pantalla para crear las noticias. Fuente: Elaboración Propia.

Figura 46:*Pantalla para gestión de blogs*

Nota: En la figura se muestra el diseño de la pantalla para los blogs y la tabla en donde se puede ver a todos los blogs ya agregados. Fuente: Elaboración Propia.

Figura 47:*Pantalla de formulario para blogs*

Nota: En la figura se muestra el diseño del formulario de la pantalla para crear un blog. Fuente: Elaboración Propia.

3.4.3.6. Descripción de resultados tercera iteración

Como resultados de entrega de la segunda iteración del sistema, el cliente manifestó su satisfacción general con las funcionalidades implementadas en los módulos desarrollados, las características presentadas cumplieron con los requerimientos funcionales iniciales y permitieron validar, en un entorno real, el enfoque adoptado para el diseño y desarrollo del sistema.

3.4.4. Cuarta Iteración

Para la cuarta y última iteración del desarrollo del sistema, se han implementado diversos módulos fundamentales, los cuales incluyen: el módulo de análisis comparativo, el algoritmo de enmascaramiento y la implementación del algoritmo.

A continuación, se presenta un análisis comparativo de diversas técnicas de enmascaramiento de datos, realizado con el objetivo de identificar la opción más adecuada para proteger la información sensible dentro del sistema, este análisis evalúa las características, ventajas y limitaciones de cada técnica, proporcionando una base sólida para la selección de la estrategia de enmascaramiento implementada en el proyecto.

3.4.4.1. Técnica de enmascaramiento de datos

- Proteger datos sensibles reemplazando su valor real.
- Nivel de protección Medio / Alto.
- No se tiene el acceso a los datos originales.
- Su utiliza en entornos de desarrollo, Testing y capacitación.
- Protege los datos reales en pruebas sin comprometerse la seguridad.
- No es útil si se requiere recuperar los valores originales.

3.4.4.2. Técnica de cifrado / encriptación

- Convertir datos en un formato ilegible sin clave.
- Nivel de protección Alto.
- Se tiene el acceso a los datos originales mediante de claves.
- Se utiliza una comunicación segura en el almacenamiento de datos.

- Alta confidencialidad y control mediante claves.
- Riesgo si se pierde o compromete la clave, más costoso computacionalmente.

3.4.4.3. Técnica de hashing

- Verificación de integridad y autenticación.
- Nivel de protección Muy Alto irreversible.
- No se tiene acceso a los datos originales.
- El uso más común se ve en las contraseñas y firmas digitales.
- Es rápido, seguro e irreversible.
- No permite recuperar el dato original.

3.4.4.4. Técnica de tokenización

- Sustituir datos sensibles por identificadores únicos.
- Nivel de protección Alto.
- Se tiene acceso a los datos originales mediante mapa de tokens.
- Se usa en el procesamiento de tarjetas de crédito y bases de datos.
- Protección efectiva en tiempo real y requiere administración del sistema de tokens.

El enmascaramiento de datos es ideal para el presente proyecto porque:

- El propósito del sistema está relacionado con el manejo de información sensible en una institución académica, posiblemente con datos históricos, personales o institucionales.
- Este tipo de datos no necesita ser procesado por usuarios externos en su forma original, por lo que no es necesario descifrarlos posteriormente, como ocurre en el caso de la encriptación.
- Se busca mantener la integridad de las estructuras de datos para pruebas, desarrollo o capacitación, sin comprometer la privacidad de la información real.
- A diferencia del cifrado, el enmascaramiento no requiere de gestión de claves, lo que simplifica su implementación y reduce riesgos asociados al robo de claves.

Como explica (Garfinkel, 2015), el enmascaramiento de datos forma parte de las técnicas de desidentificación, que tienen el objetivo de proteger la privacidad individual y reducir el riesgo de divulgación de datos personales en entornos no productivos o inseguros.

3.4.4.5. Desarrollo del Algoritmo de enmascaramiento

El diagrama de flujo describe el proceso lógico para acceder o modificar atributos que pueden estar encriptados dentro de un sistema:

- El algoritmo inicia verificando si la encriptación está habilitada.
- Si no está habilitada, los atributos son accedidos o modificados de manera normal, sin aplicar ningún tipo de transformación.
- Si la encriptación está habilitada, se verifica si el atributo en cuestión es considerado enmascarable.
 - Si no es enmascarable, simplemente se accede o modifica su valor sin cifrado adicional.
 - Si sí es enmascarable, el sistema decidirá si se necesita acceder al valor o modificarlo:

Acceder: Se descifra el valor del atributo antes de entregarlo.

Modificar: Se cifra el nuevo valor antes de guardarlo, protegiendo así la integridad y confidencialidad de la información.

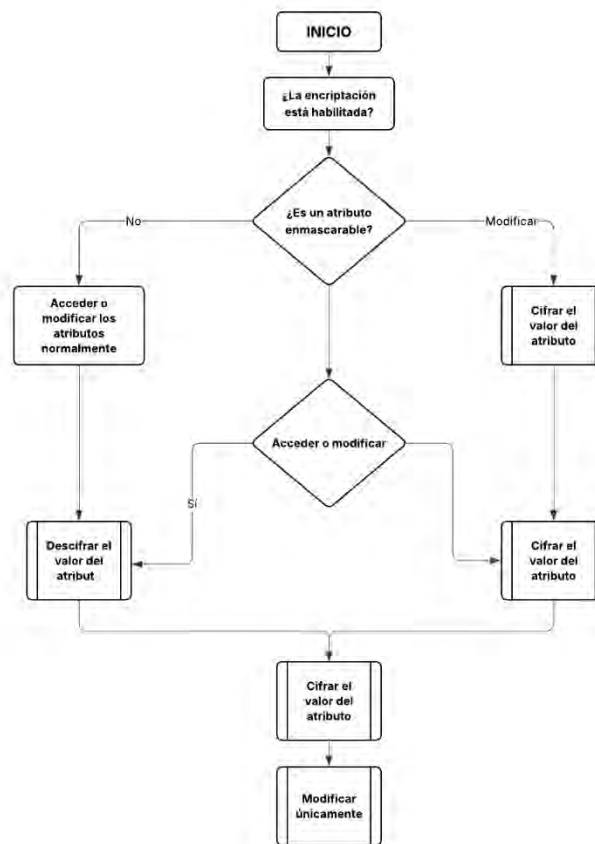
Este flujo asegura que todos los atributos sensibles se mantengan protegidos automáticamente, según las configuraciones definidas, garantizando seguridad y control sin afectar la operatividad normal del sistema.

Una vez elaborado el diagrama de flujo que representa la estructura del algoritmo de enmascaramiento de datos, se procede a detallar de manera secuencial cada etapa del código implementado, esta explicación técnica tiene como propósito guiar la comprensión del funcionamiento interno, asegurando una correcta interpretación del procedimiento antes de su ejecución práctica dentro del sistema.

Posteriormente, se lleva a cabo la implementación directa en la base de datos, lo que permite verificar su comportamiento en un entorno real, a través de este proceso, se evalúa si el algoritmo actúa conforme a lo esperado o si presenta inconsistencias que deban ser corregidas, en caso de identificarse errores, se aplicarán ajustes pertinentes para optimizar el mecanismo de enmascaramiento y garantizar la integridad y confidencialidad de la información gestionada.

Figura 48:

Algoritmo de enmascaramiento de datos



Nota: En la figura se muestra el diagrama de flujos del algoritmo de enmascaramiento. Fuente: Elaboración Propia.

3.4.4.6. Implementación del algoritmo

En esta parte del proyecto se realizará la implementación del algoritmo realizado para realizar las pruebas de enmascaramiento de datos. Ver Figuras 49 – 54.

Figura 49:

Declarando funciones

```

namespace App\Traits;

use App\Traits\Builders\EncryptionEloquentBuilder;
use App\Traits\Encrypter;
  
```

Nota: En la figura el cuadro se muestra la declaración de funciones. Fuente: Elaboración Propia.

Se utiliza una variable estática (\$enableEncryption) para controlar si la encriptación está activa o no en toda la aplicación.

Se define un arreglo protegido (\$encryptable) donde se listan los nombres de los atributos que deben ser encriptados automáticamente.

La función isEncryptable(\$key) verifica si un atributo específico debe ser encriptado, comprobando que: la encriptación esté habilitada y el atributo esté en la lista de atributos definidos como encriptables.

Figura 50:

Código algoritmo 1

```
// Indicamos que la encriptación está habilitada o no
public static $enableEncryption = true;

// Se puede cargar una lista de atributos para encriptar
protected $encryptable = [];

/**
 * Verificamos si el atributo es encriptable.
 *
 * @param string $key
 * @return bool
 */
public function isEncryptable($key)
{
    // Verifica si la encriptación está habilitada y si el atributo está en la lista de encriptables
    if (self::$enableEncryption) {
        return in_array($key, $this->encryptable);
    }
    return false;
}
```

Nota: En la figura se puede observar la primera parte del algoritmo. Fuente: Elaboración Propia.

La función getEncryptableAtributos () realizará las siguientes acciones:

- Devuelve la lista completa de atributos marcados como encriptables.
- Un array con los nombres de los atributos que deben cifrarse.
- Ayuda a conocer qué campos están protegidos con encriptación.

La función getAttribute(\$key) realizará las siguientes acciones:

- Sobrescribe la función original para obtener un atributo del modelo.
- Llama a la función original (parent::getAttribute(\$key)) para obtener el valor.
- Verifica si ese atributo debe ser encriptado con isEncryptable(\$key).
- Si es así, y el valor no es nulo ni vacío, intenta desencriptarlo usando Encrypter::decrypt(\$value).
- Si ocurre una excepción durante la desencriptación, la ignora silenciosamente.
- El valor original o desencriptado del atributo.

Este diseño garantiza que, cuando accedes a un atributo cifrado desde el modelo (por ejemplo, \$usuario->correo), el sistema internamente lo desencripta automáticamente, sin necesidad de hacerlo manualmente en cada parte del código.

Figura 51:

Código de algoritmo 2

```
/**
 * Obtenemos la lista de atributos encriptables.
 *
 * @return array
 */
public function getEncryptableAttributes()
{
    return $this->encryptable;
}

/**
 * Obtenemos los atributos
 *
 * @param string $key
 * @return mixed
 */
public function getAttribute($key)
{
    $value = parent::getAttribute($key);

    // Si el atributo es encryptable se tiene un valor no nulo y no vacío
    if ($this->isEncryptable($key) && (!is_null($value) && $value != '')) {
        try {
            $value = Encrypter::decrypt($value);
        } catch (\Exception $th) {
        }
    }
    return $value;
}
```

Nota: En la figura se puede observar la segunda parte del algoritmo. Fuente: Elaboración Propia.

La función `setAttribute($key, $value)` realizara las siguientes acciones:

- Establecer el valor de un atributo del modelo.
- Parámetros (\$key: el nombre del atributo - \$value: el valor a asignar).
- Verifica si el atributo está marcado como encriptable usando `$this->isEncryptable($key)`.
- Si es encriptable, intenta encriptar el valor usando `Encrypter::encrypt($value)`.
- Si ocurre una excepción durante la encriptación, se captura pero no se interrumpe el flujo.
- Finalmente, el atributo (ya encriptado o sin encriptar, según el caso) se asigna mediante la función original `parent::setAttribute()`.
- Devuelve el modelo actual con el atributo asignado (`$this`), lo cual permite encadenamiento de métodos.

Este método protege automáticamente los datos sensibles en el momento de su almacenamiento, sin necesidad de que el programador tenga que cifrar los datos manualmente cada vez, así se centraliza la seguridad y se reduce el riesgo de errores humanos.

Figura 52:

Código de algoritmo 3

```
/**
 * Establecemos un atributo
 *
 * @param string $key
 * @param mixed $value
 * @return $this
 */
public function setAttribute($key, $value)
{
    // Si el atributo es encriptable
    if ($this->isEncryptable($key)) {
        try {
            $value = Encrypter::encrypt($value);
        } catch (\Exception $th) {
        }
    }
    return parent::setAttribute($key, $value);
}
```

Nota: En la figura se puede observar la tercera parte del algoritmo. Fuente: Elaboración Propia.

La función `setAttribute($key, $value)` realizara las siguientes acciones:

- Retorna un array con los atributos del modelo.
- Se itera sobre cada clave-valor del array.
- Se intenta descriptar el valor mediante `Encrypter::decrypt($value)`.
- Si ocurre una excepción en el proceso de descriptación, esta es capturada y omitida sin lanzar error.
- Método `isEncryptable()`: Verifica si una clave específica está marcada para encriptación (debería estar definido en la clase).
- El bloque `try-catch` permite continuar con el procesamiento incluso si la descriptación falla, lo cual añade robustez.

Este tipo de implementación permite mantener confidencialidad de los datos en reposo y aplicar descriptación solo cuando es necesario (por ejemplo, al mostrar información al

usuario). Se utiliza ampliamente para cumplir con normativas de privacidad de datos personales.

Figura 53:

Código de algoritmo 4

```
/**
 * Convertimos los atributos en un array
 *
 * @return array
 */
public function attributesToArray()
{
    $attributes = parent::attributesToArray();

    // Si existen atributos son encriptables
    if ($attributes) {
        foreach ($attributes as $key => $value) {
            if ($this->isEncryptable($key) && (!is_null($value)) && $value != '') {
                try {
                    $attributes[$key] = Encrypter::decrypt($value);
                } catch (\Exception $th) {
                }
            }
        }
    }
    return $attributes;
}
```

Nota: En la figura se puede observar la cuarta parte del algoritmo. Fuente: Elaboración Propia.

La función `newEloquentBuilder($query)` realizara las siguientes acciones:

- Redefine el método constructor del Query Builder de Eloquent para que en lugar del constructor estándar, utilice una clase personalizada `EncryptionEloquentBuilder`.
- Esto permite extender las capacidades de Eloquent para trabajar con atributos encriptados de forma transparente.

La función `decryptAttribute($value)` realizara las siguientes acciones:

- Recibe un valor que podría estar encriptado.
- Si existe, lo desencripta utilizando `Encrypter::decrypt($value)`.
- Si el valor es null o vacío, retorna una cadena vacía.

- Esta función es clave para mostrar datos ya encriptados de forma legible en las vistas o procesos internos del sistema.

La función `encryptAttribute($value)` realizara las siguientes acciones:

- Toma un valor de entrada y lo encripta mediante `Encrypter::encrypt($value)`.
- También retorna una cadena vacía si el valor está vacío.
- Ideal para almacenar datos sensibles como nombres, direcciones o identificaciones personales de manera segura en la base de datos.

A continuación, se presentan los segmentos finales del código fuente, los cuales contienen funciones esenciales que serán utilizadas por el sistema, estas cumplen un rol clave en la ejecución del algoritmo de enmascaramiento de datos, ya que permiten su integración efectiva con los distintos módulos desarrollados durante el proyecto.

Una vez concluida la implementación de dichas funciones, se procederá a realizar una serie de pruebas para validar su correcto funcionamiento, estas consisten en ejecutar el algoritmo dentro del entorno real de la base de datos, observando su comportamiento ante distintos tipos de información sensible que requieren ser protegidos.

Finalmente, se incluirán ejemplos prácticos que reflejan cómo se ha aplicado el proceso de enmascaramiento en los módulos construidos, demostrando la eficacia del enfoque adoptado, esta etapa permite no solo comprobar que los datos están siendo transformados adecuadamente, sino también verificar que el sistema mantiene su operatividad sin afectar la integridad ni el acceso a la información para los usuarios autorizados.

Figura 54:

Código de algoritmo 5

```

/**
 * Extiende EncryptionEloquentBuilder.
 *
 * @param \Illuminate\Database\Eloquent\Builder $query
 * @return \Tu\Namespace\Builders\EncryptionEloquentBuilder
 */
public function newEloquentBuilder($query)
{
    return new EncryptionEloquentBuilder($query);
}

/**
 * Desencripta un valor de atributo.
 *
 * @param mixed $value
 * @return mixed
 */
public function decryptAttribute($value)
{
    return $value ? Encrypter::decrypt($value) : '';
}

/**
 * Encripta un valor de atributo.
 *
 * @param mixed $value
 * @return mixed
 */
public function encryptAttribute($value)
{
    return $value ? Encrypter::encrypt($value) : '';
}

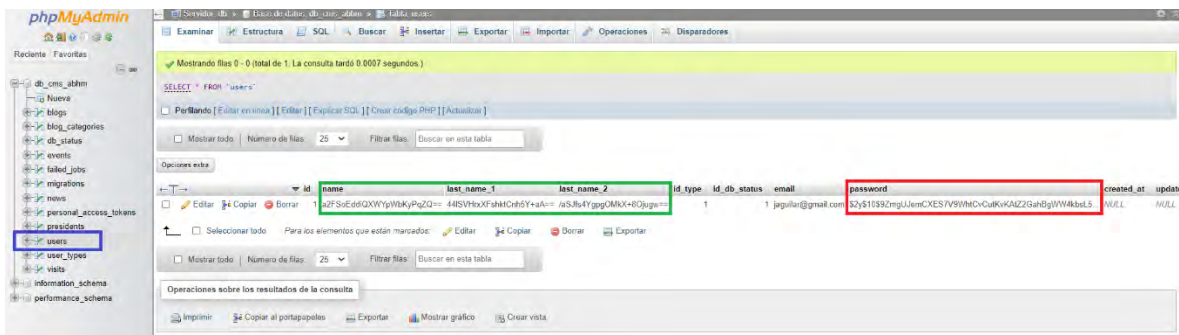
```

Nota: En la figura se puede observar la cuarta parte del algoritmo. Fuente: Elaboración Propia.

3.4.4.7. Capturas de pantalla cuarta iteración

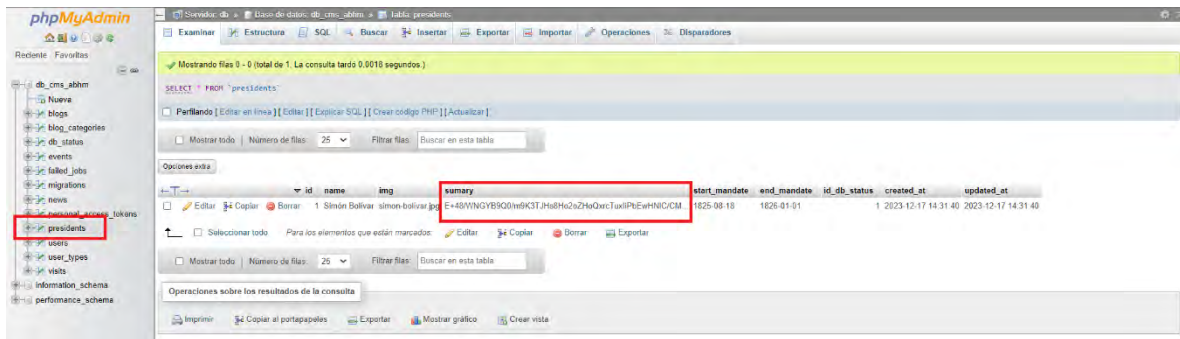
Figura 55:

Algoritmo aplicado en gestión de usuarios



Nota: En la figura se muestra cómo se aplica el enmascaramiento en las columnas seleccionadas desde el algoritmo para la tabla usuarios. Fuente: Elaboración Propia.

Figura 56:
Algoritmo aplicado en gestión de colección de presidentes



Mostrando filas 0 - 0 (total de 1. La consulta tardó 0.0018 segundos)

```
SELECT * FROM `presidentes`
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Mostrar todo | Número de filas: 25 | Filtrar filas | Buscar en esta tabla

Opciones extra

id	name	img	summary	start_mandate	end_mandate	id_db_status	created_at	updated_at
1	Simón Bolívar	simon-bolivar.jpg	E=48WNGYBSQ0v6KSTJHh8H2aZHoQwCtuxlP6EwHNCQM	1625-08-18	1625-01-01	1	2023-12-17 14:31:40	2023-12-17 14:31:40

Seleccionar todo | Para los elementos que están marcados: [Editar] [Copiar] [Borrar] [Exportar]

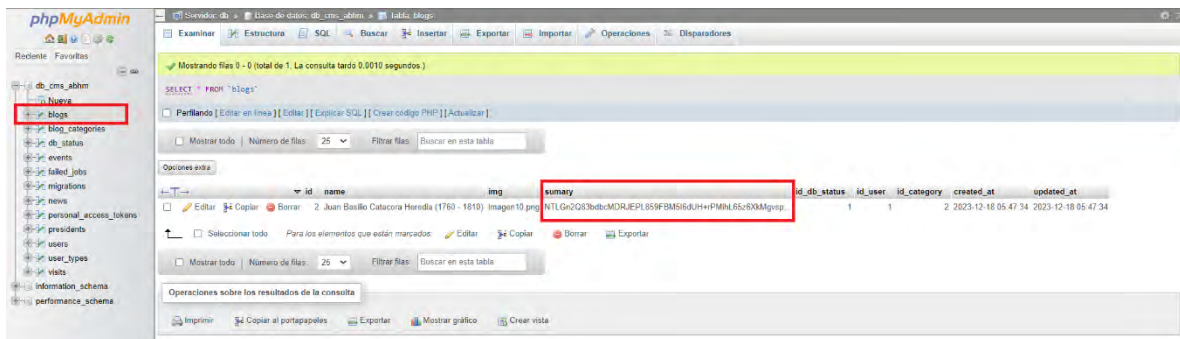
Mostrar todo | Número de filas: 25 | Filtrar filas | Buscar en esta tabla

Operaciones sobre los resultados de la consulta

[Imprimir] [Copiar al portapapeles] [Exportar] [Mostrar gráfico] [Crear vista]

Nota: En la figura se muestra cómo se aplica el enmascaramiento en las columnas seleccionadas desde el algoritmo para la tabla presidentes. Fuente: Elaboración Propia.

Figura 57:
Algoritmo aplicado en gestión de blogs



Mostrando filas 0 - 0 (total de 1. La consulta tardó 0.0010 segundos)

```
SELECT * FROM `blogs`
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Mostrar todo | Número de filas: 25 | Filtrar filas | Buscar en esta tabla

Opciones extra

id	name	img	summary	id_db_status	id_user	id_category	created_at	updated_at
2	Juan Basilio Caceres Heredia (1760 - 1810)	Imagen12.jpg	NTLGrQ283bdcMDRUEP1859F8MS6HJH+PMH4.65+5XMMytp	1	1	2	2023-12-18 05:47:34	2023-12-18 05:47:34

Seleccionar todo | Para los elementos que están marcados: [Editar] [Copiar] [Borrar] [Exportar]

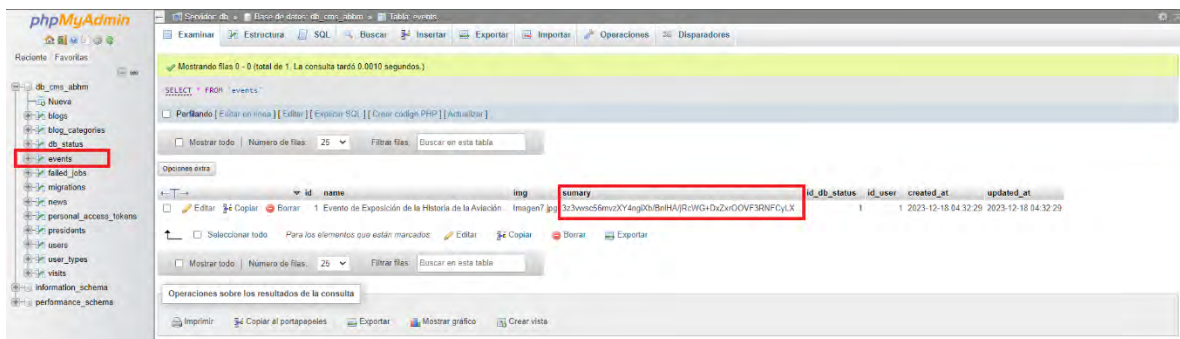
Mostrar todo | Número de filas: 25 | Filtrar filas | Buscar en esta tabla

Operaciones sobre los resultados de la consulta

[Imprimir] [Copiar al portapapeles] [Exportar] [Mostrar gráfico] [Crear vista]

Nota: En la figura se muestra cómo se aplica el enmascaramiento en las columnas seleccionadas desde el algoritmo para la tabla blogs. Fuente: Elaboración Propia.

Figura 58:
Algoritmo aplicado en gestión de eventos



Mostrando filas 0 - 0 (total de 1. La consulta tardó 0.0010 segundos)

```
SELECT * FROM `events`
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Mostrar todo | Número de filas: 25 | Filtrar filas | Buscar en esta tabla

Opciones extra

id	name	img	summary	id_db_status	id_user	created_at	updated_at
1	Evento de Exposición de la Historia de la Aviación	Imagen7.jpg	3z3vnc56mzY4ng0bBnHA/RcWG+Dz2xCOVFSRNFcyX	1	1	2023-12-18 04:32:29	2023-12-18 04:32:29

Seleccionar todo | Para los elementos que están marcados: [Editar] [Copiar] [Borrar] [Exportar]

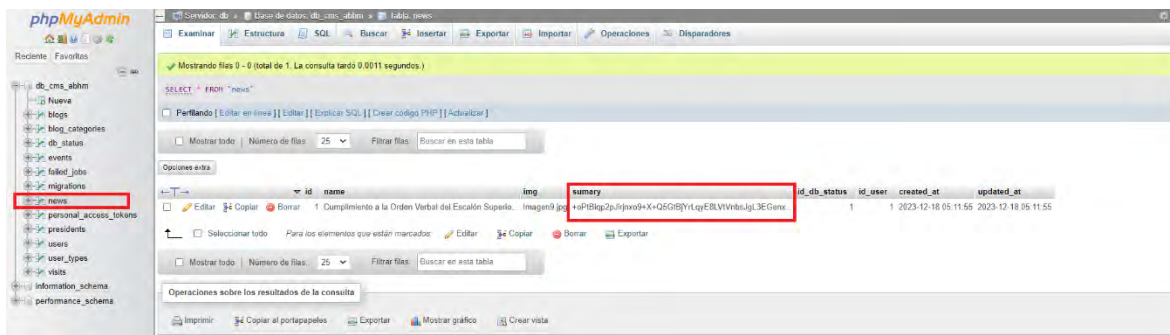
Mostrar todo | Número de filas: 25 | Filtrar filas | Buscar en esta tabla

Operaciones sobre los resultados de la consulta

[Imprimir] [Copiar al portapapeles] [Exportar] [Mostrar gráfico] [Crear vista]

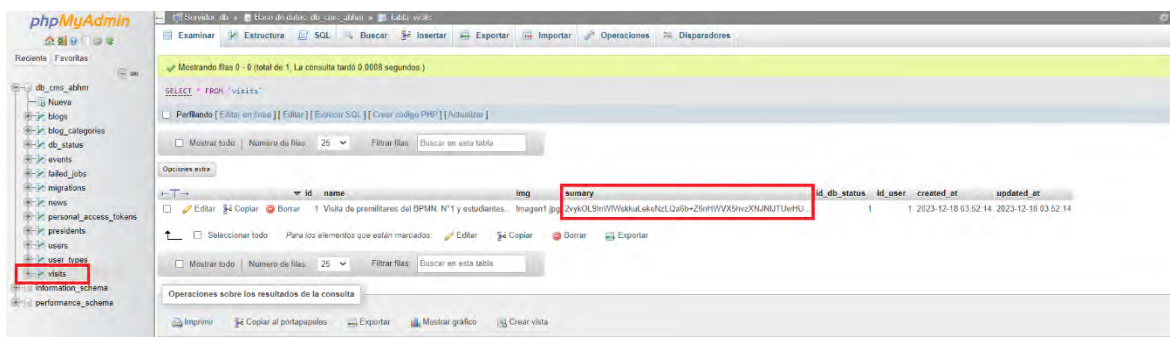
Nota: En la figura se muestra cómo se aplica el enmascaramiento en las columnas seleccionadas desde el algoritmo para la tabla eventos. Fuente: Elaboración Propia.

Figura 59:
Algoritmo aplicado en gestión de noticias



Nota: En la figura se muestra cómo se aplica el enmascaramiento en las columnas seleccionadas desde el algoritmo para la tabla noticias. Fuente: Elaboración Propia.

Figura 60:
Algoritmo aplicado en gestión de visitas



Nota: En la figura se muestra cómo se aplica el enmascaramiento en las columnas seleccionadas desde el algoritmo para la tabla visitas. Fuente: Elaboración Propia.

3.4.4.8. Descripción de resultados cuarta iteración

Como resultado de la entrega correspondiente a la cuarta iteración del sistema, se logró completar con éxito el análisis comparativo de diversas técnicas de enmascaramiento de datos, seleccionando la más adecuada para los requerimientos del proyecto, posteriormente, se procedió con el desarrollo del algoritmo de enmascaramiento y su respectiva implementación en la base de datos, actualmente el algoritmo se encuentra funcionando de manera correcta, aplicándose en todas las partes del sistema que fueron solicitadas, cumpliendo así con los objetivos establecidos para esta fase y validando la efectividad del enfoque técnico adoptado.

La integración de las técnicas de enmascaramiento de datos en el sistema de gestión de contenidos ha sido complementada con una interfaz visual intuitiva, diseñada para mejorar la experiencia del usuario, se incorporaron elementos gráficos representativos, esquemas claros y una paleta de colores amigable que facilita la navegación y reduce la carga cognitiva, estos aspectos visuales permiten que los usuarios identifiquen rápidamente las funcionalidades principales, comprendan el flujo de la información enmascarada y operen con mayor seguridad, especialmente al momento de realizar publicaciones o consultar registros sensibles, de esta manera, se garantiza no solo la protección de la información, sino también una interacción eficiente, accesible y comprensible.

Tabla 54:

Incorporación de elementos visuales

INCORPORACIÓN DE ELEMENTOS VISUALES	
Elementos visuales técnicos	Navegación compresible
Paleta de colores amigable.	Reduce el estrés visual y permite identificar secciones fácilmente.
Íconos intuitivos.	Ayudan a reconocer funciones como editar, ver, o proteger datos.
Estructura modular clara.	Permite que el usuario no se pierda al interactuar con distintas funcionalidades.
Mensajes emergentes (tooltips).	Explican brevemente las funciones de cada botón o acción disponible.
Navegación por pestañas o menús.	Facilita el acceso organizado a cada sección del sistema.

Nota: En la tabla se muestra cómo se aplican los elementos visuales técnicos en la navegación y comprensión de los elementos visuales del sistema. Fuente: Elaboración Propia.

Adicionalmente, se desarrollaron estructuras modulares en cada vista, acompañadas de esquemas bien definidos que organizan la información en forma jerárquica y contextual, la implementación del enmascaramiento se reforzó con señales como mensajes emergentes, etiquetas distintivas que informan al usuario cuando los datos visibles han sido protegidos o requieren privilegios para su visualización, esto no solo fortalece la seguridad de los datos publicados, sino que también promueve una experiencia de navegación fluida, clara y segura, en conjunto estas mejoras visuales y funcionales aseguran que el sistema no solo cumpla con los estándares de protección, sino que también facilite el cumplimiento de las actividades institucionales de la Academia, garantizando precisión, accesibilidad y confiabilidad en el manejo de su contenido histórico.

3.4.5. Pruebas unitarias

Las pruebas unitarias constituyen una fase fundamental dentro del proceso de desarrollo del sistema, ya que permiten verificar el correcto funcionamiento de cada uno de los componentes individuales que conforman el sistema de gestión de contenidos, estas pruebas se centran en evaluar, de manera aislada, funciones, métodos o módulos específicos, con el propósito de asegurar que operen conforme a lo esperado y cumplan con los requerimientos previamente definidos.

3.4.5.1. Prueba unitaria para el módulo de colección de presidentes

El módulo de colección de presidentes tiene como objetivo registrar, gestionar y visualizar las imágenes y datos históricos correspondientes a los presidentes que forman parte del archivo documental de la Academia, para asegurar su correcto funcionamiento, se realizaron pruebas unitarias enfocadas en validar cada una de sus funciones individuales.

Entre los aspectos evaluados se encuentran la correcta carga de imágenes, el registro de información asociada nombre, periodo presidencial, breve descripción, y la correcta visualización de los datos en la interfaz del sistema, como podemos ver a continuación:

Figura 61:

Creación del registro de presidente

Nuevo presidente

Nombre: Simón Bolívar

Fecha de nacimiento: 18/08/1783

Inicio de mandato: 18/01/1823

Fin de mandato: 21/01/1830

Imagen:

Resumen

Simón Bolívar: Solamente los soliflores pueden realizar grandes cosas. Quiénes se imponen objetivos superiores y entregan su vida a su patria en batalla, son los héroes. Si no es un lugar en la historia, en Simón Bolívar y más especialmente en su hijo, Simón Bolívar, este primer lugar histórico, en el mundo y en la acción, es de Simón Bolívar.

Fecha y hora de nacimiento: En Caracas, Venezuela, el 24 de julio de 1783.

Título y nombre: Primer Presidente Vitalicio de la República Boliviana. Líder de la revolución de independencia del Sur, Colombia, Venezuela, Ecuador, Perú y Bolivia. Conquistó su vida a la independencia de América y además de ser un guerrero, culta y trabajo para crear una gran federación de estados, ideal que no pudo tener realidad.

Exaltación al mundo: Padre de la Patria desde el 18 de agosto de 1823.

Edad al asumir el cargo: 42 años.

Comienzo de su mandato: 17 de enero de 1824.

Tiempo de su mandato: 4 meses y 15 días.

Muerte: Muere en Santa Marta (Colombia) el 17 de diciembre de 1830.

ACTOS ADMINISTRATIVOS

Fue el fundador de la República.

Primer presidente de la república de Bolivia.

Primer presidente y autor de la Primera Constitución Política del Estado.

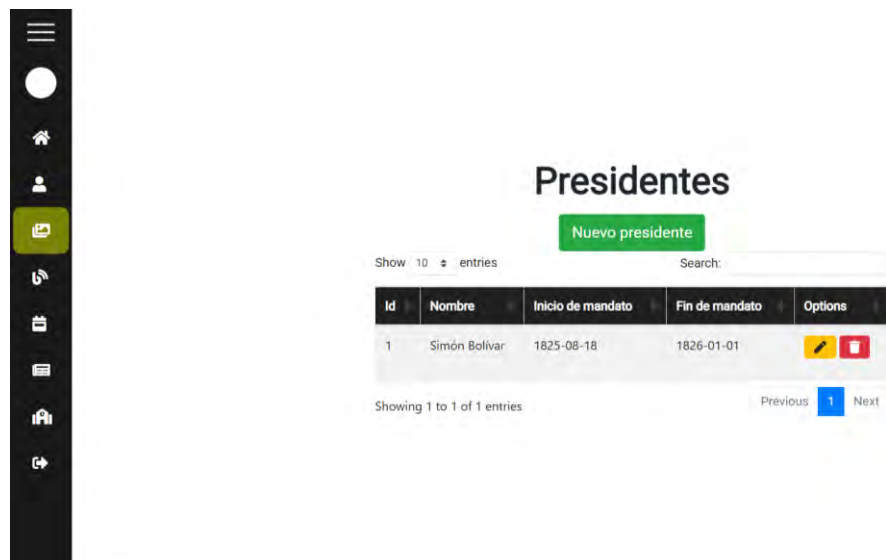
Simón Bolívar: Solamente los soliflores pueden realizar grandes cosas. Quiénes se imponen objetivos superiores y entregan su vida a su patria en batalla, son los héroes. Si no es un lugar en la historia, en Simón Bolívar y más especialmente en su hijo, Simón Bolívar, este primer lugar histórico, en el mundo y en la acción, es de Simón Bolívar.

Cancelar Guardar

Nota: En la figura se muestra cómo se crea y registra un presidente para la colección de presidentes de la Academia. Fuente: Elaboración Propia.

Figura 62:

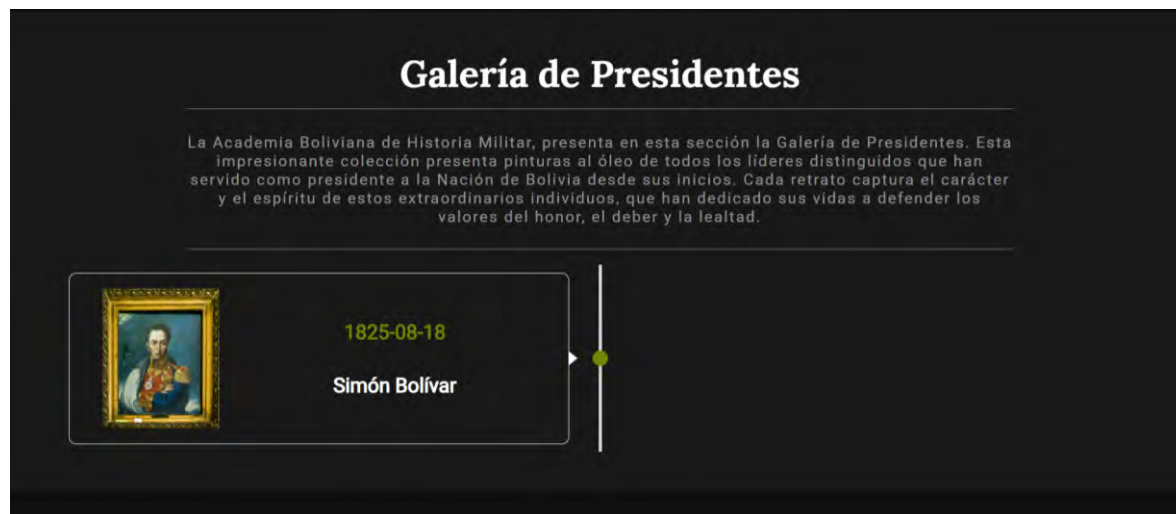
Tabla de visualización de registro de presidentes.



Nota: En la figura se muestra cómo se registra y se guarda el presidente creado para visualizarse en la tabla administrativa. Fuente: Elaboración Propia.

Figura 63:

Primera vista del sitio web, mostrando una tarjeta de presentación



Nota: En la figura se muestra cómo se visualiza en el sitio web la creación de presidente, presentándonos la fecha de inicio y el nombre. Fuente: Elaboración Propia.

Figura 64:

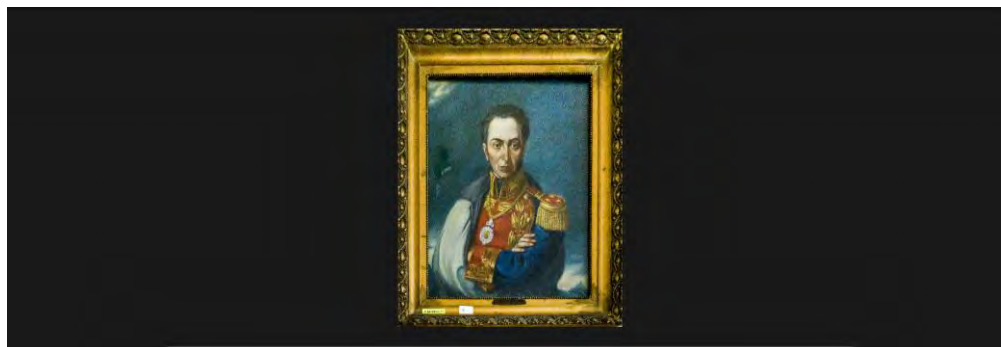
Segunda vista del sitio web, mostrando una venta modal



Nota: En la figura se observa un modal donde se ve datos resumidos del presidente. Fuente: Elaboración Propia.

Figura 65:

Presentación de la fotografía del presidente seleccionado



Nota: En la figura se muestra como primer elemento la foto del presidente. Fuente: Elaboración Propia.

Figura 66:

Vista de la información completa del presidente seleccionado



Nota: En la figura se observa la información completa del presidente. Fuente: Elaboración Propia.

3.4.5.2. Prueba unitaria para el módulo de blogs

El módulo de blogs, denominado “30 segundos con la historia”, fue desarrollado para permitir la creación, edición y visualización de entradas breves relacionadas con hechos históricos relevantes, generadas por la Academia Boliviana de Historia Militar, este módulo busca ofrecer a los usuarios una sección de contenido dinámico que fomente el interés por la historia a través de publicaciones periódicas.

Asimismo, se comprobó que las publicaciones aparezcan correctamente ordenadas por las categorías y los usuarios asignados que publicaron dicha publicación, para que los visitantes al sitio web puedan navegar y acceder al contenido sin dificultades, como resultado de esta fase, el módulo de blogs fue validado satisfactoriamente, cumpliendo con los requerimientos funcionales y demostrando estabilidad en su desempeño individual dentro del sistema.

Entre los aspectos evaluados se encuentran la correcta carga de imágenes, el registro de información asociada nombre, periodo presidencial, breve descripción, y la correcta visualización de los datos en la interfaz del sistema, como podemos ver a continuación:

Figura 67:

Creación del registro de blog

The screenshot displays a web interface for creating a blog entry. On the left is a dark sidebar with navigation icons. The main area is divided into two columns. The left column contains a form with the following fields: 'Nombre' (filled with 'Juan Basilio Catacora Heredia (1760 - 1810)'), 'Categoría' (set to 'Arte'), and 'Imagen' (with a 'Browse' button and a selected portrait of a man). The right column shows a preview of the blog entry. It includes a 'Resumen' (summary) section with a short biography, followed by a 'Contenido' (content) section with a rich text editor toolbar and a full-length historical text about the subject's life and role in the independence movement.

Nota: En la figura se muestra cómo se crea y registra un blog de la Academia. Fuente: Elaboración Propia.

Figura 68:

Tabla de visualización de registro de blogs.



Nota: En la figura se muestra cómo se registra y se guarda el blog creado para visualizarse en la tabla administrativa. Fuente: Elaboración Propia.

Figura 69:

Primera vista del sitio web, mostrando



Nota: En la figura se muestra cómo se visualiza en el sitio web la creación de presidente, presentándonos la fecha de inicio y el nombre. Fuente: Elaboración Propia.

Figura 70:

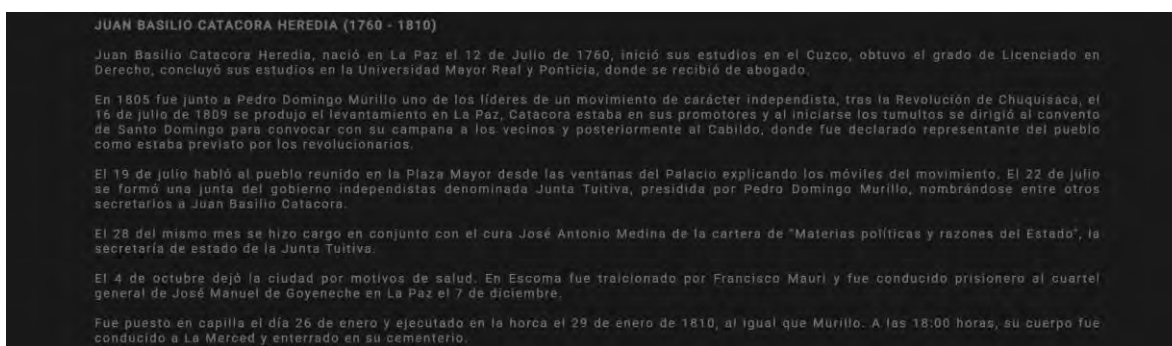
Presentación de la fotografía y título del blog seleccionado



Nota: En la figura se muestra la foto del blog, el nombre del usuario y la categoría. Fuente: Elaboración Propia.

Figura 71:

Vista de la información completa del blog seleccionado



Nota: En la figura se observa la información completa del blog seleccionado. Fuente: Elaboración Propia.

3.4.5.3. Prueba unitaria para el módulo de eventos

El módulo de eventos fue diseñado con el propósito de gestionar las actividades y exposiciones organizadas por la Academia Boliviana de Historia Militar, en este módulo permite registrar, modificar, eliminar y visualizar eventos relevantes, brindando a los usuarios una visión clara y cronológica de las actividades desarrolladas en el marco de la promoción y difusión de la historia militar.

Entre los aspectos evaluados se encuentran la correcta carga de imágenes, el registro de información asociada nombre, periodo presidencial, breve descripción, y la correcta visualización de los datos en la interfaz del sistema, como podemos ver a continuación:

Figura 72:

Creación del registro de eventos

Nota: En la figura se muestra cómo se crea y registra un evento de la Academia. Fuente: Elaboración Propia.

Figura 73:

Tabla de visualización de registro de eventos.

Id	Nombre	Usuario	Options
1	Evento de Exposición de la Historia de la Aviación Militar en Bolivia	José	[Edit] [Delete]

Nota: En la figura se muestra cómo se registra y se guarda el evento creado para visualizarse en la tabla administrativa. Fuente: Elaboración Propia.

Figura 74:

Vista desde el sitio web, una tarjeta de presentación



Nota: En la figura se muestra la creación de evento, presentándonos título general y resumen de los eventos presentados y a continuación una imagen donde se ve el ultimo evento. Fuente: Elaboración Propia.

Figura 75:

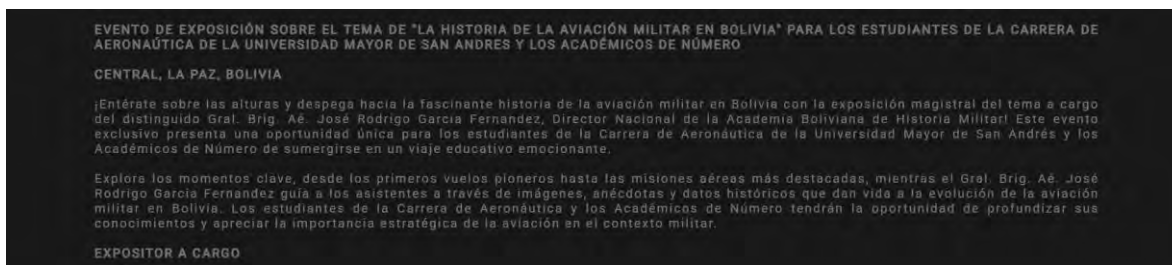
Vista de la fotografía y el título del último evento



Nota: En la figura se muestra como primer elemento la foto del evento realizado, el título general y el usuario encargado de haber creado dicha publicación. Fuente: Elaboración Propia.

Figura 76:

Vista de la información completa del evento realizado.



Nota: En la figura se observa la información completa del evento realizado. Fuente: Elaboración Propia.

Figura 77:

Fotografía del expositor a cargo del evento realizado



Nota: En la figura se observa la información del encargado que realizo el evento, mostrando una foto y nombre del expositor. Fuente: Elaboración Propia.

3.4.5.4. Conclusión de las pruebas unitarias

Tras la ejecución de las pruebas unitarias correspondientes a los módulos de colección de presidentes, blogs y eventos, se comprobó que cada componente del sistema funciona de manera correcta e independiente, cumpliendo con los requerimientos funcionales definidos. Las operaciones de inserción, modificación, eliminación y visualización de datos se realizaron sin errores, estas pruebas permiten asegurar que el sistema cuenta con una base sólida y confiable para su integración final, garantizando un desempeño estable y una correcta interacción con la base de datos.

3.5. EVALUACIÓN TÉCNICA

3.5.1. Evaluación técnica

La evaluación técnica del presente proyecto se fundamenta en el análisis de la viabilidad de los recursos tecnológicos, metodológicos y humanos necesarios para el desarrollo e implementación del Sistema de Gestión de Contenido para la Academia Boliviana de Historia Militar, bajo un enfoque de Técnicas de Enmascaramiento de Datos, haciendo cálculos referentes a la funcionalidad, usabilidad, mantenibilidad y portabilidad del sistema.

3.5.1.1. Funcionalidad

La funcionalidad del Sistema se evalúa por medio de la complejidad, no puede ser medida directamente, se lo realiza midiendo el Punto Función, que permite un resultado medible y cuantificable, para ello se toma las siguientes características:

- a. **Número de entradas de usuario:** Representan las interacciones que los usuarios tienen con el sistema, proporcionando datos que inician un proceso dentro de la aplicación, las entradas de usuario originan una operación dentro del sistema, lo que permiten que el sistema reciba, almacene y procese datos esenciales para su funcionamiento.

Tabla 55:

Tabla de las entradas de usuario

Entrada de usuario	Descripción	Complejidad	Estimado
Inicio de sesión	Autenticación con validación y sesión	Media	4
Registro de usuario	Formulario con validaciones múltiples	Media	4
Subida de imagen	Asignación colección + almacenamiento	Alta	6
Crear publicación (blog)	Formulario con imagen, texto y fecha	Alta	6
Crear evento	Formulario con validación de fechas	Media	4
Crear noticia	Formulario con entradas multimedia	Media	4
Crear visita	Formulario con validación de fechas	Media	4
Crear colección	Formulario con imagen, texto y fecha	Media	4

Nota: En la tabla se muestra las entradas de usuario, describiendo la complejidad. Fuente: Elaboración Propia.

- b. **Número de salidas de usuario:** Representan la información que el sistema entrega al usuario como resultado de procesos internos, estas salidas pueden tomar distintas formas, como pantallas, mensajes de confirmación, errores, reportes, listados o

cualquier otra forma de presentación de datos procesados, son importantes ya que reflejan el grado de interacción y retroalimentación que el sistema proporciona al usuario, afectando directamente la experiencia de uso y la funcionalidad percibida.

Tabla 56:

Tabla de las salidas de usuario

Salidas de usuario	Descripción	Complejidad	Estimado
Listado de usuarios	Muestra datos con texto	Media	5
Listado de blogs	Muestra datos con texto, imagen y fecha	Media	5
Listado de eventos	Muestra datos con texto, imagen y fecha	Media	5
Listado de noticias	Muestra datos con texto, imagen y fecha	Media	5
Listado de visitas	Muestra datos con texto, imagen y fecha	Media	5
Listado de colección	Muestra datos con texto, imagen	Media	5
Mensajes de error	Acción inmediata de parte del usuario	Baja	4
Mensajes de confirmar	Acción inmediata de parte del usuario	Baja	4
Resultado de búsqueda	Filtro dinámico por criterios	Alta	7

Nota: En la tabla se muestra las salidas de usuario, describiendo la complejidad. Fuente: Elaboración Propia.

- c. Número de peticiones del usuario:** Representan aquellas operaciones que consisten en una entrada del usuario que, al procesarse, genera una salida inmediata sin alterar los datos internos del sistema, es decir, son interacciones que permiten al usuario recuperar información específica de manera dinámica y puntual.

Tabla 57:

Tabla de las peticiones de usuario

Petición de usuario	Descripción	Complejidad	Estimado
Ver detalles blogs	Información de blogs sin editar	Baja	3
Ver detalles eventos	Información de eventos sin editar	Baja	3
Ver detalles noticias	Información de noticias sin editar	Baja	3
Ver detalles visitas	Información de visitas sin editar	Baja	3
Ver detalles usuarios	Información de usuarios sin editar	Baja	3
Búsqueda general	Entrada de texto con respuesta rápida	Media	4

Nota: En la tabla se muestra las peticiones de usuario, describiendo la complejidad. Fuente: Elaboración Propia.

- d. Número de archivos lógicos internos:** Representan los grupos de datos o estructuras lógicas mantenidas dentro del sistema por el propio software es decir que son las tablas o entidades que forman parte de la base de datos del sistema y que pueden ser creadas, consultadas, modificadas o eliminadas por el sistema.

Tabla 58:*Tabla de los archivos lógicos*

Archivos Lógicos	Descripción	Complejidad	Estimado
Usuarios	Información básica y roles de usuarios	Media	10
Roles y permisos	Gestión de accesos y restricciones	Media	10
Publicaciones	Documentos y entradas del sistema	Alta	15
Eventos	Información de eventos próximos	Alta	15
Noticias	Información de noticias por realizar	Alta	15
Visitas	Información de visitas programadas	Alta	15
Categorías	Clasificación de publicación de blogs	Baja	7

Nota: En la tabla se muestra los archivos lógicos, describiendo la complejidad. Fuente: Elaboración Propia.

Tabla 59:*Calcular total punto de función del sistema*

Parámetros de medida	Cuenta	Factor de peso			Total
		Baja	Media	Alta	
Números de entrada de usuario	36	3	4	6	108
Números de salida de usuario	45	4	5	7	180
Número de peticiones de usuario	19	3	4	6	76
Número de archivos lógicos	87	7	10	15	609
Número de interfaces	0	5	7	10	0
				TOTAL	973

Nota: En la tabla se muestra los parámetros de medida para el punto de función Fuente: Elaboración Propia.

El cálculo total del punto de función del sistema es de: **973**.

Tabla 60:*Valores de complejidad*

DATOS DE AJUSTE	VALOR DE AJUSTE
Sin influencia	0
Menor importancia	1
Moderado	2
Medio	3
Significativo	4
Esencial	5

Nota: En la tabla se muestra los valores de complejidad para el punto de función Fuente: Elaboración Propia.

La funcionalidad del sistema será obtenida por medio del punto función que se basa en la relación empírica de medidas cuantitativas del dominio de información del software. En la siguiente tabla se aplicó los valores de ajuste de complejidad a los factores de ajuste:

Tabla 61:
Ajustes de complejidad

Fi	Factores de ajuste	Valor
1	¿Requiere el sistema y la aplicación copias de seguridad?	0
2	¿Se requiere comunicación de datos?	1
3	¿Existe funciones de procesamiento distribuido?	2
4	¿Es critico el rendimiento?	3
5	¿Se ejecutará el sistema en entorno operativo existente y utilizado?	0
6	¿Se requiere datos de entrada?	5
7	¿Requiere la entrada de datos que las transiciones de entrada hagan sobre múltiples pantallas u operaciones?	0
8	¿Se utilizan los archivos maestros de forma interactiva?	3
9	¿Son complejas las entradas, las salidas, los archivos o las peticiones?	3
10	¿Es complejo el procesamiento interno?	5
11	¿Se ha diseñado el código para ser reutilizable?	5
12	¿Están incluidas en el diseño la conversión y la instalación?	0
13	¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?	0
14	¿Se ha diseñado la aplicación para facilitar los cambios y ser fácilmente utilizada por el usuario?	5
TOTAL		$\sum(f_i) = 32$

Nota: En la tabla se muestran los ajustes de complejidad para el punto de función Fuente: Elaboración Propia.

Para calcular el punto de función (PF), para el (Sistema Web) se utiliza la siguiente ecuación:

Ecuación 1: Ecuación punto de función

$$\text{Punto de función (PF)} = \text{Cuenta Total} * (0.65 + 0.01 * \text{Sum}(F_i)) \quad (1)$$

$$PF = 973 * (0.65 + 0.01 * 32)$$

$$PF = 943.81$$

$$PF(\text{Max}) = 973 * (0.65 + 0.01 * 70)$$

$$PF(\text{Max}) = 1313.55$$

$$PF = (PF/PF(\text{Max})) * 100$$

$$PF = (943.81/1313.55) * 100$$

$$PF = 71.85\%$$

La funcionalidad del Sistema representa el **71.85%**, teniendo en cuenta el punto función máximo. Lo que nos indica que el sistema cumple con los requisitos funcionales de forma satisfactoria.

3.5.1.2. Usabilidad

Para su evaluación, se ha aplicado un cuestionario estructurado compuesto por seis preguntas, cada una con opciones de valoración que permiten calificar distintos aspectos de la experiencia del usuario. La puntuación obtenida corresponde al promedio de las respuestas proporcionadas por cuatro usuarios, considerando los valores máximos y mínimos asignados a cada ítem evaluado. Esta metodología permite obtener una medición objetiva de la usabilidad percibida del sistema.

Tabla 62:

Escala de ajuste de usabilidad

DESCRIPCIÓN	ESCALA
Pésimo	1
Malo	2
Regular	3
Bueno	4
Muy Bueno	5

Nota: En la tabla se muestra la escala de ajustes de usabilidad. Fuente: Elaboración Propia.

Tabla 63:

Cuestionario de usabilidad

FACTOR	VALOR
¿Es entendible es sistema?	5
¿Puede utilizarlo fácilmente?	5
¿Es sencillo acceder a los datos?	4
¿El sistema responde rápido a sus solicitudes?	4
¿El sistema le proporciono las respuestas requeridas?	3
¿Está de acuerdo con el funcionamiento del software?	4
TOTAL	25

Nota: En la tabla se muestra un cuestionario de usabilidad. Fuente: Elaboración Propia.

Ecuación 2: Ecuación de usabilidad del software

$$Usabilidad = \frac{\left[\left(\frac{\Sigma valor}{n}\right) * 100\right]}{5} \quad (2)$$

$$Usabilidad = \frac{\left[\left(\frac{\Sigma valor}{n}\right) * 100\right]}{5}$$

$$Usabilidad = \frac{\left[\left(\frac{25}{6}\right) * 100\right]}{5}$$

$$Usabilidad = 83.30\%$$

Por lo tanto, la usabilidad del software corresponde a un **96.66%**.

3.5.1.3. Mantenibilidad

Es un conjunto de atributos relacionados con la facilidad de extender, modificar o corregir errores en un sistema software. Para hallar el índice se usa la siguiente formula:

Ecuación 3: Ecuación de índice de madurez del software

$$IMC = \frac{[Mt - (Fa + Fm + Fe)]}{Mt} \quad (3)$$

Donde:

Mt: Número de módulos de la versión actual

Fm: Número de módulos que se han modificado

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

Fe: Número de módulos que se han borrado en la versión actual

Tabla 64:
Mantenibilidad del sistema

DESCRIPCIÓN	ESCALA
Mt	12
Fm	0
Fa	0
Fe	0

Nota: En la tabla se muestra la mantenibilidad del sistema. Fuente: Elaboración Propia.

$$IMC = \frac{[12 - (0 + 0 + 0)]}{12}$$

$$IMC = 100.00\%$$

La mantenibilidad del sistema es del **100.00%**, esto quiere decir que el sistema tiene nivel de mantenibilidad buena.

3.5.1.4. Portabilidad

La portabilidad me permite llevar el sistema de un ambiente a otro, para ello se debe considerar los siguientes aspectos:

- **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.

Tabla 65:
Portabilidad del sistema

FACTOR	VALOR %
¿El sistema puede ser transferido de un entorno a otro?	90
¿Se puede adaptar con facilidad?	70
¿Es fácil de instalar?	80
¿Es capaz de reemplazar a una aplicación similar?	60
TOTAL	75%

Nota: En la tabla se muestra la portabilidad del sistema. Fuente: Elaboración Propia.

El sistema tiene la capacidad de ser ejecutado en diferentes entornos, estableciendo una portabilidad de **75.00%**.

3.5.1.5. Calidad Total

Es importante saber la calidad total del Software, para saber la satisfacción que tendrá el cliente al interactuar con el sistema, para ello sacaremos la media de todas las medidas expresadas en porcentajes:

Tabla 66:
Resultados calidad total

CARACTERISTICAS	RESULTADO DEL SISTEMA
Funcionalidad	71.85 %
Usabilidad	96.66%
Mantenibilidad	100.00%
Portabilidad	75.00%
TOTAL	72.34%

Nota: En la tabla se muestra el resultado de la calidad total del sistema. Fuente: Elaboración Propia.

Luego de aplicar los instrumentos de evaluación definidos, se determinó que la calidad total del software alcanza un **72.34%**, valor que refleja el nivel de cumplimiento de los atributos de calidad establecidos y, al mismo tiempo, representa el grado de satisfacción percibido por los usuarios finales durante su interacción con el sistema.

Cabe resaltar que este valor también permite identificar oportunidades de mejora para futuras versiones del sistema, orientadas a incrementar aún más la satisfacción del usuario y perfeccionar aspectos específicos del diseño y funcionamiento mejorando las herramientas tecnológicas al servicio de la Academia Boliviana de Historia Militar.

3.6. EVALUACIÓN ECONÓMICA

Para determinar el costo del proyecto se realiza una evaluación del presente Proyecto de Grado, donde se tomará en cuenta el costo de estudio, desarrollo del software, hardware y el precio total del sistema propuesto.

3.6.1. Costos de estudio

Los costos de estudio hacen referencia a los recursos económicos necesarios durante el desarrollo del presente proyecto de grado, estos incluyen gastos asociados al uso de internet para la búsqueda, análisis y validación de la información teórica y técnica; impresiones y copias de documentación relevante; asimismo se consideran los materiales y herramientas utilizados para el diseño, desarrollo, implementación y pruebas del sistema propuesto.

Tabla 67:

Tabla de costos de estudio

Nro.	Descripción	Cantidad	Precio Unitario Bs	Total (Bs)
1	Internet	344 Hrs.	179	61.576
2	Impresiones a color	9 unidades	35	315
3	Pasajes	8 visitas	8	64
SUB TOTAL				61.955
IMPREVISTOS (10%) DEL SUBTOTAL				6.195
TOTAL				67,771

Nota: En la tabla se muestra un análisis del costo de estudio. Fuente: Elaboración Propia.

Con los datos obtenidos se determinó que para la construcción del presente Proyecto de Grado es necesario **Bs 67.771**, tomando en cuenta un 10% para imprevistos.

3.6.2. Calculamos (KLOC)

Para ello se emplea Correlación de Código Fuente por Punto de Función (LDC/PF), donde el valor corresponde a la siguiente Tabla 70.

Tabla 68:*Factor LDC/PF de lenguajes de programación*

COSTO TOTAL DEL PROYECTO	
LENGUAJE DE PROGRAMACIÓN	FACTOR LOC / PF
JSON	9,660
PHP	8,524
SCSS	4,482
CSS	3,166
JAVASCRIPT	633
XML	56
MARKDOWN	67

Nota: En la tabla se muestra el factor LDC/PF de Lenguajes de Programación. Fuente: Elaboración Propia.

Ecuación 4: Ecuación de líneas de código

$$LOC = PF * Factor LOC/PF \quad (4)$$

Puntos de Función a miles de líneas de código (LDC) para el lenguaje de programación JSON son de 9117.20:

$$LOC = PF * Factor LOC/PF$$

$$LOC = 943.81 * 9.660$$

$$LOC = 9117.20$$

Puntos de Función a miles de líneas de código (LDC) para el lenguaje de programación PHP son de 3956451.52:

$$LOC = PF * Factor LOC/PF$$

$$LOC = 943.81 * 8,524$$

$$LOC = 3956451.52$$

Puntos de Función a miles de líneas de código (LDC) para el lenguaje de programación SCSS son de 1819665.68:

$$LOC = PF * Factor LOC/PF$$

$$LOC = 943.81 * 4,482$$

$$LOC = 1819665.68$$

Puntos de Función a miles de líneas de código (LDC) para el lenguaje de programación CSS son de 470017.38:

$$LOC = PF * Factor\ LOC / PF$$

$$LOC = 943.81 * 3,166$$

$$LOC = 470017.38$$

Puntos de Función a miles de líneas de código (LDC) para el lenguaje de programación JAVASCRIPT son de 597431.73:

$$LOC = PF * Factor\ LOC / PF$$

$$LOC = 943.81 * 633$$

$$LOC = 597431.73$$

Puntos de Función a miles de líneas de código (LDC) para el lenguaje de programación XML son de 52853.36:

$$LOC = PF * Factor\ LOC / PF$$

$$LOC = 943.81 * 56$$

$$LOC = 52853.36$$

Puntos de Función a miles de líneas de código (LDC) para el lenguaje de programación Markdown son de 63235.27:

$$LOC = PF * Factor\ LOC / PF$$

$$LOC = 943.81 * 67$$

$$LOC = 63235.27$$

Ahora sumamos las líneas de código (LOC) tanto de PHP y JavaScript.

$$LOC = PHP + JavaScript$$

$$LOC = 3956451.52 + 597431.73$$

$$LOC = 4553883.25$$

Las líneas de código en su totalidad son 4553883.25 de las cuales se estima que aproximadamente un 75% del código es reutilizable.

Ecuación 5: Ecuación de líneas de código distribuidas en el proyecto

$$KLOC = \frac{LOC}{1000} \quad (5)$$

$$KLOC = \frac{4553883.25}{1000}$$

$$KLOC = 4,553.88325$$

Por lo tanto, existen 4,553.88325 líneas de código distribuidas para el proyecto.

3.6.3. Calcular esfuerzo (PM)

Aplicando el modelo COCOMO para proyectos orgánicos:

Ecuación 6: Ecuación de esfuerzo

$$PM = 2.4 * (KLOC)^{1.05} \quad (6)$$

Donde, *PM*: Esfuerzo estimado (persona - mes), y *KLOC*: Miles de líneas de código

$$PM = 2.4 * (KLOC)^{1.05}$$

$$PM = 2.4 * (4,553.88325)^{1.05}$$

$$PM = 12,040.05 \text{ (persona - mes)}$$

3.6.4. Tiempo de desarrollo (TDEV)

Ecuación 7: Ecuación tiempo de desarrollo de software

$$TDEV = 2.5 * (PM)^{0.38} \quad (7)$$

Donde, *TDEV*: Tiempo de desarrollo, y *PM*: Esfuerzo estimado

$$TDEV = 2.5 * (PM)^{0.38}$$

$$TDEV = 2.5 * (12,040.05)^{0.38}$$

$$TDEV = 54.82 \text{ (meses)}$$

3.6.5. Calcular costo de software

Ecuación 8: Ecuación costo de software

$$\text{Costo de software} = PM * \text{Salario Mensual} \quad (8)$$

Donde, *PM*: Esfuerzo estimado

$$\text{Costo de software} = PM * \text{Salario Mensual}$$

$$\text{Costo de software} = 12,040.05 * 5,000$$

$$\text{Costo de software} = 60,200,250 \text{ bolivianos}$$

3.6.6. Calcular total del proyecto

Ecuación 9: Ecuación costo total del software

$$\text{Costo total} = \text{Costo de estudio} + \text{Software} + \text{Hardware} \quad (9)$$

$$\text{Costo total} = \text{Costo de estudio} + \text{Software} + \text{Hardware}$$

$$\text{Costo total} = 5,000 + 60,200,250 + 0$$

$$\text{Costo total} = 60,205,250$$

Tras realizar un análisis exhaustivo de todos los elementos involucrados en la ejecución del sistema propuesto, se concluye que el costo total del proyecto asciende a Bs. 60,205,250. Esta cifra contempla de manera integral los distintos rubros necesarios para el desarrollo completo de la solución, incluyendo el estudio de viabilidad, el diseño y desarrollo del software, la implementación de técnicas de enmascaramiento de datos, los recursos humanos, así como los costos asociados al hardware y software utilizados.

Además, se han considerado los gastos vinculados al tiempo de desarrollo, el esfuerzo estimado según el modelo COCOMO II, y los recursos técnicos y logísticos requeridos para llevar a cabo cada una de las etapas del proyecto, desde el análisis de requerimientos hasta la puesta en marcha del sistema. Este resultado refleja no solo la inversión necesaria para concretar una solución tecnológica robusta y segura para la Academia Boliviana de Historia Militar, sino también el valor agregado que representa contar con una plataforma moderna, funcional y orientada a la protección de la información histórica y educativa de la institución.



CAPÍTULO 4.

CONCLUSIONES Y

RECOMENDACIONES

4.1. CONCLUSIONES

Para el cumplimiento del objetivo general, se logró desarrollar un sistema de gestión de contenidos funcional y adaptado a las necesidades específicas de la Academia Boliviana de Historia Militar, incorporando técnicas de enmascaramiento que garantizan un manejo seguro de los datos sensibles.

Para el cumplimiento del primer objetivo específico, se hizo el análisis del entorno y actividades de la Academia, se identificaron claramente los requerimientos funcionales y no funcionales, lo que permitió diseñar un sistema alineado con los procesos institucionales.

Para el cumplimiento del segundo objetivo específico, se realizó la arquitectura propuesta demostrando ser adecuada para una implementación modular, escalable y mantenible, facilitando futuras ampliaciones o mejoras del sistema.

Para el cumplimiento del tercer objetivo específico, se hizo un análisis comparativo permitiéndonos seleccionar e implementar una técnica de enmascaramiento eficiente, mejorando la seguridad de los datos sin afectar el rendimiento del sistema.

Para el cumplimiento del cuarto objetivo específico, se integraron elementos visuales adecuados y una interfaz amigable, lo que favorece la experiencia del usuario, en especial para el personal administrativo y técnico de la Academia.

Para el cumplimiento del quinto objetivo específico, se hicieron las pruebas funcionales y unitarias confirmaron la estabilidad, integridad y eficacia del sistema, cumpliendo con los estándares esperados para un software institucional.

4.2. RECOMENDACIONES

Se recomienda realizar talleres de capacitación periódicos para el personal encargado del uso y mantenimiento del sistema, con el fin de aprovechar al máximo sus funcionalidades.

Dado que las amenazas de seguridad evolucionan, es aconsejable evaluar periódicamente nuevas técnicas de enmascaramiento y considerar su implementación si representan mejoras sustanciales.

En futuras versiones, se puede ampliar la funcionalidad del sistema para incluir módulos adicionales como gestión de libros digitales o eventos interactivos relacionados con la historia militar.

Desde una perspectiva académica, sería altamente beneficioso que el sistema permita la subida y gestión de contenido audiovisual, como videos educativos, conferencias históricas grabadas, entrevistas o reportajes documentales.

Se recomienda desarrollar un módulo especializado para la gestión de libros digitales e inventario bibliográfico, que incluya funciones para clasificar, consultar, prestar y registrar libros de forma sistemática.



BIBLIOGRAFÍA

BIBLIOGRAFÍA

- Academia Boliviana Historia Militar. (24 de Noviembre de 2024). *Academia Boliviana de Historia Militar*. Obtenido de Academia Boliviana de Historia Militar: <https://academiabolivianadehistoriamilitar.com>
- Amazon AWS. (2023). *¿En qué consiste el enmascaramiento de datos?: Amazon AWS*, . Obtenido de Amazon AWS, : <https://aws.amazon.com/es/what-is/data-masking/>
- Amezcu, M. (2003). *La entrevista en grupo. Características, tipos y utilidades en investigación cualitativa*. Granada: University of Granada. doi:<https://www.sciencedirect.com/science/article/abs/pii/S1130862103737917?via%3Dihub>
- Arana Flores, V. (10 de Marzo de 2023). *Introducción a las migraciones en Laravel: Guía para principiantes: Coders Free*. Obtenido de Coders Free: <https://codersfree.com/posts/introduccion-a-las-migraciones-en-laravel-guia-para-principiantes>
- Beck, K., & Cunningham, W. (1989). *A laboratory for teaching object-oriented thinking*. Luisiana, Estados Unidos: Association for Computing Machinery (ACM). doi:<https://dl.acm.org/doi/abs/10.1145/74878.74879>
- Belcic, I., & Stryker, C. (27 de Junio de 2024). *¿Qué es el modelado de procesos?: IBM*. Obtenido de IBM: https://www.ibm.com/es-es/think/topics/process-modeling?utm_source=
- Booch, G., Rumbaugh, J., & Jacobson, I. (1998). *Unified Modeling Language User Guide*,. Boston, Estados Unidos: Addison Wesley. Obtenido de https://d1wqtxts1xzle7.cloudfront.net/33850665/_Grady_Booch__James_Rumbaugh__Ivar_Jacobson__The_U%28BookFi.org%29-libre.pdf?1401701975=&response-content-disposition=inline%3B+filename%3DThe_Unified_Modeling_Language_User_Guide.pdf&Expires=1744783113&Signa
- Bridges, J. (28 de Marzo de 2025). *¿Qué es el Estudio de Viabilidad de un Proyecto?: Project Manager*. Obtenido de Project Manager: <https://www.projectmanager.com/es/estudio-viabilidad-de-un-proyecto>
- Cardona, K. (17 de Julio de 2024). *¿Qué es MySQL?: Appix*. Obtenido de Appix: <https://appix.mx/que-es-mysql/>
- Cohn, M. (2004). *User Stories Applied For Agile Software Development*. Massachusetts, Estados Unidos: Addison-Wesley Professional. Obtenido de <https://books.google.es/books?hl=es&lr=&id=SvlwuX4SVigC&oi=fnd&pg=PR13&dq>

=Cohn,+M.+(2004).+User+stories+applied:+For+agile+software+development.+Addison-

Wesley+Professional.&ots=VsVce9uUSM&sig=n69uwn8tzG6xWokCddUZLLk_Vus#v=onepage&q&f=false

Coronel Montoya, D. M. (2018). *ENMASCARAMIENTO EN LA BASE DE DATOS ORACLE PARA RESGUARDAR LA INFORMACIÓN FINANCIERA Y PERSONAL EN LA COOPERATIVA DE AHORRO Y CRÉDITO DE LA PEQUEÑA EMPRESA DE COTOPAXI*. Trabajo Dirigido, Ambato. Obtenido de https://repositorio.uta.edu.ec/bitstream/123456789/28313/1/Tesis_t1438si.pdf

Coronel, C., & Morris, S. (2019). *Database Systems: Design, Implementation, & Management*. Boston, Estados Unidos: Cengage Learning. Obtenido de <https://thuviensho.hoasen.edu.vn/handle/123456789/9321>

Corrales, J. A. (02 de Agosto de 2019). *Interfaz de usuario o UI: ¿qué es y cuáles son sus características?:* rockcontent. Obtenido de rockcontent blog: <https://rockcontent.com/es/blog/interfaz-de-usuario/>

Cosmin, C. (17 de Enero de 2025). *Una Guía Completa sobre la Autenticación en Laravel: Kinsta*. Obtenido de Kinsta: <https://kinsta.com/es/blog/autenticacion-laravel/>

Crispin, L., & Gregory, J. (2009). *Agile Testing: A Practical Guide for Testers and Agile Teams*. Nueva Jersey, Estados Unidos: Addison-Wesley Professional. doi:<https://dl.acm.org/doi/abs/10.5555/1529932>

Duckett, J. (2011). *HTML and CSS: Design and Build Websites*. Indianapolis, Canada: Wiley Publishing Inc. Obtenido de [https://books.google.es/books?hl=es&lr=&id=vOZLwYJHymQC&oi=fnd&pg=PA308&dq=Duckett,+J.+\(2011\).+HTML+and+CSS:+Design+and+Build+Websites.+Wiley.&ots=WvfjJ12m_1&sig=Dt1c_e8AJTU2Uw19nARFnYmg7XQ#v=onepage&q&f=false](https://books.google.es/books?hl=es&lr=&id=vOZLwYJHymQC&oi=fnd&pg=PA308&dq=Duckett,+J.+(2011).+HTML+and+CSS:+Design+and+Build+Websites.+Wiley.&ots=WvfjJ12m_1&sig=Dt1c_e8AJTU2Uw19nARFnYmg7XQ#v=onepage&q&f=false)

Egaña, M. (2022). *Diseño arquitectónico de software*. Caracas: Universidad Nacional Experimental de Caracas UNEXCA. Obtenido de https://es.slideshare.net/slideshow/tema-4-diseo-arquitectnico-de-software/251402128?utm_source=

ESG Innova, G. (1 de Febrero de 2018). *Los tres pilares de la seguridad de la información: confidencialidad, integridad y disponibilidad: ESG Innova Group*. Obtenido de ESG Innova Group: <https://www.pmg-ssi.com/2018/02/confidencialidad-integridad-y-disponibilidad/>

- Espacios Media. (18 de Septiembre de 2016). *¿Qué es un Análisis de Requerimientos?: Espacios Business Media*. Obtenido de Espacios Business Media: <https://www.espacios.media/que-es-un-analisis-de-requerimientos/>
- Falcon, W. J. (3 de Noviembre de 2022). *¿Qué es el levantamiento de información?: Universidad "Fermín Toro"*. Obtenido de Universidad "Fermín Toro": https://view.genially.com/6363f91871fe5d00181b41ff/interactive-content-tecnicas-de-levantamiento-de-informacion?utm_source=
- Flanagan, D. (2020). *JavaScript: The Definitive Guide*. Sebastopol, Estados Unidos: O'Reilly Media. Obtenido de [https://books.google.es/books?hl=es&lr=&id=4RChxt67lvwC&oi=fnd&pg=PR7&dq=Flanagan,+D.+\(2020\).+JavaScript:+The+Definitive+Guide+\(7.%C2%AA+ed.\).+O%27Reilly+Media.&ots=tkZ5HmFXt4&sig=k60KVghOzG2Z7e49z2N8jOYT5g#v=onepage&q&f=false](https://books.google.es/books?hl=es&lr=&id=4RChxt67lvwC&oi=fnd&pg=PR7&dq=Flanagan,+D.+(2020).+JavaScript:+The+Definitive+Guide+(7.%C2%AA+ed.).+O%27Reilly+Media.&ots=tkZ5HmFXt4&sig=k60KVghOzG2Z7e49z2N8jOYT5g#v=onepage&q&f=false)
- Fowler, M. (2003). *Patterns of Enterprise Application Architecture*. Boston, Estados Unidos: Addison-Wesley Professional. Obtenido de [https://books.google.es/books?hl=es&lr=&id=vqTfNFDzzdIC&oi=fnd&pg=PR7&dq=Fowler,+M.+\(2003\).+Patterns+of+Enterprise+Application+Architecture.&ots=oVDkGvQMtv&sig=bjloFde6CqOGHO-ncJPSkZ9fOZo#v=onepage&q&f=false](https://books.google.es/books?hl=es&lr=&id=vqTfNFDzzdIC&oi=fnd&pg=PR7&dq=Fowler,+M.+(2003).+Patterns+of+Enterprise+Application+Architecture.&ots=oVDkGvQMtv&sig=bjloFde6CqOGHO-ncJPSkZ9fOZo#v=onepage&q&f=false)
- Garfinkel, S. L. (2015). *De-Identification of Personal*. Estados Unidos: National Institute of Standards and Technology (NIST). doi:<https://nvlpubs.nist.gov/nistpubs/ir/2015/NIST.IR.8053.pdf>
- Hernández Mendoza, S. L., & Duana Avila, D. (2020). Técnicas e instrumentos de recolección de datos. En S. L. Hernández Mendoza, & D. Duana Avila, *Boletín Científico de las Ciencias Económico Administrativas del ICEA*, 9(17) (Vol. 9, págs. 51-53). México: Universidad Autónoma del Estado de Hidalgo. doi:<https://doi.org/10.29057/icea.v9i17.6019>
- Hernández Sampieri, R., Fernández Collado, C., & Baptista Lucio, M. d. (2014). *Metodología de la investigación 6ta Edición*. México: McGRAW-HILL / INTERAMERICANA EDITORES, S.A. DE C.V. Obtenido de <https://www.esup.edu.pe/wp-content/uploads/2020/12/2.%20Hernandez%2C%20Fernandez%20y%20Baptista-metodolog%C3%ADa%20Investigacion%20Cientifica%206ta%20ed.pdf>
- Hodder, O. (1994). *The Interpretation of Documents and Material Culture*. Estados Unido: Thousand Oaks, CA: Sage Publications. doi:<https://psycnet.apa.org/doiLanding?doi=10.1037%2Fh0059881>

- Kendall, K., & Kendall, J. (2011). *ANÁLISIS Y DISEÑO DE SISTEMAS*. Naucalpan de Juárez, México : PEARSON EDUCACIÓN S.A. Obtenido de http://cotana.informatica.edu.bo/downloads/Id-Analisis%20y%20Diseno%20de%20Sistemas_Kendall-8va.pdf
- Kuuse, M. (22 de Marzo de 2024). *¿Qué es la mejora de procesos y cómo implementarla?: MRPeasy*. Obtenido de MRPeasy: https://www.mrpeasy.com/blog/es/mejora-de-procesos/?utm_source=
- Lara Galicia, F. P. (27 de Agosto de 2024). *Diseño de interfaces: Principios, elementos y proceso: GoDaddy*. Obtenido de GoDaddy: <https://www.godaddy.com/resources/latam/disenio/que-es-una-interfaz-web>
- Laragon. (10 de Noviembre de 2018). *¿Que es Laragón?: Laragon*. Obtenido de Laragon: <https://laragon.org/docs/>
- Laudon, K. C., & Laudon, J. P. (2021). *SISTEMAS DE INFORMACIÓN GENERAL*. Monterrey, México: Pearson Educación. Obtenido de <https://dspace.itsjapon.edu.ec/jspui/bitstream/123456789/1420/1/sistemas-de%20informaci%C3%B3n%20gerencial.pdf>
- Lenis, A. (17 de Abril de 2023). *Qué es la interfaz de usuario, qué tipos existen y ejemplos: HubSpot*. Obtenido de HubSpot: <https://blog.hubspot.es/website/interfaz-usuario>
- Letelier, P., & Penadés, M. C. (2006). *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP) (Vol. 5)*. Buenos Aires, Argentina: Técnica Administrativa. Obtenido de <https://www.cyta.com.ar/ta0502/v5n2a1.htm>
- López Mendoza, M. (18 de Septiembre de 2020). *Extreme Programming: Qué es y cómo aplicarlo: OpenWebinars*. Obtenido de OpenWebinars: <https://openwebinars.net/blog/extreme-programming-que-es-y-como-aplicarlo/>
- Lopez, S. (17 de Mayo de 2013). *¿Qué es UML?¿Qué diagramas componen UML?: Diagramas UML*. Obtenido de Diagramas UML: <https://diagramasuml.com/>
- López, S. (13 de Agosto de 2020). *Diagrama de casos de uso: Diagramas UML*. Obtenido de Diagramas UML: <https://diagramasuml.com/casos-de-uso/>
- López, S. (31 de Marzo de 2020). *Diagrama de clases: Diagramas UML*. Obtenido de Diagramas UML: <https://diagramasuml.com/diagrama-de-clases/>
- Lucidchart. (13 de Febrero de 2023). *Qué es un modelo de base de datos: Lucidchart*. Obtenido de Lucidchart: <https://www.lucidchart.com/pages/es/que-es-un-modelo-de-base-de-datos>

- Mata Solís, L. D. (04 de Febrero de 2020). *La entrevista en la investigación cualitativa: investigalia*. Obtenido de investigalia: <https://investigaliacr.com/investigacion/la-entrevista-en-la-investigacion-cualitativa/>
- Megino Barquinero, J. M. (18 de Marzo de 2013). *Tipos de relaciones en diagramas de casos de uso. UML: BlogSEAS*. Obtenido de BlogSEAS: <https://www.seas.es/blog/informatica/tipos-de-relaciones-en-diagramas-de-casos-de-uso-uml/>
- Meléndez Valladarez, S. M., Gaitan, M. E., & Pérez Reyes, N. N. (2016). *Metodología Ágil de Desarrollo de Software Programación Extrema*. Tesis, Managua. Obtenido de <https://repositorio.unan.edu.ni/1365/1/62161.pdf>
- Merkel, D. (Marzo de 2014). Docker: Lightweight Linux Containers for Consistent Development and. *Linux Journal*, pág. 239. Obtenido de <https://www.seltzer.com/margo/teaching/CS508.19/papers/merkel14.pdf>
- Meyer, E. (2011). *CSS: The Definitive Guide*. Sebastopol, Estados Unidos: O'Reilly Media Inc. Obtenido de [https://books.google.es/books?hl=es&lr=&id=rdtCRLXAL78C&oi=fnd&pg=PR7&dq=Meyer,+E.+A.+\(2011\).+CSS:+The+Definitive+Guide.+O%27Reilly+Media.&ots=mXVSohXb1x&sig=wDKXLj73pAfZB8j9ViW_KeXXKJl#v=onepage&q&f=false](https://books.google.es/books?hl=es&lr=&id=rdtCRLXAL78C&oi=fnd&pg=PR7&dq=Meyer,+E.+A.+(2011).+CSS:+The+Definitive+Guide.+O%27Reilly+Media.&ots=mXVSohXb1x&sig=wDKXLj73pAfZB8j9ViW_KeXXKJl#v=onepage&q&f=false)
- Mojena Marín, J., & Melo Pérez, O. V. (2014). *HERRAMIENTA PARA EL ENMASCARADO DE DATOS EN BASES DE DATOS POSTGRESQL, MYSQL Y SQL SERVER*. Trabajo de Diploma. Obtenido de https://repositorio.uci.cu/bitstream/ident/9043/2/TD_07739_14.pdf
- Moreno Merchán, J. H., & Lucero Gonzáles, L. J. (2015). *IMPLEMENTACIÓN DE UNA TIENDA VIRTUAL UTILIZANDO EL CMS JOOMLA Y VIRTUEMART PARA LA OFERTA DE CURSOS Y SEMINARIOS DE CAPACITACIÓN EN LA FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN DE LA ESPOL*. Proyecto de Grado, Guayaquil. Obtenido de <https://www.dspace.espol.edu.ec/xmlui/bitstream/handle/123456789/30287/D-84579.pdf?sequence=1&isAllowed=y>
- Negocios Atalaya, I. (19 de Julio de 2023). *Cómo identificar procesos clave en tu empresa y por qué es importante: Atalaya*. Obtenido de Atalaya: https://www.atalaya.info/post/c%C3%B3mo-identificar-procesos-clave-en-tu-empresa-y-por-qu%C3%A9-es-importante?utm_source=

- Oluseye, J. (12 de Agosto de 2024). *Introducción a los disparadores SQL: Guía para desarrolladores*: Datacamp. Obtenido de Datacamp: <https://www.datacamp.com/es/tutorial/sql-triggers>
- Paradigm, V. (09 de Febrero de 2022). *Casos De Uso «Incluir» Y «Extender»*: Visual Paradigm Blog. Obtenido de Visual Paradigm Blog: <https://blog.visual-paradigm.com/es/include-and-extend-use-cases/>
- Pressman, R. S. (2021). *Software Engineering: A Practitioner's Approach*. New York, Estados Unidos: McGraw-Hill Education. Obtenido de [https://books.google.es/books?hl=es&lr=&id=bL7QZHtWvaUC&oi=fnd&pg=PR28&dq=Pressman,+R.+S.,+%26+Maxim,+B.+R.+\(2021\).+Software+Engineering:+A+Practitioner%27s+Approach+\(9.%C2%AA+ed.\).+McGraw-Hill+Education.&ots=O9xbcSqKbl&sig=6AXF_9WKCIS8xWli5vamC9SR8WI#v=o](https://books.google.es/books?hl=es&lr=&id=bL7QZHtWvaUC&oi=fnd&pg=PR28&dq=Pressman,+R.+S.,+%26+Maxim,+B.+R.+(2021).+Software+Engineering:+A+Practitioner%27s+Approach+(9.%C2%AA+ed.).+McGraw-Hill+Education.&ots=O9xbcSqKbl&sig=6AXF_9WKCIS8xWli5vamC9SR8WI#v=o)
- Robledano, A. (24 de Septiembre de 2019). *Qué es MySQL Características y ventajas*: Open Webinars. Obtenido de Open Webinars: <https://openwebinars.net/blog/que-es-mysql/>
- Rodríguez Mira, A. (08 de Febrero de 2025). *Características para el diseño de interfaces de usuario (UI)*: Tokio School. Obtenido de Tokio School: <https://www.tokioschool.com/noticias/caracteristicas-ideales-que-deben-tener-interfaces-usuario/>
- Sáenz Landazábal, A. J. (2015). *TÉCNICAS DE TRANSPARENCIA Y ENCRIPCIÓN DE INFORMACIÓN*. Proyecto de Grado, Bogotá. Obtenido de <https://repository.unilibre.edu.co/bitstream/handle/10901/8911/tesis%20andres%20saenz.pdf?sequence=1>
- Segovia Martínez, D. C. (2017). *OPTIMIZACIÓN DEL DISEÑO DEL SISTEMA DE GESTIÓN DE CONTENIDOS (CMS) PARA MEJORAR EL POSICIONAMIENTO WEB DE LA UNIDAD DE POSGRADO DE LA FACULTAD DE INGENIERÍA MECÁNICA DE LA UNIVERSIDAD NACIONAL DE INGENIERÍA*. Proyecto de Grado, Lima. Obtenido de https://repositorio.utp.edu.pe/bitstream/handle/20.500.12867/1485/Diana%20Segovia_Trabajo%20de%20Suficiencia%20Profesional_T%C3%ADulo%20Profesional_%202017.pdf?sequence=1&isAllowed=y
- Senn, J. (1992). *Análisis y diseño de sistemas de información*. México D.F., México: McGraw Hill. Obtenido de <https://www.sidalc.net/search/Record/UnerFceco:8439/Holdings#description>

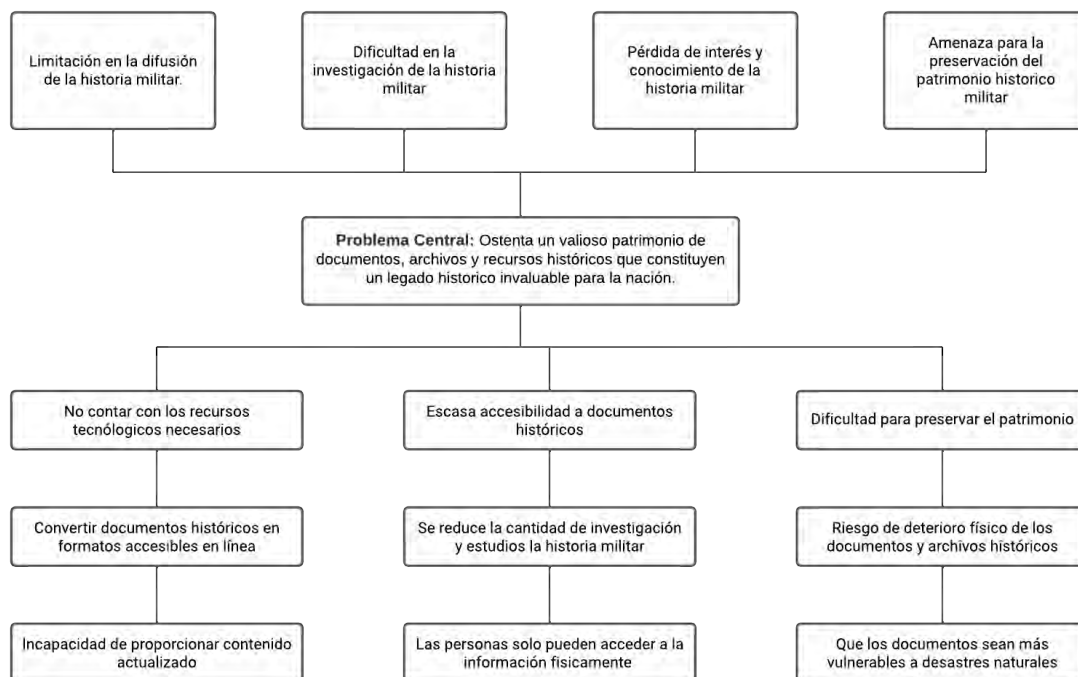
- Sommerville, I. (2005). *Ingeniería del software*. Madrid, España: PEARSON EDUCACIÓN S.A. Obtenido de <https://ulagos.files.wordpress.com/2010/07/ian-sommerville-ingenieria-de-software-7-ed.pdf>
- Sommerville, I. (2016). *Software Engineering*. Inglaterra, Reino Unido: IEEE Software. doi:<https://doi.org/10.1109/MS.2016.43>
- Taylor, O. (2015). *Laravel: Up & Running: A Framework for Building Modern PHP Apps*. California, Estados Unidos: O'Reilly Media. Obtenido de [https://books.google.es/books?hl=es&lr=&id=GcqPDwAAQBAJ&oi=fnd&pg=PT29&dq=Taylor,+T.+\(2015\).+Laravel:+Up+%26+Running:+A+Framework+for+Building+Modern+PHP+Apps.&ots=UtQSEdkx1b&sig=dOA8EQzQbksEHyGrKMBOlzfc7LI#v=onepage&q&f=false](https://books.google.es/books?hl=es&lr=&id=GcqPDwAAQBAJ&oi=fnd&pg=PT29&dq=Taylor,+T.+(2015).+Laravel:+Up+%26+Running:+A+Framework+for+Building+Modern+PHP+Apps.&ots=UtQSEdkx1b&sig=dOA8EQzQbksEHyGrKMBOlzfc7LI#v=onepage&q&f=false)
- The Bridge. (31 de Octubre de 2024). *Principios de diseño UI: The Bridge*. Obtenido de The Bridge: <https://thebridge.tech/blog/principios-basicos-del-ui-design/>
- Uwe Flick. (2007). *El diseño de Investigación cualitativa*. Madrid: Ediciones Morata, S.L. Obtenido de https://edmorata.es/wp-content/uploads/2020/06/Flick.Disen%CC%83oInvestigacionCualitativa.PR_.pdf
- Visure. (03 de Agosto de 2023). Obtenido de Qué es una Validación de Requisitos: Definición, Proceso y Herramientas: Visure: <https://visuresolutions.com/es/blog/validacion-requisitos/>
- Welling, L., & Thomsom, L. (2009). *PHP and MySQL Web Development*. Indianapolis, Canada: Addison-Wesley. Obtenido de [https://books.google.es/books?hl=es&lr=&id=G4dTRyvpfhoC&oi=fnd&pg=PA1&dq=Welling,+L.,+%26+Thomson,+L.+\(2009\).+PHP+and+MySQL+Web+Development+\(4.%C2%AA+ed.\).+Addison-Wesley.&ots=TIHgrE9fma&sig=WgUErfhec7zAX-5IIAEH8XP-mG4#v=onepage&q&f=false](https://books.google.es/books?hl=es&lr=&id=G4dTRyvpfhoC&oi=fnd&pg=PA1&dq=Welling,+L.,+%26+Thomson,+L.+(2009).+PHP+and+MySQL+Web+Development+(4.%C2%AA+ed.).+Addison-Wesley.&ots=TIHgrE9fma&sig=WgUErfhec7zAX-5IIAEH8XP-mG4#v=onepage&q&f=false)
- Yin, R. K. (2007). *Investigación sobre estudios de casos diseño y métodos*. London: International Educational and Professional Publisher. Obtenido de <https://panel.inkuba.com/sites/2/archivos/YIN%20ROBERT%20.pdf>
- Yuni, J. A., & Urbano, C. A. (2024). *Técnicas para investigar Recursos Metodológicos para la Preparación de Proyectos de Investigación* (Segunda ed., Vol. II). Córdoba, Argentina: Editorial Brujas. Obtenido de <https://abacoenred.org/wp-content/uploads/2016/01/T%c3%a9cnicas-para-investigar-2-Brujas-2014-pdf.pdf>



ANEXOS

ANEXOS

ANEXOS A:
Árbol de problemas



Nota: En el presente anexo se muestra el árbol de problemas sus causas y efecto que utilizaremos para nuestro proyecto. Fuente: Elaboración Propia

ANEXOS B:***Ubicación academia boliviana de historia militar***

Nota: En el presente anexo se la ubicación de la academia boliviana de historia militar que se encuentra en la ciudad de La Paz, cerca de la Plaza Murillo. Fuente: Elaboración Propia

ANEXOS C:**Tabla de objetivos específicos y sus acciones**

OBJETIVOS ESPECÍFICOS	ACCIONES
Realizar un análisis de las condiciones en las que se desarrollarán las actividades de la Academia Boliviana de Historia Militar para determinar los requerimientos funcionales y no funcionales del sistema.	Realizar una evaluación de las condiciones actuales
	Analizar requerimientos funcionales y no funcionales
	Documentar y priorizar requerimientos
Diseñar la arquitectura y estructura del sistema de gestión de contenidos, que sirva de apoyo para cumplir las actividades de la Academia Boliviana de Historia Militar.	Identificar las etapas de trabajo
	Definir roles y responsabilidades
	Diseñar la arquitectura del sistema
Implementar una técnica de enmascaramiento en base a un análisis comparativo de las existentes, para lograr un almacenamiento seguro en la base de datos.	Análisis comparativo de técnicas de enmascaramiento de datos
	Implementación de técnicas de enmascaramiento
Desarrollar e integrar las técnicas de enmascaramiento con el sistema de gestión de contenidos incorporando elementos visuales claros y colores amigables para facilitar la navegación y comprensión de los usuarios.	Desarrollar técnicas de enmascaramiento de datos
	Diseño de elementos visuales claros
	Implementación de esquemas de color amigables
Validar el sistema de gestión de contenidos realizando pruebas unitarias y funcionales para determinar el correcto funcionamiento.	Realizar pruebas unitarias
	Realizar pruebas funcionales completas
	Registrar y documentar resultados

Nota: En el presente anexo realizamos un análisis de los objetivos específicos, y en cada una de ellas estamos mostrando las acciones que necesitamos. Fuente: Elaboración Propia

ANEXOS D:**Tabla de contenido temático**

OBJETIVOS ESPECÍFICOS	ACCIONES	FUNDAMENTACIÓN TEÓRICA
Realizar un análisis de las condiciones en las que se desarrollarán las actividades de la Academia Boliviana de Historia Militar para determinar los requerimientos funcionales y no funcionales del sistema.	Realizar una evaluación de las condiciones actuales.	• Ingeniería de Software. • Técnicas de recolección de datos. • Análisis y Diseño de sistemas.
	Analizar requerimientos funcionales y no funcionales.	• Ingeniería de Software. • Lenguaje Unificado de Modelado. • Metodología XP.
	Documentar y priorizar requerimientos.	• Metodología XP.
Diseñar la arquitectura y estructura del sistema de gestión de contenidos, que sirva de apoyo para cumplir las actividades de la Academia Boliviana de Historia Militar.	Identificar las etapas del flujo de trabajo.	• Diseño de procesos.
	Definir roles y responsabilidades.	• Diseño de procesos.
	Diseñar la arquitectura del sistema.	• Diagramas UML. • Tipos de sistemas. • Diseño de interfaces.
Implementar una técnica de enmascaramiento en base a un análisis comparativo de las existentes, para lograr un almacenamiento seguro en la base de datos.	Análisis comparativo de técnicas de enmascaramiento de datos.	• Seguridad de sistemas • Técnicas de enmascaramiento. • Seguridad de bases de datos.
	Implementación de técnicas de enmascaramiento.	• Seguridad de sistemas. • Técnicas de enmascaramiento. • Seguridad de bases de datos.

Desarrollar e integrar las técnicas de enmascaramiento con el sistema de gestión de contenidos incorporando elementos visuales claros y colores amigables para facilitar la navegación y comprensión de los usuarios.	Desarrollar técnicas de enmascaramiento de datos.	• Tipos de combinación de enmascaramiento. • Laravel Frontend - Backend • Lenguajes de programación como PHP, JS.
	Diseño de elementos visuales claros.	• Diseño centrado en el usuario.
	Implementación de esquemas de color amigables.	• Desarrollo de interfaces web dinámicas.
Validar el sistema de gestión de contenidos realizando pruebas unitarias y funcionales para determinar el correcto funcionamiento.	Realizar pruebas unitarias.	• Plan de pruebas. • Aseguramiento de calidad.
	Realizar pruebas funcionales completas.	• Plan de pruebas • Aseguramiento de calidad.
	Registrar y documentar resultados.	• Aseguramiento de calidad.

Nota: En el presente anexo realizamos un análisis de los objetivos específicos, y en cada una de ellas estamos mostrando las acciones que necesitamos y añadiendo la fundamentación teórica para cada una de las actividades. Fuente: Elaboración Propia

ANEXOS E:**Cuadro de entrevistas**

CUADRO DE ENTREVISTA DESARROLLADOR - ACADEMIA	
DESARROLLADOR (PREGUNTAS)	ACADEMIA BOLIVIANA DE HISTORIA MILITAR (RESPUESTAS)
¿Cuál es la situación actual de la academia en cuanto a visibilidad pública?	Actualmente hemos visto un crecimiento de visitas e interés por parte de estudiantes, historiadores y público en general que sugieren tener una presencia digital.
¿Qué tipo de plataforma o sistema habían usado anteriormente? ¿Qué problemas enfrentaron?	Se utilizó un sistema tipo blog, pero fue desechado debido a problemas de seguridad y errores en publicaciones, que causaron inconvenientes incluso a nivel internacional.
¿Qué contenido desean publicar a través del nuevo sitio web?	Fotografías institucionales, libros disponibles, artículos históricos, blogs breves, noticias, visitas académicas y eventos especiales.
¿Qué tipos de usuarios requiere para el acceso al sistema?	Requerimos un sistema con gestión de usuarios diferenciados, con roles y permisos específicos.
¿Qué importancia tiene la seguridad de la información publicada?	Es algo fundamental que buscamos ya que la academia no puede correr el riesgo de que su información sea manipulada o publicada erróneamente.
¿Podemos implementar un historial ordenado de publicaciones realizadas?	Sí, se espera tener tablas administrativas para gestionar y consultar fácilmente toda la información publicada.
¿Qué tipo de contenido histórico es prioritario para mostrar en el sitio web?	Las colecciones presidenciales, blogs institucionales (“30 segundos con la historia”), visitas estudiantiles, eventos académicos, y noticias de la central y filiales.
¿Qué esperan del sistema a nivel de experiencia de usuario?	Que sea accesible, intuitivo y represente adecuadamente la identidad de la Academia como institución de preservación histórica.

Nota: En el presente anexo se muestra el cuadro donde se realizó las preguntas y respuestas que se obtuvo en la entrevista con la academia boliviana de historia militar. Fuente: Elaboración Propia