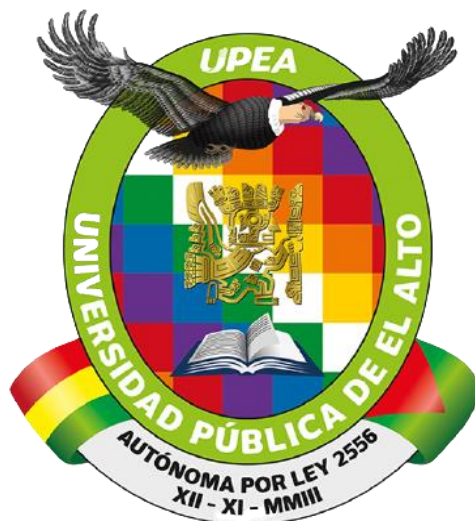


UNIVERSIDAD PÚBLICA DE EL ALTO

CARRERA INGENIERÍA DE SISTEMAS



PROYECTO DE GRADO

**“SISTEMA WEB DE GESTIÓN Y CONTROL PARA EL
SEGUIMIENTO DE HOJAS DE RUTA”**

CASO: GOBIERNO AUTÓNOMO MUNICIPAL DE HUATAJATA

Para Optar al Título de Licenciatura en Ingeniería de Sistemas

MENCIÓN: INFORMÁTICA Y COMUNICACIONES

Postulante: Graice Callizaya Chambi
Tutor Metodológico: M. Sc. Lic. Ing. Marisol Arguedas Balladares
Tutor Especialista: M. Sc. Lic. Ing. Ramiro Kantuta Limachi
Tutor Revisor: Lic. Ing. Edwin Mamani Viscarra

EL ALTO – BOLIVIA

2024

DECLARACIÓN JURADA DE AUTENTICIDAD Y RESPONSABILIDAD

Yo, **Graice Callizaya Chambi con C.I. 8422512 LP.** mediante la presente **declaro** de manera pública que la propuesta del **TRABAJO DE GRADO** titulada **“SISTEMA WEB DE GESTIÓN Y CONTROL PARA EL SEGUIMIENTO DE HOJAS DE RUTA” CASO: GOBIERNO AUTÓNOMO MUNICIPAL DE HUATAJATA** es original, siendo resultado de mi trabajo personal y no constituye una copia o replica de trabajos similares elaborados.

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

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

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

El Alto, 12 junio de 2024

Graice Callizaya Chambi
C.I. 8422512 LP
Graicecallizaya1234@gmail.com

Dedicatoria

Con Infinito Cariño:

Primero a Dios, quien me ayudo

en todo momento de mi vida.

A mis queridos padres, que siempre estuvieron para mí, es para ellos,

quienes siempre me impulsaron a culminar esta etapa de mi

vida a quien les debo todo.

A mis hermanos y amigos

por darme todo su apoyo en todo el transcurso del tiempo,

a creer en mí y ayudarme en momentos difíciles de la vida.

Graice Callizaya Chambi

Agradecimientos

Con profundo agradecimiento:

Al culminar el Proyecto de Grado, me permito confesarles que siento una profunda emoción y alegría, tengo la sensación de que el tiempo transcurre para alcanzar mi primer logro académico, por ello me siento aún más feliz, pero también siento que no tengo palabras para expresar mi gratitud a todas las personas que me colaboraron durante el proyecto de grado.

*A mis tutores por haberme impartido sus conocimientos, guiándome hasta la culminación de la Carrera de **Ingeniería de Sistemas**.*

¡Muchas Gracias a todos!!

Graice Callizaya Chambi

ÍNDICE DE CONTENIDOS

	Página
CAPÍTULO I	
1. MARCO REFERENCIAL	1
1.1. INTRODUCCIÓN	1
1.2. ANTECEDENTES	2
1.2.1. Antecedentes Institucionales.....	2
1.2.2. Antecedentes Internacionales	4
1.2.3. Antecedentes Nacionales.....	5
1.2.4. Antecedentes Locales	5
1.3. PLANTEAMIENTO DEL PROBLEMA	6
1.3.1. Problema General	6
1.3.2. Problemas Específicos.....	7
1.3.3. Formulación del Problema	7
1.4. OBJETIVO	8
1.4.1. Objetivo General	8
1.4.2. Objetivos Específicos	8
1.5. JUSTIFICACIÓN	8
1.5.1. Justificación Técnica	8
1.5.2. Justificación Económica	9
1.5.3. Justificación Social.....	9
1.6. METODOLOGÍA	10
1.6.1. Método de Ingeniería UWE (Uml-based Web Engineering)	10
1.7. MÉTRICAS DE CALIDAD	11
1.7.1. Control de calidad ISO/IEC 25000	11
1.7.2. Control de Seguridad ISO/IEC 27001	11

1.8.	ESTIMACIÓN DE COSTOS – COCOMO II.....	11
1.9.	PRUEBAS DE SOFTWARE	12
1.9.1.	Caja Blanca.....	13
1.9.2.	Caja Negra.....	13
1.9.3.	Pruebas de Estrés.....	14
1.10.	HERRAMIENTAS DE DESARROLLO	14
1.11.	LÍMITES Y ALCANCES	16
1.11.1.	Limites.....	16
1.11.2.	Alcances.....	17
1.12.	APORTES	17

CAPÍTULO II

2.	INTRODUCCIÓN.....	19
2.1.	DEFINICIONES Y CONCEPTOS.....	19
2.1.1.	Sistema.....	19
2.1.2.	Sistema Web.....	19
2.1.3.	Seguimiento	19
2.1.4.	Hojas de ruta.....	20
2.1.5.	Gestión.....	20
2.1.6.	Control	20
2.2.	METODOLOGÍA DE DESARROLLO DE SOFTWARE	20
2.2.1.	Metodología UWE	21
2.2.2.	Modelos de UWE	21
2.2.3.	Modelos de la Metodología UWE	22
2.2.4.	Codificación Del Sistema.	28
2.2.5.	Pruebas.....	29
2.2.6.	Instalación o Fase de Implementación	29
2.2.7.	Mantenimiento.....	29

2.3.	MÉTRICAS DE CALIDAD	29
2.3.1.	Métricas de Calidad de Software	29
2.3.2.	Generalidades ISO.....	30
2.3.3.	ISO/IEC 25000.....	30
2.4.	SEGURIDAD DE SOFTWARE ISO – 27001	43
2.5.	Pruebas de Funcionamiento	45
2.5.1.	Objetivos.....	45
2.5.2.	Tipos de prueba de Software	46
2.5.3.	Técnicas De Prueba.....	47
2.5.4.	Técnicas Estáticas	48
2.5.5.	Técnicas Dinámicas	48
2.5.6.	Técnicas De Caja Blanca.....	48
2.5.7.	Técnica de Caja Negra	49
2.5.8.	Pruebas de Estrés.....	50
2.6.	ESTIMACIÓN DE COSTO COCOMO II	50
2.6.1.	Modelo De Composición De Aplicación	51
2.6.2.	Modelo Diseño Temprano	52
2.6.3.	Modelo Post – Arquitectura	53
2.7.	Herramientas.....	55
2.7.1.	Lenguaje De Programación.....	55
2.7.2.	Base de datos	57
2.7.3.	Framework	58
2.7.4.	HTML	59
2.7.5.	CSS	60
2.7.6.	Editor de código	61
2.7.7.	Composer	61

CAPÍTULO III

3.	INTRODUCCIÓN.....	62
3.1.	METODOLOGÍA UWE – CONSTRUCCIÓN E IMPLEMENTACIÓN.....	62
3.1.1.	Modelo de Requerimientos	63
3.1.2.	Lista de Requerimientos del Sistema	64
3.1.3.	Diseño conceptual.....	72
3.1.4.	Modelo de Navegación	73
3.1.5.	Modelo de Presentación	75
3.1.6.	Modelo de Implementación	76

CAPÍTULO IV

4.	INTRODUCCIÓN.....	84
4.1.	MÉTRICAS DE CALIDAD ISO 25000	84
4.1.1.	Funcionalidad.....	84
4.1.2.	Confiabilidad	88
4.1.3.	Usabilidad	89
4.1.4.	Eficiencia.....	91
4.1.5.	Mantenibilidad.....	92
4.1.6.	Portabilidad	93
4.2.	SEGURIDAD.....	95
4.2.1.	Seguridad Lógica	95
4.2.2.	Seguridad Física	96
4.3.	PRUEBAS.....	97
4.3.1.	Pruebas de caja blanca.....	97
4.3.2.	Pruebas de Caja Negra.....	99
1.1.1.	Prueba de estrés.....	100
4.4.	ESTIMACION DE COSTOS – COCOMO II.....	101

CAPÍTULO V

5. CONCLUSIONES Y RECOMENDACIONES	104
5.1. CONCLUSIONES	104
5.2. RECOMENDACIONES	104
BIBLIOGRAFÍA	106

ÍNDICE DE TABLAS

	Página
Tabla 1. Definición de métricas de calidad y fórmulas de la adecuación funcional	33
Tabla 2. Definición de métricas y fórmulas de la eficiencia de desempeño	34
Tabla 3. Definición de métricas y fórmulas de la compatibilidad	35
Tabla 4. Definición de métricas y fórmulas de la usabilidad	36
Tabla 5. Definición de métricas fórmulas de la fiabilidad.....	37
Tabla 6. Definición de métricas y fórmulas de la seguridad	38
Tabla 7. Definición de métricas y fórmulas de la mantenibilidad	39
Tabla 8. Definición de métricas y fórmulas de la portabilidad.....	40
Tabla 9. Tabla de actores	63
Tabla 10. Obtención de requisitos.....	64
Tabla 11. Requerimientos funcionales del sistema	65
Tabla 12. Requerimientos no funcionales del sistema	65
Tabla 13. Descripción de caso de uso - Gestionar usuarios	67
Tabla 14. Descripción de caso de uso "Administración de unidades"	68
Tabla 15. Descripción de caso de uso - Administración de usuario	69
Tabla 16. Caso de uso cuenta de usuario.....	71
Tabla 17. Números de entrada de usuarios	84
Tabla 18. Número de salidas de usuarios	85
Tabla 19. Parámetros de medición de punto de fusión	85
Tabla 20. Punto de función sin ajustar	85
Tabla 21. Factores de complejidad	86
Tabla 22. Parámetros de medición de usabilidad.....	90
Tabla 23. Usabilidad del sistema	90

Tabla 24. Escala de valores de eficiencia	91
Tabla 25. Valoración de la eficiencia del sistema.....	91
Tabla 26. Valores para la mantenibilidad	93
Tabla 27. Resultados	95
Tabla 28. Caja blanca	97
Tabla 29. Caso de prueba "ingreso al sistema".....	100
Tabla 30. Coeficientes a, b, c, d, de Cocomo II.....	101
Tabla 31. Ecuaciones de Cocomo II	101
Tabla 32. Calculo de atributos FAE.....	102
Tabla 33. Resumen de costo	103

ÍNDICE DE FIGURAS

	Página
Figura 1. Organigrama GAMH	3
Figura 2. Diagramas de cosos de uso	22
Figura 3. Diagrama de actividades	23
Figura 4. Modulo contenido	24
Figura 5. Modelo navegacional.....	25
Figura 6. Modelo de presentación	26
Figura 7. Modelo de estructura de procesos.....	27
Figura 8. Modelo de flujo de procesos	28
Figura 9. Métricas de calidad.....	30
Figura 10. Familia de normas ISO/IEC 25000	31
Figura 11. Características de la calidad ISO/IEC 25010	32
Figura 12. Ventajas de la ISO/IEC 27001	45
Figura 13. Esquema de caja blanca	49
Figura 14. Esquema de caja negra.....	50
Figura 15. Diagrama de casos de uso general del sistema	66
Figura 16. Diagrama de casos de uso - Gestionar usuarios	67
Figura 17. Diagrama de cosos de uso - Administración de unidades	68
Figura 18. Diagrama de cosos de uso - Administrador de usuarios.....	69
Figura 19. Diagrama de caso de uso hojas de ruta	70
Figura 20. Diagramas de caso de uso módulo de hoja de ruta.....	70
Figura 21. Diagrama de caso de uso cuenta de usuario.....	71
Figura 22. Diseño conceptual	72
Figura 23. Modelo de navegación - Administrador.....	73

Figura 24. Modelo de navegación - Técnico	74
Figura 25. Modelo de navegación - usuario	75
Figura 26 Administración de usuarios	76
Figura 27. Tipo de contrato.....	76
Figura 28. Tipo de categoría.....	77
Figura 29. Grados académicos	77
Figura 30. Ámbito - profesional.....	78
Figura 31. Mae - Unidad	78
Figura 32. Unidades administrativas.....	79
Figura 33. secretaria municipal - Dirección.....	79
Figura 34. Horarios	80
Figura 35. Género - Estado civil	80
Figura 36. Contratos	81
Figura 37. Tipo de tramite.....	81
Figura 38. Tipos de estado	82
Figura 39. Habilitar tramite	82

INDICE DE ECUACIONES

	Página
(1). Ecuación de composición de aplicación.....	51
(2). Ecuación de tamaño del nuevo software a desarrollar	52
(3). Ecuación de modelo diseño temprano	52
(4). Ecuación del esfuerzo nominal	53
(5). Ecuación del factor exponencial de escala	53
(6). Ecuación de modelo post – arquitectura	54
(7). Ecuación de factor de complejidad.....	54
(8). Ecuación del punto función	54
(9). Ecuación de factor LDC/PF	54
(10). Ecuación número estimado de líneas de código	54
(11). Ecuación de COCOMO básico.....	55
(12). Ecuación de punto de función	86
(13). Ecuación de la sumatoria de los factores de ajuste	87
(14). Ecuación del cálculo de la funcionalidad.....	87
(15). Ecuación de la confiabilidad	88
(16). Ecuación de la usabilidad.....	89
(17). Ecuación de la eficiencia.....	91
(18). Ecuación de la mantenibilidad.....	92
(19). Ecuación de la portabilidad	94
(20). Ecuación de la complejidad ciclométrica del grafo	98

RESUMEN

El Gobierno Autónomo Municipal de Huatajata avanza con su pueblo, y quiere fomentar en conjunto con la sociedad el desarrollo humano, garantizar el acceso a oportunidades y el ejercicio de los derechos fundamentales como salud, educación y tecnología, sin embargo, la Alcaldía necesita centralizar los datos de los funcionarios, obtener información en el menor tiempo posible, para generar de manera inmediata las hojas de ruta para los distintos tramites, por lo cual se propuso el Sistema Web De Gestión Y Control Para El Seguimiento De Hojas de Ruta, el mismo puede procesar, toda la información importante del GAMH de forma confiable, segura y eficiente. Por esta razón es que se desarrolla el Sistema de Información Web de Control y Seguimiento para la administración de servicios de internet y tv cable, para ello se utilizó como gestor de base de datos Postgresql, un sistema de gestión de bases de datos relacionales, para el almacenamiento de los datos. Laravel, es un framework de desarrollo y diseño web basado en PHP, que proporciona una estructura sólida y herramientas avanzadas para la creación de aplicaciones web eficientes. La metodología que se aplicó es UWE (UML - Based Web Engineering), ya que sus fases se adaptan y cumplen con los requisitos para el desarrollo del sistema. En relación a la calidad se aplicó la norma ISO/IEC 25000, que permite realizar una evaluación exhaustiva del sistema y asegurar su cumplimiento de los estándares de calidad establecidos. El presente proyecto se desarrolló con una interfaz gráfica amigable diseñada para ofrecer una experiencia intuitiva y fácil de usar para los usuarios, mediante una navegación sencilla.

Palabras claves: Sistema Web, Tramite, Control

ABSTRAC

The Municipal Autonomous Government of Huatajata is progressing alongside its community, aiming to foster human development collaboratively with society, ensuring access to opportunities and the exercise of fundamental rights such as healthcare, education, and technology. However, the municipality needs to centralize employee data quickly to generate immediate roadmaps for various processes. To achieve this goal, the Web-based Management and Control System for Roadmap Tracking was proposed. This system processes essential GAMH information reliably, securely, and efficiently. Therefore, a Web Information System for controlling and monitoring internet and cable TV services was developed using PostgreSQL as the relational database management system for data storage. Laravel, a PHP-based web development framework, was employed to provide a robust structure and advanced tools for creating efficient web applications. The UWE (UML-Based Web Engineering) methodology was utilized, as its phases are well-suited to meet the system development requirements. In terms of quality, adherence to ISO/IEC 25000 standards was applied to conduct a comprehensive evaluation of the system and ensure compliance with established quality benchmarks. The project features a user-friendly graphical interface designed to offer an intuitive and easy-to-navigate experience for users, promoting straightforward usability.

Keywords: Web System, Procedure, Control

LISTA DE ABREVIATURAS

- (ISO) International Organization for Standardization - Organización Internacional de Normalización
- (G.A.M.H.) Gobierno Autónomo Municipal de Huatajata
- (COCOMO) Constructive Cost Model - Modelo Constructivo de Costos
- (PHP) Hypertext Preprocessor - Preprocesador de hipertexto
- (MVC) Modelo, vista, controlador
- (JS) JavaScript
- (CSS) Cascading Style Sheets - Hojas de estilo en cascada

CAPÍTULO I

MARCO

REFERENCIAL

CAPÍTULO I

1. MARCO REFERENCIAL

1.1. INTRODUCCIÓN

Actualmente la tecnología juega un papel muy importante en las empresas e instituciones, dado que esto se ha convertido en una parte fundamental para realizar muchos procesos que anteriormente se hacían de manera manual hoy en día, se encuentran soluciones innovadoras para optimizar procesos y aumentar la productividad permitiendo realizar las actividades y tareas de una manera más eficiente y eficaz, subsanando problemas y brindando un apoyo significativo a empresas e instituciones.

Las instituciones públicas como el Gobierno Autónomo Municipal de Huatajata no quedan excluidas de este avance tecnológico, ya que al sistematizar las tareas y procesos que realizan, obtienen un mejor tratamiento de la información, lo cual beneficia a los servidores públicos que trabajan en esta institución y al público en general del Municipio de Huatajata.

El Gobierno Autónomo Municipal de Huatajata es la encargada de organizar, preservar y custodiar el patrimonio documental producido por la misma, integrando a las unidades organizacionales que participan en la generación y administración de la documentación municipal.

El presente proyecto tiene como objetivo realizar un Sistema Web de Seguimiento de Tramites, en las respectivas Hojas de Ruta, mediante la cual integrara las funcionalidades como almacenar información digital de los documentos internos respecto a los solicitados y hacer seguimiento hasta su archivo.

Así, el presente proyecto de grado pretende contar con información oportuna, confiable y segura al realizar el seguimiento y control físico de los documentos, aplicando técnicas y métodos para su diseño. Para el desarrollo del Sistema Web se utilizará las herramientas de

desarrollo como ser: JavaScript, Laravel y PostgreSQL como gestor de base de datos, se aplicará la metodología UWE para el desarrollo del sistema.

1.2. ANTECEDENTES

1.2.1. Antecedentes Institucionales

Huatajata es una localidad y municipio de Bolivia, ubicado a orillas del Lago Titicaca en la provincia Omasuyos del departamento de La Paz. La localidad se encuentra a 80 km de la ciudad de El Alto. Después de la Reforma Agraria del 52, fue fundado el 22 de enero de 1958, formalizando diez años más tarde como cantón de la provincia Omasuyos mediante Ley N.º 439 del 4 de diciembre de 1968. Mediante la Ley N.º 033 de agosto de 2010, se creó el municipio de Huatajata en la Provincia de Omasuyos con su capital del mismo nombre.

Misión

Huatajata, Avanza con su pueblo, y se quiere fomentar en conjunto con la sociedad el desarrollo humano. Garantizar el acceso a oportunidades y el ejercicio de los derechos fundamentales como salud y educación, gestionara recursos para la mejora de las condiciones de vida de sus habitantes impulsando el crecimiento económico en un ámbito territorial articulado e integrado, soportado en una institucionalidad moderna efectiva y flexible. Promotor de la construcción de un municipio seguro, con espacios e infraestructuras públicas modernas e incluyente en armonía con el medio ambiente, sustentado en una gestión transparente y eficiente, responsable socialmente y con cultura de servicio de excelencia a los habitantes de nuestro municipio.

Visión

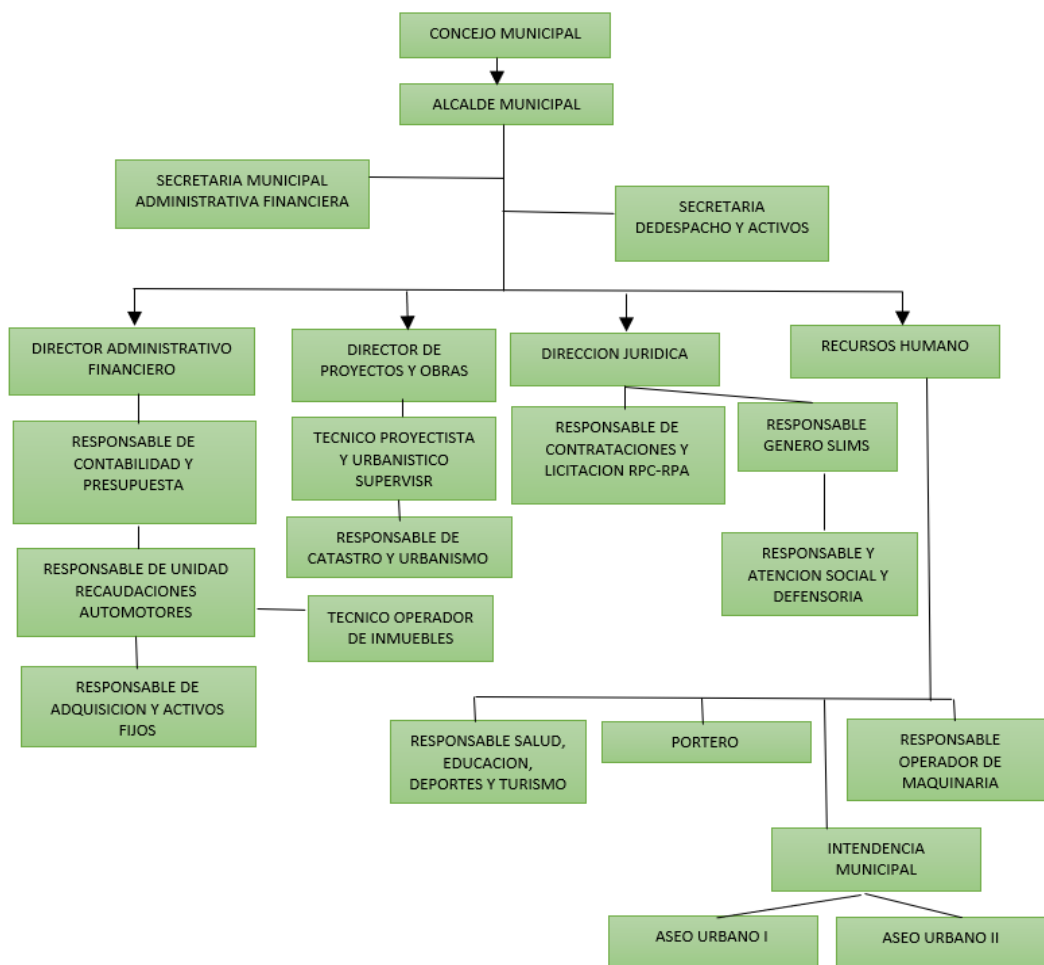
Huatajata, En el 2026 será el modelo municipal de toda la provincia Omasuyos, con el desempeño eficiente en sus funciones, con una alcaldía caracterizada por su profesionalidad, idoneidad y transparencia, motor esencial del desarrollo de su capital humano, altamente

proactivo, dinámico y comprometidos con la excelencia institucional, para brindar servicios de alta calidad a sus habitantes, con transparencia y objetividad en el manejo de los recursos, sustentado en el desarrollo de relaciones interpersonales y trabajo en equipo, que permitan contribuir con el bienestar social y económico de nuestra población.

Ser un municipio con alto nivel de desarrollo humano. Donde los sectores sociales y económicos junto al gobierno local trabajen para mejorar la calidad de vida de sus habitantes; donde los máximos valores sean equidad, justicia, democracia y solidaridad.

Figura 1.

Organigrama GAMH



Nota. Organigrama del Gobierno Autónomo Municipal de Huatajata.

1.2.2. Antecedentes Internacionales

A continuación, se describe antecedentes consultados en relación al tema del proyecto.

- (Rosas y Guevara, 2018) “Diseño De Un Sistema De Gestión Documental Y Prototipo Del Módulo De Migración Para La Preservación De Los Documentos”, presentada en la universidad Pontificia Bolivariana de Medellín, con el objetivo de diseñar la arquitectura de un sistema de información para el almacenamiento y administración de documentos digitales con un prototipo funcional que cumpla con la normativa vigente del sector publico colombiano. Quienes utilizaron las metodologías SCRUM y UML, conjuntamente con las herramientas: lenguaje de programación C# y gestor de base de datos SQLServer. (Rosas Guevara & Estrada Correa, 2018, p. 10).

- (Chavez Ligia ,2019)” Sistema De Gestión Documental Digital Para La Empresa CODSCA S.A.” Objetivo General del Proyecto: Analizar los procedimientos administrativos que ejecuta el personal de la empresa Codosca S.A. para el diseño de un sistema de gestión documental digital de acuerdo con las normas ISO 303000-30301. (Rica, 2019).

- (Castro y Félix, 2020)” Sistema de Gestión Documental para la Gestión de Titulaciones para la Vicerrectoría de Calidad e Innovación Académica”, presentado en la Universidad Europea de Madrid, con el objetivo de implementar un Sistema de Gestión Documental basado en Workflow para la Automatización de los procesos de Gestión de Titulaciones de la Vicerrectoría de Calidad e Innovación Académica de la Universidad Europea de Madrid. Quienes utilizaron la metodología SCRUM conjuntamente con las herramientas: lenguaje de programación C# y SQLServer. (Madrid, 2020)

1.2.3. Antecedentes Nacionales

- (Alarcón Cortez Ximena, 2017) “Sistema Web De Gestión Y Seguimiento De Tramites Para La Otorgación De Títulos Y Diplomas Académicos Caso: División De Sistemas De Sistemas De Información Y Estadística”. con el objetivo de sistematizar el seguimiento de trámites de titulación y la emisión de diplomas académicos, realizando un flujo de trabajo de interacción de la División de Títulos con otras unidades de la Universidad Mayor de San Andrés la metodología UWE para el modelado de sistemas Web, utilizando las fases: Modelado de Casos de Uso, Modelo Conceptual, Modelo de Navegación, y Modelo de Presentación. Para obtener la calidad de software del proyecto se basó en la metodología WEB-SITE QEM (Quality Evaluation Methodology) que es parte de un modelo de calidad que proporciona un enfoque cuantitativo y sistemático para evaluar y comparar productos Web tanto en la fase operativa como en la fase de desarrollo de ciclo de vida de un producto. El principal objetivo de Web-Site QEM es evaluar y determinar el nivel de cumplimiento de los factores de calidad descritos en el estándar ISO 9126. (Andres, repositorio institucional, 2017)

- (Chávez 2020) “Sistema de Información Vía Web para el Seguimiento y Control de Trámites Caso: Escuela Superior de Formación de Maestros Simón Bolívar”. Objetivo del Proyecto: Agilizar y optimizar los procesos que realizan los maestros o funcionarios en la solicitud de tramites, haciendo un seguimiento y control de las mismas, para el desarrollo del Sistema Web utiliza una combinación de las metodologías RUP y UWE. (Andres, repositorios latinoamericanos , 2020)

1.2.4. Antecedentes Locales

- (Mamani Yujra Ángel; 2020) “Sistema De Información Web Para El Seguimiento De Documentación Catastral Referentes A La Administración Urbana Caso: Sub Alcaldía Distrito 6 De La Ciudad De El Alto” tiene como objetivo colaborar al personal administrativo para

mejorar el tiempo y esfuerzo de los procesos de trabajo sobre los documentos mediante el uso de un sistema de información Web. Para el desarrollo de sistema se usa la metodología UWE para el análisis y diseño en base a los requerimientos recolectados de la Unidad de Administración Urbana. Para determinar la calidad el sistema, se emplea la norma ISO 25010 con todas las características necesarias y también cuenta con todas las medidas de seguridad para ello se utilizará la ISO/IEC 27002 para el buen funcionamiento y custodia de la información. Para la estimación de costos de producto se usó COSMIC basado en puntos de función. La arquitectura MVC (Movimiento Vista Controlador) ayuda al desarrollo y a mantener separado los aspectos visuales de la lógica de negocios, PHP es un lenguaje de programación y trabaja muy eficientemente con MariaDB como gestor de base de datos. (upea, 2020).

- [Ali Ticona Jhon Reynaldo, 2020] Sistema Web Control DE Registros De Archivos Notariales Para Notaria De Fe Publica N.º 015 el desarrollo del presente proyecto se utilizó la metodología UWE (UML – Base Web Engineering) ya que esta metodología se especializa en el diseño de aplicaciones Web. El sistema Web fue desarrollado en PHP, JQuery, HTML5, CSS, Bootstrap y como gestor de base de datos MySQL. La calidad fue evaluada mediante la calidad estándar ISO 9126, en la parte de Costo/Beneficio utilizamos el modelo COCOMO II. (upea, 2020)

1.3. PLANTEAMIENTO DEL PROBLEMA

1.3.1. *Problema General*

Dentro del Gobierno Autónomo Municipal de Huatajata, se observa que los métodos y procesos empleados para el seguimiento y control de las Hojas de Ruta no es el adecuado, las hojas de ruta manuales se han convertido en un obstáculo para la eficacia operativa en el Municipio de Huatajata, debido a que están propensos a errores, desde errores de escritura hasta omisiones críticas, lo que afecta la precisión de las tareas planificadas y ejecutadas, no se puede realizar el seguimiento adecuado , hay hojas de ruta que no son despachadas en el

tiempo prudente, no se sabe dónde fueron archivadas, cuales están pendientes, eso conlleva a malestar en los usuarios.

1.3.2. Problemas Específicos

A continuación, se describe algunos problemas secundarios.

- La constante manipulación de documentos en formato papel no es eficiente en su búsqueda y contribuye al deterioro general de los mismos, por lo que el seguimiento es tardío y muchas veces sin respuesta.
- Dificultad para localizar rápidamente la documentación necesaria debido a su dispersión, lo que aumenta el tiempo de respuesta.
- Pérdida de confianza por parte de los empleados municipales debido a las demoras frecuentes.
- Falta de integración y coordinación entre las unidades que manejan las hojas de ruta, lo que dificulta la sincronización de la información.
- Emisión de informes incompletos, no se sabe que personal tiene en su bandeja pendiente, para la toma de decisiones.

1.3.3. Formulación del Problema

¿Cómo coadyuvar para una eficiente gestión de documentos, optimizar el seguimiento de las hojas de ruta, y mejorar la toma de decisiones en el Gobierno Autónomo Municipal de Huatajata?

1.4. OBJETIVO

1.4.1. *Objetivo General*

Desarrollar un Sistema Web de Gestión y Control de Seguimiento de Hojas de Ruta, incrementando la eficiencia operativa para satisfacción de los usuarios y la correcta toma de decisiones en el Gobierno Autónomo Municipal de Huatajata.

1.4.2. *Objetivos Específicos*

- Realizar un análisis exhaustivo de la situación actual del proceso de seguimiento de las hojas de ruta.
- Diseñar una interfaz web amigable que permita gestionar y controlar los registros de la base de datos.
- Mejorar los procesos operativos para reducir demoras, restaurando la confianza de la comunidad y los empleados municipales.
- Concretar la coordinación entre las unidades responsables de hojas de ruta para asegurar la coherencia y actualización oportuna de la información.
- Seleccionar protocolos específicos de control de calidad de la información, enfocados en minimizar riesgos para una toma de decisiones precisa, y asegurar la disponibilidad oportuna de datos estratégicos en el ámbito municipal.
- Generar reportes estadísticos de la situación y ubicación de las hojas de ruta.

1.5. JUSTIFICACIÓN

1.5.1. *Justificación Técnica*

Para el desarrollo del presente proyecto se cuentan actualmente con el equipo y conexión a internet y las herramientas adecuadas para la implementación del sistema

propuesto, también existe los medios necesarios en el Gobierno Autónomo Municipal de Huatajata para realizar el análisis, diseño e implementación y mantenimiento del sistema para su funcionamiento en la Web.

1.5.2. Justificación Económica

El presente Perfil se justifica económicamente por el ahorro de material de escritorio, tales como hojas de papel, las cuales se usan para el registro de usuarios, ya que tales características serán implementadas en el sistema.

Por otra parte el sistema web se desarrollara utilizando software libre, el trabajo realizado de forma manual será reducido considerablemente, gracias a la implementación del presente proyecto, generando de esta forma reportes de manera inmediata que ayuden al seguimiento de cada una de las hojas de ruta, así realizar una correcta toma de decisiones, cuyo beneficio para el Gobierno Autónomo Municipal de Huatajata, De esta manera evitando pérdidas económicas, por tener que responder a denuncias por la mala praxis de la correspondencia.

1.5.3. Justificación Social

El presente proyecto se justifica socialmente porque se beneficiará a todo el municipio, debido a que ellos tendrán el acceso a toda la información y administración de todos los usuarios en tiempo real, la cual ayudara a tener la toma de decisiones de forma oportuna. Así también el municipio será beneficiado, a que podrán acceder de forma rápida y segura a los tramites que requieran, no tendrán que lidiar con más denuncias por tardar en responder a la correspondencia.

1.6. METODOLOGÍA

La metodología de desarrollo de software es el conjunto de técnicas y métodos que se utilizan para diseñar una solución de software informático. Es importante señalar que existen varias, la elección es decisión de cada equipo.

1.6.1. *Método de Ingeniería UWE (Uml-based Web Engineering)*

UWE está basado en estándares de la OMG como UML, Model Driven Architecture de OMG (MDA), Object Constraint Language (OCL) y eXtensible Markup Language (XML), asegurando su seguimiento mediante guías y especificaciones para el uso de tecnologías orientadas a objetos.

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

La notación de UWE se define como una ligera extensión de UML, proporcionando un perfil UML para el dominio específico de la web. (Danielthewolf1993, Metodología UWE, 2015)

Etapas de la metodología UWE

- Captura, análisis y especificación de requisitos
- Diseño del sistema
- Codificación del software
- Pruebas
- La Instalación o Fase de Implementación
- Mantenimiento

1.7. MÉTRICAS DE CALIDAD

1.7.1. Control de calidad ISO/IEC 25000

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

La familia ISO/IEC 25000 es el resultado de la evolución de otras normas anteriores, especialmente de las normas ISO/IEC 9126, que describe las particularidades de un modelo de calidad del producto software, e ISO/IEC 14598, que abordaba el proceso de evaluación de productos software. Esta familia de normas ISO/IEC 25000 se encuentra compuesta por cinco divisiones (NORMAS ISO 25000, 2022).

1.7.2. Control de Seguridad ISO/IEC 27001

Para el proyecto se utilizará la norma ISO 27000 para la seguridad de nuestro proyecto sistema web de gestión y control para el seguimiento de hojas de ruta.

ISO 27001 se ha convertido en la principal norma a nivel mundial para la seguridad de la información y muchas empresas han certificado su cumplimiento; aquí se puede ver la cantidad de certificados en los últimos años:

El eje central de ISO 27001 es proteger la confidencialidad, integridad y disponibilidad de la información en una empresa. Esto lo hace investigando cuáles son los potenciales problemas que podrían afectar la información (es decir, la evaluación de riesgos) y luego definiendo lo que es necesario hacer para evitar que estos problemas se produzcan (es decir, mitigación o tratamiento del riesgo). (Alvarez, 2021)

1.8. ESTIMACIÓN DE COSTOS – COCOMO II

En el proyecto se utilizar el método COCOMO II es un modelo capaz de conseguir estimaciones de precisión, permite calibrarse por organización, aunque para ello requiere la

experiencia de un número grande de proyectos ya finalizados que puedan aportar datos necesarios para la recalibración.

Este modelo permite realizar estimaciones en función del tamaño del software, y de un conjunto de factores de costo y de escala. Los factores de costo describen aspectos relacionados con la naturaleza del producto, hardware utilizado, personal involucrado, y características propias del proyecto. El conjunto de factores de escala explica las economías y des economías de escala producidas a medida que un proyecto de software incrementa su tamaño.

El modelo COCOMO II posee tres modelos denominados:

- Modelo de composición de aplicación
- Modelo de fase de diseño previo
- Modelo de fase posterior a la arquitectura

Al igual que todos los modelos de estimación de software. El modelo COCOMO II descrito antes necesita información del tamaño. Dentro de la jerarquía del modelo hay tres opciones de tamaño distintas: puntos objeto, puntos de función, y líneas de código fuente. (Gracia, 2012)

1.9. PRUEBAS DE SOFTWARE

Las pruebas de software son un conjunto de métodos que nos ayudan a garantizar que el producto que estamos creando cumpla con los estándares de calidad en cada una de las fases de su desarrollo. De esta forma nos ayuda a detectar y prevenir errores, reducción de costos y mejora en el rendimiento.

1.9.1. Caja Blanca

La prueba de caja blanca, en ocasiones llamada prueba de caja de vidrio, es una filosofía de diseño de casos de prueba que usa la estructura de control descrita como parte del diseño a nivel de componentes para derivar casos de prueba. Al usar los métodos de prueba de caja blanca, puede derivar casos de prueba que: 1) garanticen que todas las rutas independientes dentro de un módulo se revisaron al menos una vez, 2) revisen todas las decisiones lógicas en sus lados verdadero y falso, 3) ejecuten todos los bucles en sus fronteras y dentro de sus fronteras operativas y 4) revisen estructuras de datos internas para garantizar su validez (Pressman, 2010).

1.9.2. Caja Negra

Las pruebas de caja negra, también llamadas pruebas de comportamiento, se enfocan en los requerimientos funcionales del software; es decir, las técnicas de prueba de caja negra le permiten derivar conjuntos de condiciones de entrada que revisarán por completo todos los requerimientos funcionales para un programa. Las pruebas de caja negra tienden a aplicarse durante las últimas etapas de la prueba puesto que, a propósito, la prueba de caja negra no considera la estructura de control, la atención se enfoca en el dominio de la información (Pressman, 2010).

Las pruebas se diseñan para responder a las siguientes preguntas.

¿Cómo se prueba la validez funcional?

¿Cómo se prueban el comportamiento y el rendimiento del sistema?

¿Qué clases de entrada harán buenos casos de prueba?

¿El sistema es particularmente sensible a ciertos valores de entrada?

¿Cómo se aíslan las fronteras de una clase de datos?

¿Qué tasas y volumen de datos puede tolerar el sistema?

¿Qué efecto tendrán sobre la operación del sistema algunas combinaciones específicas de datos?

1.9.3. Pruebas de Estrés

Las pruebas de estrés de software, también conocidas como pruebas de resistencia, son un recurso utilizado durante la etapa de *testing*, con el objetivo de probar los límites de un sistema y prever escenarios de riesgo ante cargas extremas. Para ello los desarrolladores simulan el uso del producto informático mediante el envío de peticiones constantes, más allá del parámetro normal que podrían atender en periodos reducidos. (estres, 2022)

1.10. HERRAMIENTAS DE DESARROLLO

En la propuesta del proyecto se utilizó las siguientes herramientas para el desarrollo del sistema web:

PHP: Es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web de contenido dinámico y puede ser incrustado dentro del código HTML. Las características de PHP son velocidad, estabilidad, seguridad y simplicidad. Además, se considera uno de los lenguajes más flexibles, potentes y de alto rendimiento, (Vaswani, 2019).

PostgreSQL, comúnmente pronunciado "Post-GRES", es una base de datos de código abierto que tiene una sólida reputación por su fiabilidad, flexibilidad y soporte de estándares técnicos abiertos. A diferencia de otros RDMBS (sistemas de gestión de bases de datos relacionales), PostgreSQL (enlace externo a ibm.com) soporta tipos de datos relacionales y no relacionales. Esto la convierte en una de las bases de datos relacionales más compatibles, estables y maduras disponibles actualmente. (PostgreSQL, 2024)

Navicat: Navicat Premium es una herramienta de desarrollo de bases de datos que le permite conectarse simultáneamente a bases de datos MySQL, Redis, MariaDB, MongoDB,

SQL Server, Oracle, PostgreSQL y SQLite desde una sola aplicación. Compatible con bases de datos cloud como Amazon RDS, Amazon Aurora, Amazon Redshift, Amazon ElastiCache, Microsoft Azure, Oracle Cloud, Google Cloud, MongoDB Atlas y Redis Enterprise Cloud. Podrá crear, administrar y mantener sus bases de datos de forma rápida y fácil. (Navicat, 2024)

HTML: Es un lenguaje en el que se diseñan páginas para que se visualicen a través del navegador. Este lenguaje se basa en etiquetas y atributos. Una página HTML contiene texto con un cierto formato y referencias a archivos externos que contienen imágenes, sonidos y animaciones, (Gauchat, 2012).

CSS: CSS, al igual que HTML, es uno de los lenguajes centrales de Internet. Mientras que para añadir texto a un sitio web se utiliza HTML y se estructura semánticamente, para definir el diseño del contenido se utiliza CSS. Aunque HTML y CSS se utilizan en combinación, las instrucciones de diseño de CSS y los elementos de HTML existen por separado. Esto significa que una máquina puede leer un documento electrónico incluso sin CSS. Con la ayuda de CSS, el contenido del navegador se prepara visualmente y se presenta de forma atractiva. (S.L.U., 2021)

Framework Laravel: Laravel es un framework de código abierto para desarrollar aplicaciones y servicios web con PHP 5, PHP 7 y PHP 8. Su filosofía es desarrollar código PHP de forma elegante y simple, evitando el "código espagueti". Fue creado en 2011 y tiene una gran influencia de frameworks como Ruby on Rails, Sinatra y ASP.NET MVC.2.

Bootstrap: Bootstrap es un framework CSS de código abierto que favorece el desarrollo web de un modo más sencillo y rápido. Incluye plantillas de diseño basadas en HTML y CSS con la que es posible modificar tipografías, formularios, botones, tablas, navegaciones, menús desplegados, etc. También existe la posibilidad de utilizar extensiones de Javascript adicionales. (System B. G., 2020)

Javascript: Es un lenguaje de programación diseñado en un principio para añadir interactividad a las páginas webs y crear aplicaciones web. A pesar de la similitud en el

nombre, no está relacionado con Java. Se emplea en el desarrollo de páginas web para tareas como cambiar automáticamente la fecha de una página, hacer que una página aparezca en una ventana emergente al hacer clic en un enlace o que un texto o imagen cambien al pasar el ratón por encima. También suele emplearse para hacer encuestas y formularios. Se ejecuta en el ordenador del visitante a la web, por lo que no requiere descargas constantes desde el sitio web (Gauchat, 2012).

Composer: Composer es el gestor de dependencias para PHP. Sirve para descargar y gestionar las librerías y dependencias de un proyecto PHP de manera automatizada. Cuando se desarrolla una aplicación o sitio web en PHP, a menudo se utilizan diferentes librerías o frameworks de terceros para añadir funcionalidades adicionales. Composer permite gestionar estas dependencias de manera sencilla, descargándolas y manteniéndolas actualizadas de manera automática. (Composer, 2022)

1.11. LÍMITES Y ALCANCES

1.11.1. *Limites*

El Sistema Web tendrá las siguientes limitaciones:

- El presente proyecto se limita a optimizar el control y seguimiento de hojas de ruta de acuerdo a sus necesidades y requerimientos específicos.
- El sistema se desarrollará específicamente en el marco del proceso de las hojas de ruta por unidades.
- El sistema no contendrá la parte contable o financiera
- El sistema contara con niveles de administración, por lo que ciertos usuarios tendrán acceso a diferentes módulos

1.11.2. Alcances

El sistema contendrá los siguientes módulos:

- Modulo usuario: Donde se realizará el registro de los usuarios, la asignación de roles y sus respectivos permisos.
- Backups: Se encarga de realizar copias de seguridad de la base de datos.
- Módulo de tramites: Creación, edición y eliminación de hojas de ruta.
- Asignación de tareas y responsabilidades a usuarios específicos.
- Facilitar la comunicación y relación con los involucrados para dar respuesta a las Hojas de Ruta.
- Módulo de reportes pdf, se generará reporte de la hoja de ruta, así mismo se podrá ver el personal que tiene más cantidad de tramites en su bandeja de lo aceptable.

1.12. APORTES

El sistema propuesto para el Gobierno Autónomo Municipal de Huatajata se perfila como una contribución esencial para optimizar la gestión operativa y elevar la satisfacción de los usuarios. Al abordar la constante manipulación de documentos, se espera preservar la integridad y seguridad de la información, reduciendo así los riesgos asociados con su deterioro. La mejora en la búsqueda y recuperación de documentos dispersos no solo disminuirá los tiempos de respuesta operativa, sino que también optimizará la eficiencia en la ejecución de procesos. La coordinación entre las unidades responsables de hojas de ruta, facilitada por el sistema, promoverá la coherencia y actualización constante de la información. Este aspecto no solo mejorará la colaboración interna, sino que también redundará en una experiencia más fluida y confiable para la comunidad y los empleados municipales. La

reducción de demoras operativas, asociada con la pérdida de confianza previa, jugará un papel crucial en la restauración de la credibilidad y la percepción positiva hacia las operaciones del Gobierno Autónomo Municipal de Huatajata. El sistema también tiene como objetivo mitigar el riesgo de toma de decisiones erróneas al establecer un sistema de control sólido. Esto garantizará que la información crucial esté disponible de manera confiable y que se tomen decisiones fundamentadas en datos precisos.

CAPÍTULO II

MARCO TEÓRICO

CAPÍTULO II

2. INTRODUCCIÓN

A continuación, en el siguiente capítulo se muestra la justificación documental de los diferentes conceptos que se tiene para poder sustentar bases teóricas para el desarrollo del proyecto de grado en sus diferentes etapas.

2.1. DEFINICIONES Y CONCEPTOS

2.1.1. *Sistema*

Se entiendo por un sistema a un conjunto de datos ordenados conforme a una serie de instrucciones o algoritmos, que permiten su ubicación y recuperación rápida y simple, eso es un sistema de información o informático, concepto que también emplean otras ciencias de la información como la bibliotecología, pero que en el caso de la informática está administrado de manera automática por un computador (Etecè, 2021).

2.1.2. *Sistema Web*

Hoy en día el sistema web es muy utilizado por sus potentes funcionalidades. Si aún no tienes muy claro sobre su definición, pues en este post, te cuento en que consiste el sistema web. El sistema web o también denominado aplicaciones web se define como aplicaciones de software que se puede usar en un servicio web por medio de internet o de una intranet desde un navegador. Actualmente, el sistema web es muy utilizado por la razón de que es muy rápida y práctica en el navegador web. De hecho, las aplicaciones web evita gastos lo que significa a que no será necesario en aprender a manejar nuevos programas que puedan ser costosos y podrás trabajar en cualquier lugar donde se encuentre (System, 2022).

2.1.3. *Seguimiento*

La palabra seguimiento es la acción y efecto de seguir o seguirse, en el contexto popular suele usarse como sinónimo de persecución, observación o vigilancia. Siendo este

mismo usado principalmente en el contexto de investigaciones policiales, detectivescas, jurídicas, medicas, científicas, estadística, entre otras; para observar y analizar la evolución un determinado caso. Aunque el término puede aplicarse a cualquier investigación, proceso o proyecto con observación constante (Adrian, 2023).

2.1.4. Hojas de ruta

La expresión 'Hoja de Ruta' hace referencia a todo aquel material o soporte sobre el cual se establece un plan organizativo que sirve para ubicarse a nivel geográfico o espacial durante un viaje o travesía. La hoja de ruta también puede ser entendida como el plan de temas o problemáticas a tratar durante una reunión, una asamblea o un espacio en el cual hay varias cosas de las que ocuparse (Bembibre, 2010).

2.1.5. Gestión

La gestión es un conjunto de procedimientos y acciones que se llevan a cabo para lograr un determinado objetivo. El objetivo de la gestión es alcanzar un objetivo optimizando al máximo posible los recursos recursos disponibles (Westreicher, 2020).

2.1.6. Control

El control es el acto de supervisar y regular diversas actividades para que se ajusten a estándares o criterios preestablecidos. Implica la evaluación constante de procesos, resultados y desviaciones con el propósito de garantizar la consecución de metas y objetivos. (7Graus, 2023)

2.2. METODOLOGÍA DE DESARROLLO DE SOFTWARE

La metodología de desarrollo de software es el conjunto de técnicas y métodos que se utilizan para diseñar una solución de software informático. Es importante señalar que existen varias, de manera que es una decisión de cada equipo. Trabajar con una metodología es imprescindible por una cuestión de organización. No en vano, los factores tienen que estar

ordenados y saber cómo se van a utilizar. Por otra parte, las metodologías también sirven para controlar el desarrollo del trabajo. Esto sirve para minimizar los márgenes de errores y anticiparse a esa situación.

Otra ventaja de utilizar una metodología es que te hace ahorrar tiempo y gestionar mejor los recursos disponibles. Esto sucede tanto en metodologías a corto como a largo plazo. Cuando te decantes por un sistema, has de tener en cuenta este factor. Al final, uno de los elementos básicos es optimizar los recursos a tu alcance (Carleman, 2024).

2.2.1. Metodología UWE

El principal objetivo del enfoque UWE es proporcionar: un lenguaje de modelado específico del dominio basado en UML; una metodología dirigida por modelos; herramientas de soporte para el diseño sistemático; y herramientas de soporte para la generación semi-automática de Aplicaciones Web. La notación de UWE se define como una ligera extensión de UML, proporcionando un perfil UML para el dominio específico de la web (Danielthewolf1993, 2015).

2.2.2. Modelos de UWE

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

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

- Un Modelo de Requerimientos que captura los requerimientos del sistema.
- Un Modelo Conceptual para el contenido (modelo de contenido).
- Un Modelo de Usuario.
- UN Modelo de Navegación que comprende la estructura de la navegación.

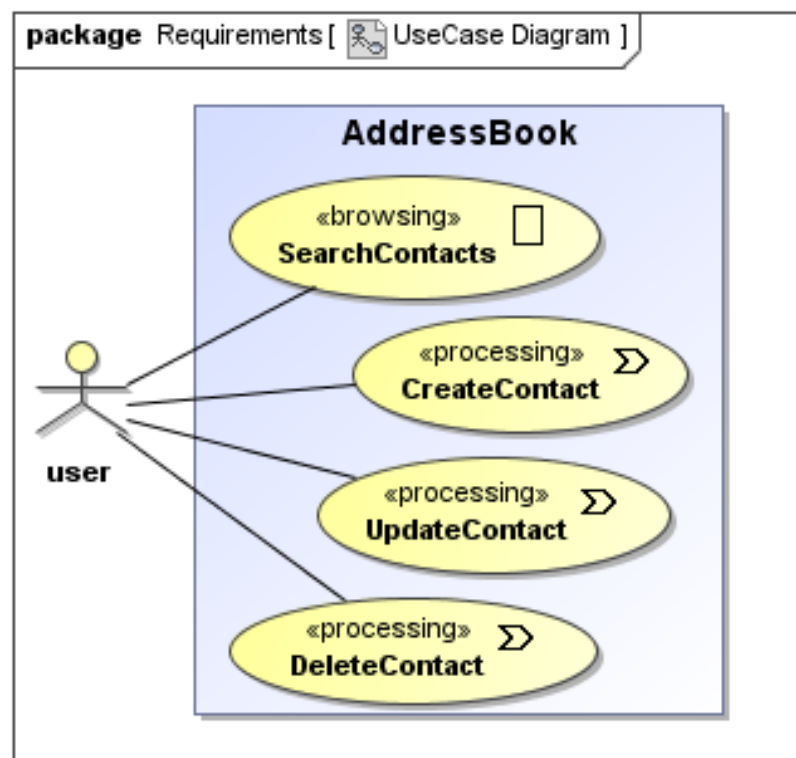
- Un Modelo de Presentación que abarca modelos estáticos y dinámicos (modelo de estructura de la presentación, modelo del flujo de la presentación, modelo de interface abstracta de usuario, y modelo de ciclo de vida del objeto).
- Un modelo de adaptación (Danielthewolf1993, 2015).

2.2.3. Modelos de la Metodología UWE

2.2.3.1. **Modelo De Requerimiento.** En UWE el modelado de requisitos consiste de dos partes:

Figura 2.

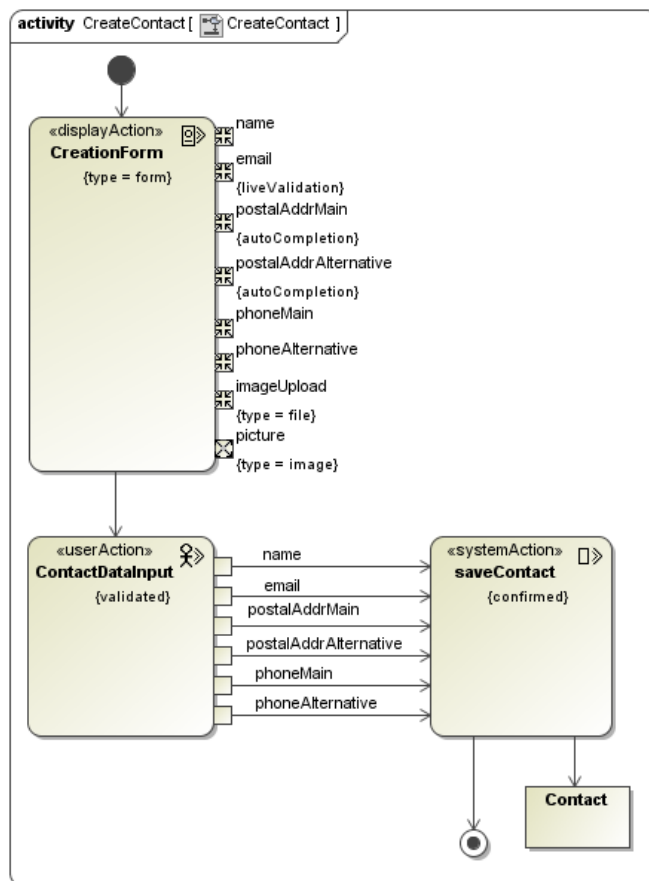
Diagramas de cosos de uso



Nota. Adaptado UWE – UML – Based Web Engineering, de (Munche,2016)

Figura 3.

Diagrama de actividades

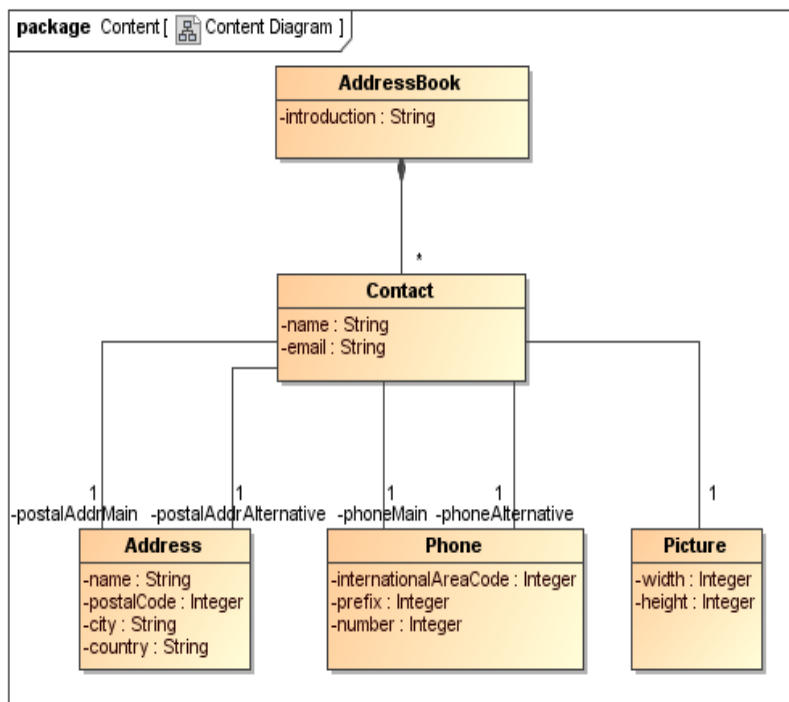


Nota. Adaptado UWE – UML – Based Web Engineering, de (Munchen,2016)

2.2.3.2. Modelo Contenido. Este es un diagrama UML normal de clases, por ello debemos pensar en las clases que son necesarias para nuestro ejemplo. Primero queremos disponer de una clase agenda ("AddressBook") conteniendo un conjunto de contactos. Cada contacto debe contener un nombre, y puede contener una dirección de correo, dos números de teléfono y dos direcciones postales. El nombre y la dirección de correo son Strings, el teléfono y la dirección postal son clases que representan más información, como se ilustra en la siguiente figura:

Figura 4.

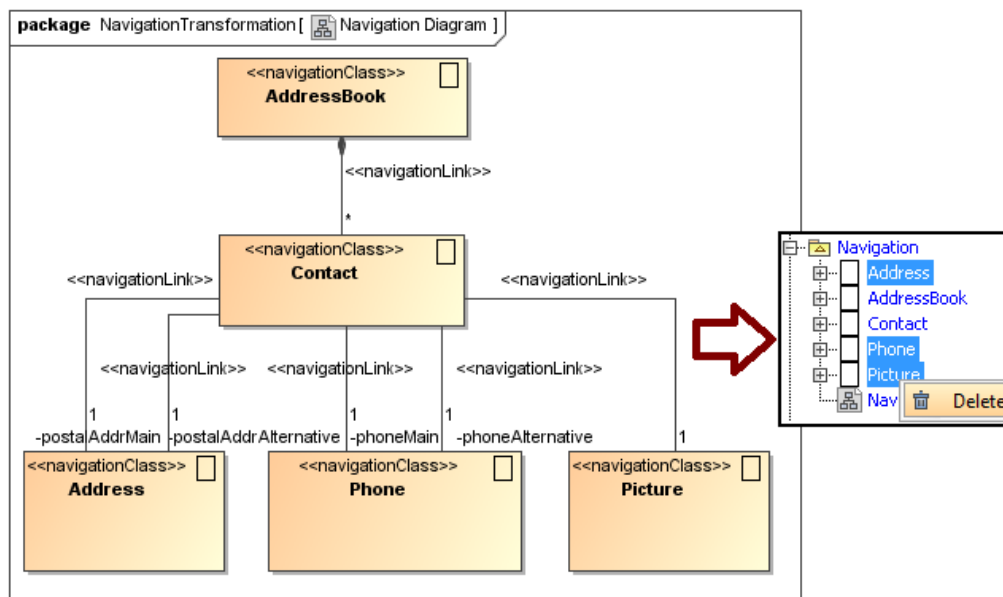
Modulo contenido



Nota. Adaptado UWE – UML – Based Web Engineering, de (Munche,2016)

2.2.3.3. Modelo de navegación. En un sistema para la web es útil saber cómo están enlazadas las páginas. Ello significa que necesitamos un diagrama conteniendo nodos (nodes) y enlaces (links). ¿Pero que es un nodo? Nodos son unidades de navegación y están conectados por medio de enlaces. Nodos pueden ser presentados en diferentes páginas o en una misma página. UWE provee diferentes estereotipos, los que presentaremos mediante nuestro ejemplo. La forma más simple de obtener un Diagrama de Navegación básico es utilizando la transformación. En este caso obtenemos para nuestro ejemplo un diagrama que contiene más nodos de los necesarios. Para los nodos y enlaces son usados los estereotipos «navigationClass» and «navigationLink»:

Figura 5.
Modelo navegacional

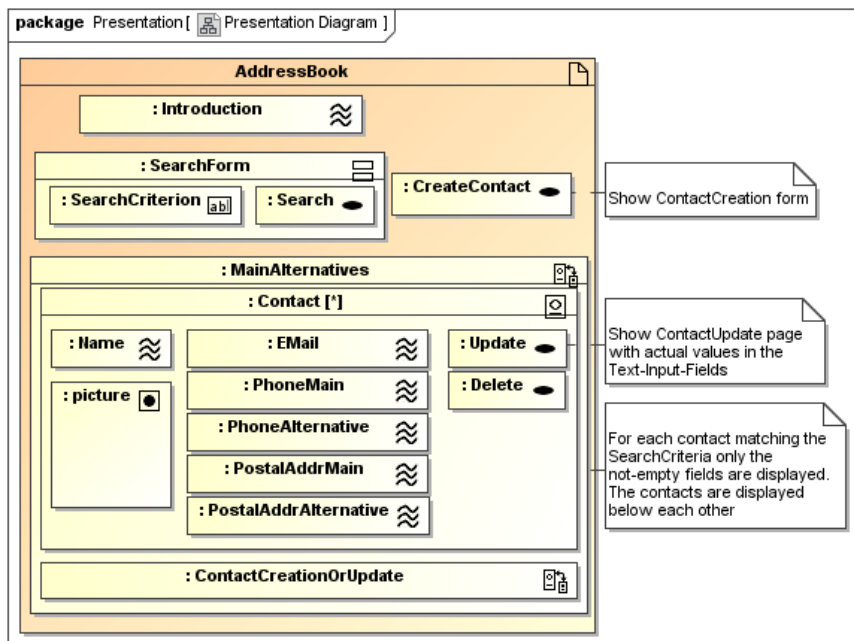


Nota. Adaptado UWE – UML – Based Web Engineering, de (Munchen,2016)

2.2.3.4. Modelo de Presentación El Modelo de Navegación no indica cuáles son las clases de navegación y de proceso que pertenecen a una página web. ¿Podemos usar un diagrama de presentación con el fin de proveer esta información! Agrega una de las propiedades con los estereotipos de UWE en ellos para expresar, que el elemento está ubicado en una página web. Las propiedades pueden anidarse, por ejemplo, cada contacto («presentationGroup»-property) cubre diferentes textos y botones. Ello significa, que para cada contacto la correspondiente dirección de correo y los correspondientes campos de teléfonos y direcciones serán visualizados en la página.

Figura 6.

Modelo de presentación

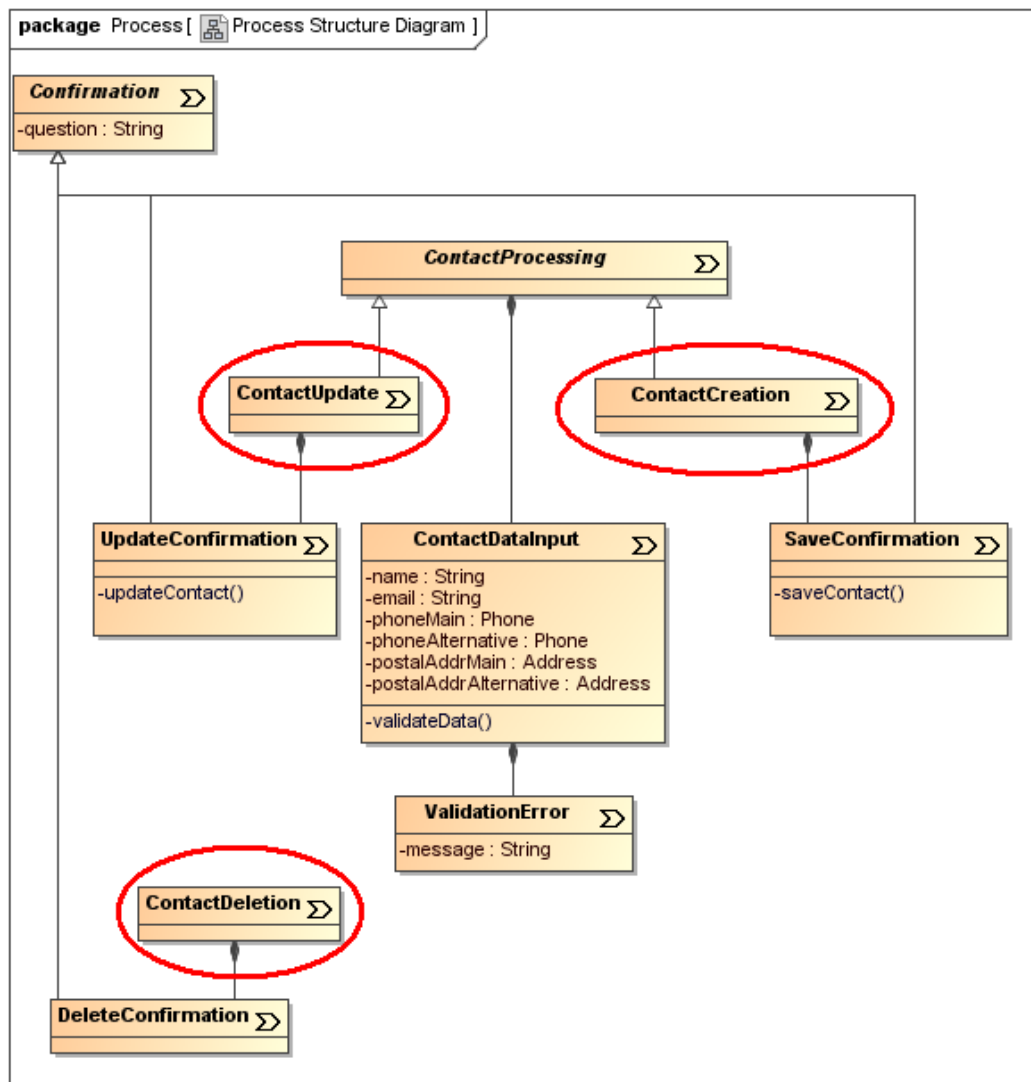


Nota. Adaptado UWE – UML – Based Web Engineering, de (Munche,2016)

2.2.3.5. Modelo de Procesos. Con el fin de describir las relaciones entre las diferentes clases de proceso, creamos un diagrama de clases, usando la transformación de navegación a estructura de procesos (Navigation to Process Structure Transformation). Después de ejecutar la transformación tenemos un diagrama de clases con tres clases enmarcadas con un borde rojo.

Figura 7.

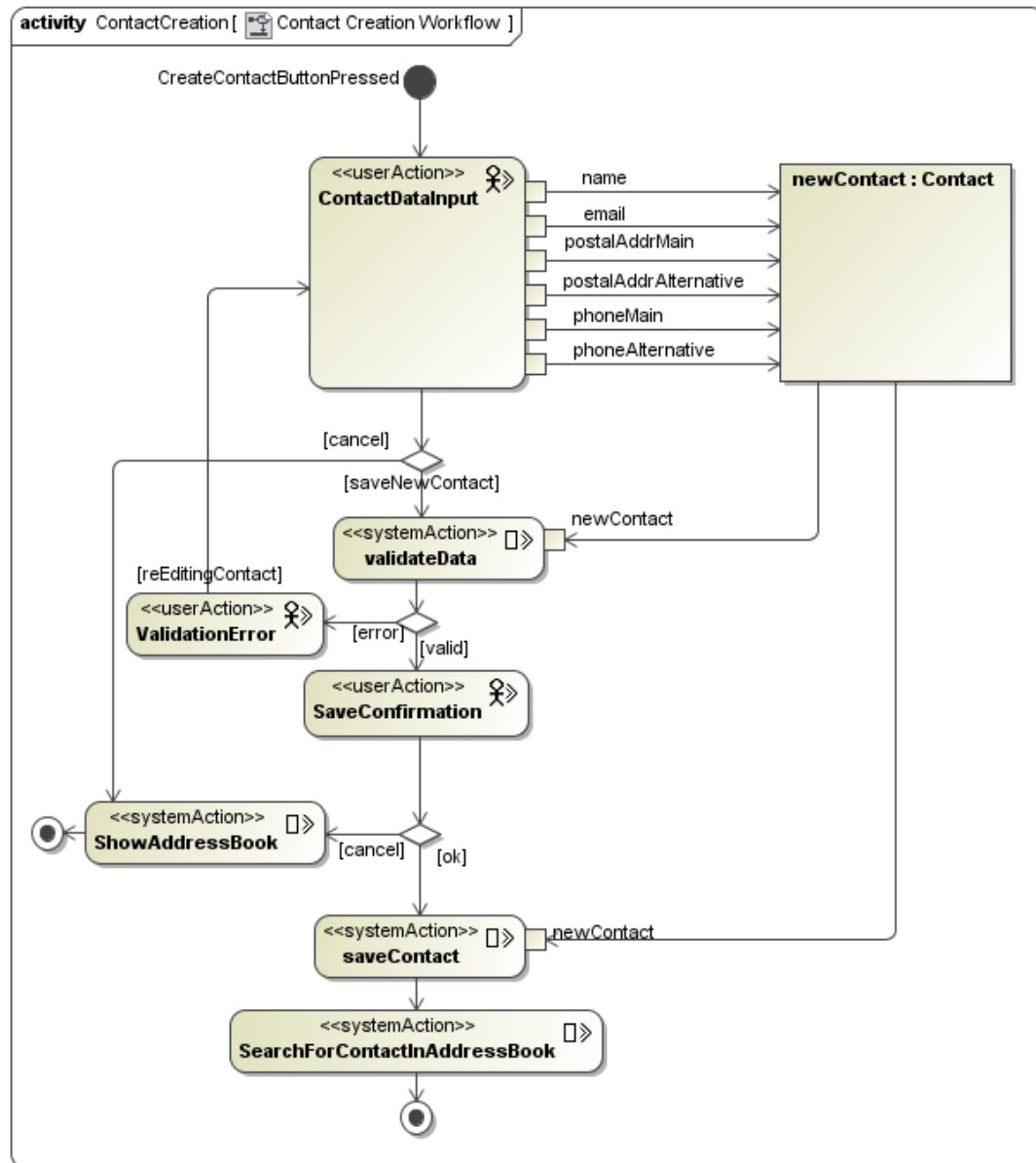
Modelo de estructura de procesos



Nota. Adaptado UWE – UML – Based Web Engineering, de (Munich,2016)

Figura 8.

Modelo de flujo de procesos



Nota. Adaptado UWE – UML – Based Web Engineering, de (Munche,2016)

2.2.4. Codificación Del Sistema.

Durante esta etapa, se llevan a cabo las tareas de programación, que implican traducir el diseño previo en código fuente utilizando el lenguaje de programación seleccionado. Esta fase es crucial para materializar la visión conceptual en un producto funcional.

2.2.5. Pruebas

Las pruebas de caja blanca y caja negra son técnicas ampliamente utilizadas en la actualidad. Las pruebas de caja blanca, consideradas pruebas estructurales, se centran en evaluar el código interno del software, mientras que las pruebas de caja negra se enfocan en probar las entradas y salidas del sistema, independientemente de su implementación interna.

2.2.6. Instalación o Fase de Implementación

La implementación es el proceso mediante el cual los programas desarrollados se transfieren y configuran correctamente en el sistema de destino, con el fin de que estén listos para su uso por parte del usuario final. Este proceso abarca la implementación de la arquitectura, la estructura del espacio de trabajo, el modelo de usuario, la interfaz de usuario y los mecanismos adaptativos, así como la integración de todas estas implementaciones para garantizar su funcionamiento fluido.

2.2.7. Mantenimiento

El concepto mantenimiento como el conjunto de operaciones necesarias para que edificios, máquinas e instalaciones puedan seguir desempeñando adecuadamente su función. Y casi siempre nos olvidamos de hacer mención a una parte fundamental del mantenimiento, la gestión (Cabaleiro, 2015).

2.3. MÉTRICAS DE CALIDAD

2.3.1. Métricas de Calidad de Software

Las Métricas de Calidad proporcionan una indicación de cómo se ajusta el software, a los requerimientos implícitos y explícitos del cliente. El objetivo principal de la ingeniería del software es producir un producto de alta calidad. Para lograr este objetivo, los ingenieros del software deben utilizar mediciones que evalúen la calidad del análisis y los modelos de desafío, el código fuente, y los casos de prueba que se han creado al aplicar la ingeniería del

software. Para lograr esta evaluación de la calidad en tiempo real, el ingeniero debe utilizar medidas técnicas que evalúan la calidad con objetividad, no con subjetividad.

El primer objetivo del equipo de proyecto es medir errores y defectos. Las métricas que provienen de estas medidas proporcionan una indicación de la efectividad de las actividades de control y de la garantía de calidad (Ecured, 2019).

Figura 9.

Métricas de calidad



Nota. Adaptado Ecured, de (Royers,2019)

2.3.2. Generalidades ISO

Las normas ISO son un conjunto de estándares con reconocimiento internacional que fueron creados con el objetivo de ayudar a las empresas a establecer unos niveles de homogeneidad en relación con la gestión, prestación de servicios y desarrollo de productos en la industria (GlobalSuite ISO 27001, 2023).

2.3.3. ISO/IEC 25000

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

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

Figura 10.

Familia de normas ISO/IEC 25000



Nota. Adaptado de Normas ISO/IEC 25000 por (Normas ISO 25000, 2022)

2.3.3.1. ISO/IEC 2500n – Gestión de Calidad de Software. Las normas que forman este apartado definen todos los modelos, términos y definiciones comunes referenciados por todas las otras normas de la familia 25000. Actualmente esta división se encuentra formada por:

- ISO/IEC 25000 - *Guide to SQuaRE*: contiene el modelo de la arquitectura de SQuaRE, la terminología de la familia, un resumen de las partes, los usuarios previstos y las partes asociadas, así como los modelos de referencia.

- ISO/IEC 25001 - *Planning and Management*: establece los requisitos y orientaciones para gestionar la evaluación y especificación de los requisitos del producto software.

2.3.3.2. ISO/IEC 2501n – Modelo de Calidad de Software. Las normas de este apartado presentan modelos de calidad detallados incluyendo características para calidad interna, externa y en uso del producto software. Actualmente esta división se encuentra formada por:

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

Figura 11.

Características de la calidad ISO/IEC 25010



Nota. Adaptado de Normas ISO/IEC 25000 por (Normas ISO 25000, 2022)

- **Adecuación Funcional:** De un producto de software se refiere a su capacidad para proporcionar las funciones que cumplen con los requisitos explícitos e implícitos, en diversas condiciones de uso. Para mejorarla, es crucial comprender a fondo los requisitos del usuario, realizar pruebas exhaustivas, iterar continuamente el desarrollo basado en retroalimentación del usuario, considerar una variedad de escenarios de uso y mantener el software actualizado con mejoras continuas.

Tabla 1.

Definición de métricas de calidad y fórmulas de la adecuación funcional

Característica	Métrica	Fórmula	Descripción
COMPLETITUD	Compleitud de la implementación funcional	$CP = \left[\frac{((x - y) * 100)}{x} \right]$ <p>X= Número total de funcionalidades</p> <p>Y= Numero de las Funcionalidades Incorrectas</p>	Grado en el cual todas las funcionalidades cubren correctamente las tareas para las que fueron diseñadas.
CORRELACIÓN	Correlación entre las funcionalidades	$CF = \left[\frac{((x - y) * 100)}{x} \right]$ <p>x = Total de funcionalidades correlacionales</p> <p>y = total de funcionalidades no correlacionales</p>	Capacidad de la plataforma para relacionar sus funcionalidades y exista un común denominador entre ellas.
PERTINENCIA	Compleitud de la Idoneidad Funcional	$CF = \frac{\left[\frac{((x - y) * 100)}{x} \right]}{10}$	Grado en el cual todas las funcionalidades cubren los objetivos de enseñanza relacionados a la programación.

Nota. La plataforma tiene la capacidad de ser funcionalmente correcto, se divide en 3 subcaracterísticas para evaluar más detalladamente sus métricas de calidad y fórmulas.

- **Eficiencia de Desempeño:** Se trata del rendimiento del producto en función de la cantidad de recursos que consume.

Tabla 2

Definición de métricas y fórmulas de la eficiencia de desempeño

Característica	Métrica	Fórmula	Descripción
COMPORTAMIENTO TEMPORAL	Tiempo de respuesta	$TPR = \frac{(x1 + x2 + x3)}{3}$ <p>X= Tiene 1 de respuesta</p> $CP = (10 * TP) - [11 * (TP - 1)]$	Tiempos de cuando la plataforma lleva acabo una función.
UTILIZACIÓN DE RECURSOS	Utilizando CPU	$UR = \left[x - \left(11 * \left(\frac{x}{10 - 1} \right) \right) \right]$ <p>x = Porcentaje del uso del procesador</p>	Cantidad de recursos utilizados cuando la plataforma lleva acabo una función bajo condición determinada.
CAPACIDAD	Tamaño de memoria usada por el proceso	$C = \left(\frac{x}{10} \right) - \left[11 * \left(\left(\frac{x}{10} \right) - 1 \right) \right]$ <p>x = Tamaño de la memoria usada por el proceso</p>	Cantidad de memoria que emplea por el proceso que se ejecuta en la plataforma.

Nota. Funciona de acuerdo a una cantidad especifica de recursos bajo distintos escenarios ya sean hardware o software.

- **Compatibilidad:** Capacidad de dos o más sistemas o componentes de intercambiar información y llevar a cabo funciones específicas bajo el mismo entorno de hardware y software.

Tabla 3.

Definición de métricas y fórmulas de la compatibilidad

Característica	Métrica	Fórmula	Descripción
COEXISTENCIA	Numero de programas causantes de problemas	$C = (10 * x) - [(10 - x) - 10 * x]$ x = Numero de programas causantes de problemas	Grado en el que la plataforma puede coexistir con otro software independiente.
INTEROPERABILIDAD	Capacidad de intercambiar información con otras plataformas	I = Capacidad de intercambiar información con otras plataformas o programas	Grado en el que dos o más plataformas tienen la capacidad de intercambiar información y emplear la información que fue compartida.

Nota. Llevar a cabo sus funciones cuando comparten el mismo entorno sea hardware o software, se divide en dos subcaracterísticas para evaluar aspectos más detallados de la compatibilidad.

- **Usabilidad:** Capacidad del producto para ser aprendido, atractivo, usado y entendido por el usuario bajo determinadas condiciones.

Tabla 4.

Definición de métricas y fórmulas de la usabilidad

Característica	Métrica	Fórmula	Descripción
INTELIGIBILIDAD	Capacidad software de detectar vacíos conceptuales	$I = \text{Capacidad del software de detectar vacíos conceptuales}$	Capacidad de la plataforma para detectar de acuerdo a los resultados obtenidos por el usuario la deficiencia de conceptos.
APRENDIZAJE	Numero características game-based-learning que cumple la plataforma	$A = \frac{(10 * x)}{7}$ x = Numero características game-based-learning que cumple	Capacidad de la plataforma de cumplir con características que potencialicen las habilidades de la tendencia game based-learning.
OPERABILIDAD	Claridad en mensajes	$O = \frac{[(x - y) * 100]}{x}$ x = Número total de entradas y = Número de entradas de validación	Capacidad de la plataforma para ser de fácil uso a través de mensajes para la aclaración de cualquier punto que pueda ser sujeto a dudas.
PROTECCION FRENTE A ERROR DE USUARIO	Verificación entradas validadas	$PEU = \frac{[(x - y) * 100]}{x}$	Capacidad de la plataforma para proteger a los usuarios a través de validaciones y evitar posibles errores.
ESTETICA	Números de los principios de la evaluación heurística de la interfaz usuario	$E = \frac{(10 * x)}{12}$ x = Numero de principios de estética que cumple	Grado en el que la plataforma cumple con los principios heurísticos de la estética en la interfaz de usuario.
ACCESIBILIDAD	Número de sistemas operativos compatibles.	$E = \frac{(10 * x)}{4}$ x = Número total de sistemas operativos compatibles.	Capacidad de la plataforma de ser accesible a distintos entornos de software.

Nota. Debe ser comprendida, entendida, usada y resuelta por parte del usuario final, por ello se divide en 6 características para evaluar aspectos más detallados de la usabilidad, como métricas y fórmulas.

- **Fiabilidad:** Representa el desempeño de un sistema o componente a la hora de realizar funciones específicas bajo determinadas condiciones y periodos de tiempo determinados. Fiabilidad: Calidad/Tiempo.

Tabla 5.

Definición de métricas fórmulas de la fiabilidad

Característica	Métrica	Fórmula	Descripción
MADUREZ	Numero de fracasos en casos de prueba	$M = \frac{[(x - y) * 100]}{x}$ $M = \frac{[(x - y) * 100]}{10}$ x = Numero de pruebas realizadas y = Numero de fallas detectadas	Capacidad de la plataforma para detectar fallas que determinen la completitud de la misma a través de pruebas de validación.
DISPONIBILIDAD	Tiempo de operación de servicios	$D = \frac{[(x - y) * 100]}{x}$ $D = \frac{[(x - y) * 100]}{10}$ x = Tiempo de uso en la plataforma en segundos y = Tiempo de no disponibilidad en la plataforma	Capacidad de la plataforma o componente de estar operativo por su uso cuando se requiere en un periodo de tiempo específico.
TOLERANCIA ERRORES	A Tiempo perdido por errores	$TF = (x - y) * NI$ x = Tiempo total de uso de la plataforma en minutos y = Sumatoria de tiempo de duración de errores	Capacidad de la plataforma para operar según lo previsto en presencia de errores ya sean de hardware o software en un tiempo específico.
CAPACITACION	Tiempo promedio de recuperación	$TPR = \frac{(x1 + x2 + x3)}{3}$ x = Tiempo de respuesta en un minuto $CR = \frac{(10 * TP)}{[11 * (TP - 1)]}$	Capacidad de la plataforma de recuperarse ante un error inesperado y por ende establecerse en un periodo de tiempo específico.

Nota. Capacidad de desempeñar una función específica, se divide en 4 características para evaluar aspectos más detallados de la fiabilidad, con sus métricas y fórmulas.

- **Seguridad:** Capacidad de proteger la información de manera tal que no puedan ser leídos o modificados a cualquier persona o sistema no autorizados.

Tabla 6.*Definición de métricas y fórmulas de la seguridad*

Característica	Métrica	Fórmula	Descripción
CONFIDENCIALIDAD	Numero de métodos de recuperación y validación de datos	$M = (x-y) * NI$ $x =$ Numero de formas de recuperación y validación de datos $y =$ Numero de formas de recuperación o validación de datos no validas en la plataforma	Capacidad de la plataforma para notificar accesos o modificaciones por personas externas no autorizadas. Si se cumple que $x=y$ es positivo. Si se cumple $x<y$ es negativo.
INTEGRIDAD	Numero solicitudes y mensajes de alerta recibidos	$I = (x-y) * NI$ $x =$ Número de solicitudes de cambio $y =$ Numero de mensajes de alerta recibidos	Formas que tiene la plataforma para demostrar la identidad correcta de un usuario.
AUTENTICIDAD	Numero de formas de autenticación	$A = (x-y) * NI$ $x =$ Numero de formas de autenticación $y =$ Numero de formas que no permite autenticación	Capacidad de la plataforma para rastrear la forma precisa las acciones maliciosas de una persona no autorizada. Si se cumple que $x=y$ es positivo. Si se cumple $x<y$ es negativo.
RESPONSABILIDAD	Capacidad de la plataforma de rastrear ingresos maliciosos	$x =$ Numero de ingresos maliciosos $y =$ Numero de rastreos exitosos	Capacidad de la plataforma para proteger la información y los datos y personas.

Nota. Para proteger la información y los datos y por ende personas no autorizadas no puedan leer o modificarlos, se divide en 4 características para evaluar aspectos de la seguridad, como sus métricas y fórmulas.

- **Mantenibilidad:** Representa el esfuerzo requerido para realizar modificaciones de forma efectiva y eficiente debido a necesidades.

Tabla 7.

Definición de métricas y fórmulas de la mantenibilidad

Característica	Métrica	Fórmula	Descripción
MODULARIDAD	Numero de módulos concatenados	$M = \frac{[(x - y) * 100]}{x}$ $M = \frac{[(x - y) * 100]}{10}$ x = Número total de módulos y = Numero de módulos no concatenados	Grado en el que la plataforma tiene separados todos sus componentes para que la afección en alguno tenga un impacto mínimo en los demás.
REUZABILIDAD	Número de atributos y funciones heredadas	$R = (x - y) * NI$ x = Suma de atributos y funciones de todos los objetos del software y = Suma de atributos y funciones heredadas de todos los objetos	Grado en el que el código de la plataforma puede ser utilizado en más de un sistema software o en la construcción de otros.
ANALIZABILIDAD	Fallas diagnosticadas	$A = \frac{[(x - y) * 100]}{x}$ $A = \frac{[(x - y) * 100]}{10}$ x = Número de las fallas encontradas y = Número total de fallas no diagnosticadas	Grado en el que la plataforma tiene la facilidad de encontrar fallas en el mismo y diagnosticar las causas de los errores.
CAPACIDAD DE SER MODIFICADO	Complejidad de la modificación	$CM = \frac{x}{y}$ x = Numero de modificaciones y = Tiempo que le toma al desarrollador hacer las modificaciones	Grado en el que la plataforma permite que sea modificado de forma eficiente sin afectar su desempeño.
CAPACIDAD DE SER PROBADO	Número de reinicios para pruebas	$A = \frac{[(x - y) * 100]}{x}$ $A = \frac{[(x - y) * 100]}{10}$ x = Número total de reinicios y = Numero de reinicios fallidos	Grado en el que la plataforma tiene la facilidad de reiniciar sus funciones y restablecerse después de las pruebas realizadas.

Nota. Se divide en 5 características para evaluar aspectos más detallados de la mantenibilidad, como sus métricas y fórmulas.

- **Portabilidad:** Comprobar la capacidad del software para ser utilizado en diferentes plataformas y entornos (ISO, 2024).

Tabla 8.

Definición de métricas y fórmulas de la portabilidad

Característica	Métrica	Fórmula	Descripción
ADAPTABILIDAD	Capacidad de que pueda adaptarse a otros entornos	A=Capacidad de adaptarse a otros entornos Tabla de cumplimiento	Capacidad del producto para ser adaptado de forma exitosa a diferentes entornos determinados de hardware o software.
FACILIDAD DE INSTALACION	Número de pasos previo a la Instalación	$D = (10 * x) - [10 * (x - 1)]$ x = Numero de pasos para la instalación	Grado en el que la plataforma tiene la facilidad para ser instalado de forma exitosa.
CAPACIDAD DE SER REEMPLAZADO	Capacidad de ser reemplazado	CR = Capacidad de ser reemplazado Tabla de cumplimiento	Capacidad de la plataforma para que pueda ser utilizada en lugar de otro software con el mismo propósito.

Nota. Capacidad de la plataforma para ser transferida exitosamente a cualquier entorno, se divide en 3 características para evaluar aspectos más detallados de la portabilidad, como sus métricas y fórmulas.

- **ISO/IEC 25012** - El modelo de Calidad de Datos representa los cimientos sobre los cuales se construye un sistema para la evaluación de un producto de datos. En un modelo de Calidad de Datos se establecen las características de Calidad de Datos que se deben tener en cuenta a la hora de evaluar las propiedades de un producto de datos determinado. Las características de Calidad de Datos están clasificadas en dos grandes categorías:
 - **Calidad de Datos Inherente:** Se refiere al grado con el que las características de calidad de los datos tienen el potencial intrínseco para satisfacer las

necesidades establecidas y necesarias cuando los datos son utilizados bajo condiciones específicas. Desde el punto de vista inherente, la Calidad de Datos se refiere a los mismos datos.

- **Calidad de Datos Dependiente del Sistema:** Se refiere al grado con el que la Calidad de Datos es alcanzada y preservada a través de un sistema informático cuando los datos son utilizados bajo condiciones específicas.

Desde el punto de vista dependiente del sistema, la Calidad de Datos depende del dominio tecnológico en el que los datos se utilizan, y se alcanza mediante las capacidades de los componentes del sistema informático tales como: dispositivos hardware (e.g., Respaldo Software para alcanzar la Recuperabilidad), y otro software (e.g., Herramientas de migración para alcanzar la Portabilidad) (ISO 25012, 2024)

2.3.3.3. ISO/IEC 2502n – Modelo de Calidad de Software. Estas normas incluyen un

modelo de referencia de la medición de la calidad del producto, definiciones de medidas de calidad (interna, externa y en uso) y guías prácticas para su aplicación.

Actualmente esta división se encuentra formada por:

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

- ISO/IEC 25022 - *Measurement of quality in use*: define específicamente las métricas para realizar la medición de la calidad en uso del producto.
- ISO/IEC 25023 - *Measurement of system and software product quality*: define específicamente las métricas para realizar la medición de la calidad de productos y sistemas software.
- ISO/IEC 25024 - *Measurement of data quality*: define específicamente las métricas para realizar la medición de la calidad de datos (ISO/IEC 2502n, 2024).

2.3.3.4. ISO/IEC 2503n – Requisitos de Calidad de Software. Las normas que forman este apartado ayudan a especificar requisitos de calidad que pueden ser utilizados en el proceso de elicitación de requisitos de calidad del producto software a desarrollar o como entrada del proceso de evaluación. Para ello, este apartado se compone de:

- ISO/IEC 25030 - *Quality requirements*: provee de un conjunto de recomendaciones para realizar la especificación de los requisitos de calidad del producto software (ISO/IEC 2503n, 2024).

2.3.3.5. ISO/IEC 2504n – Evaluación De Calidad De Software. Este apartado incluye normas que proporcionan requisitos, recomendaciones y guías para llevar a cabo el proceso de evaluación del producto software. Esta división se encuentra formada por:

- ISO/IEC 25040 - *Evaluation reference model and guide*: propone un modelo de referencia general para la evaluación, que considera las entradas al proceso de evaluación, las restricciones y los recursos necesarios para obtener las correspondientes salidas.
- ISO/IEC 25041 - *Evaluation guide for developers, acquirers and independent evaluators*: describe los requisitos y recomendaciones para la implementación

práctica de la evaluación del producto software desde el punto de vista de los desarrolladores, de los adquirentes y de los evaluadores independientes.

- ISO/IEC 25042 - *Evaluation modules*: define lo que la Norma considera un módulo de evaluación y la documentación, estructura y contenido que se debe utilizar a la hora de definir uno de estos módulos.
- ISO/IEC 25045 - *Evaluation module for recoverability*: define un módulo para la evaluación de la subcaracterística Recuperabilidad (Recoverability) (ISO/IEC 2504n, 2024).

2.4. SEGURIDAD DE SOFTWARE ISO – 27001

La norma ISO 27001 es un estándar internacional que establece los requisitos para la implementación, mantenimiento y mejora continua de un Sistema de Gestión de la Seguridad de la Información (SGSI). Este sistema se utiliza para proteger la confidencialidad, integridad y disponibilidad de la información. La norma proporciona un marco para la seguridad de la información que ayuda a las organizaciones a identificar y gestionar sus riesgos de seguridad de la información de manera efectiva.

Aplicabilidad de la norma ISO 27001: La norma ISO 27001 se aplica a cualquier tipo de organización, incluyendo pequeñas y medianas empresas, grandes corporaciones, instituciones gubernamentales y sin fines de lucro. También se puede aplicar en cualquier sector, incluyendo tecnología de la información, finanzas, salud y servicios públicos.

Proceso de implementación de la norma ISO 27001: El proceso de implementación de la norma ISO 27001 se divide en cuatro fases: planificación, implementación, evaluación y mejora continua.

- **Fase de planificación:** Durante la fase de planificación, la organización identifica sus requisitos de seguridad de la información y establece un plan para implementar el SGSI.
- **Fase de implementación:** La fase de implementación incluye la creación de políticas, procedimientos y controles para proteger la información.
- **Fase de evaluación:** Durante la fase de evaluación, la organización evalúa la eficacia de su SGSI e identifica áreas de mejora.
- **Fase de mejora continua:** La fase de mejora continua implica la identificación y aplicación de mejoras a los procesos y controles del SGSI. Una vez implementado y certificado, el SGSI debe ser revisado y actualizado regularmente para garantizar su continuo cumplimiento con los requisitos de seguridad de la información. La certificación ISO 27001, aunque no es obligatoria, también puede mejorar la imagen de la marca y la confianza de los clientes, ya que demuestra que la organización está comprometida con la protección de la información y esto lo acredita una entidad certificadora independiente.

Implantar la norma ISO/IEC 27001 con un software puede ofrecer varias ventajas, tales como se mostrará en la figura (GlobalSuite ISO 27001, 2023).

Figura 12.

Ventajas de la ISO/IEC 27001



Nota. Adaptado de GlobalSuite, Norma ISO 27001 (Solutions, 2023)

2.5. Pruebas de Funcionamiento

La usabilidad, el rendimiento y la fiabilidad son de vital importancia en su sector, y «defectuoso» es la palabra más nociva que alguien puede utilizar para definir el software. Su software y sus productos deben ser fiables y fáciles de usar. Necesita tener la confianza de que puedan resistir cualquier cosa: pruebas de estrés, localización a nuevos mercados de destino y el uso práctico cotidiano (Vistatec, 2024).

2.5.1. Objetivos

- Probar si el software no hace lo que debe.
- Probar si el software hace lo que no debe, es decir, si provoca efectos secundarios adversos.
- Descubrir un error que aún no ha sido descubierto.
- Encontrar el mayor número de errores con la menor cantidad de tiempo y esfuerzo posibles.

- Mostrar hasta qué punto las funciones del software operan de acuerdo con las especificaciones y requisitos del cliente.

2.5.2. Tipos de prueba de Software

Hay muchos tipos diferentes de pruebas de software, cada una con objetivos y estrategias específicos:

- **Prueba de aceptación:** verifica si todo el sistema funciona según lo previsto.
- **Pruebas de integración:** asegura que los componentes o funciones del software operen juntos.
- **Pruebas de unidad:** valida que cada unidad de software funcione según lo esperado. Una unidad es el componente de prueba más pequeño de una aplicación.
- **Pruebas funcionales:** verifica funciones mediante la emulación de escenarios de negocio, en función de los requisitos funcionales. La prueba de caja negra es una forma común de verificar funciones.
- **Pruebas de rendimiento:** prueba cómo funciona el software bajo diferentes cargas de trabajo. Las pruebas de carga, por ejemplo, se utilizan para evaluar el rendimiento en condiciones de carga reales.
- **Pruebas de regresión:** verifica si las nuevas características rompen o degradan la funcionalidad. Las pruebas de cordura se pueden utilizar para verificar menús, funciones y comandos a nivel superficial, cuando no hay tiempo para una prueba de regresión completa.
- **Pruebas de estrés:** prueba cuánta tensión puede soportar el sistema antes de que falle. Considerado como un tipo de prueba no funcional.

- **Pruebas de usabilidad:** valida qué tan bien un cliente puede usar un sistema o una aplicación web para completar una tarea.

En cada caso, la validación de los requisitos básicos es una evaluación crítica. Igual de importante, las pruebas exploratorias ayudan a un tester o equipo de pruebas a descubrir escenarios y situaciones difíciles de predecir que pueden conducir a errores de software.

Incluso una aplicación simple puede estar sujeta a una gran cantidad y variedad de pruebas. Un plan de gestión de pruebas ayuda a priorizar qué tipos de pruebas proporcionan el mayor valor, dado el tiempo y los recursos disponibles. La efectividad de las pruebas se optimiza ejecutando la menor cantidad de pruebas para encontrar la mayor cantidad de defectos.

2.5.3. Técnicas De Prueba

Los tipos más comunes de pruebas de software incluyen:

- Técnicas estáticas
- Técnicas dinámicas.
- Técnicas basadas en la experiencia.

Como se verá más adelante, las pruebas dinámicas detectan los fallos, mientras que las pruebas estáticas detectan sus causas, los defectos.

2.5.4. Técnicas Estáticas

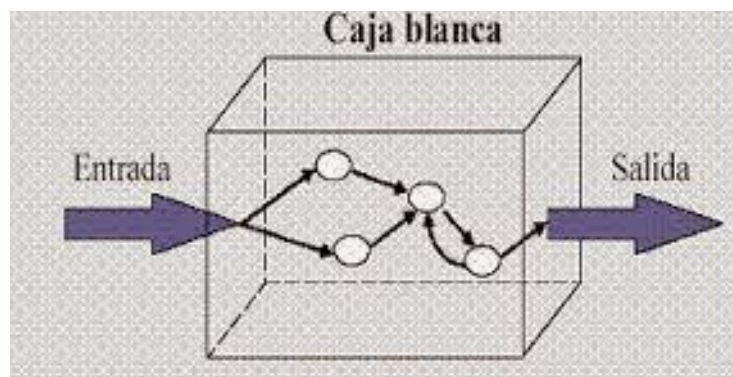
2.5.4.1. Análisis estático. El análisis estático no implica la ejecución dinámica del software bajo prueba y puede detectar posibles defectos en una etapa temprana, antes de ejecutar el programa. El análisis estático se realiza durante o después de la codificación y antes de ejecutar las pruebas unitarias. Puede ser ejecutado por un motor de análisis de código para "recorrer" automáticamente el código fuente y detectar reglas que no cumplen, o errores léxicos, sintácticos e incluso algunos semánticos.

2.5.5. Técnicas Dinámicas

Este tipo de técnicas son las realizadas ejecutando la aplicación y son las utilizadas para el diseño de los casos de prueba. La mayoría del software puede probarse de dos maneras diferentes. Conociendo el funcionamiento interno, podemos probar que todos los módulos encajan unos con otros, es decir, desde una visión interna. Estas pruebas son las pruebas de caja blanca. Al conocer las funciones específicas del producto se pueden llevar a cabo pruebas que demuestren que estas funciones son operativas y la búsqueda de errores en dichas funciones. Estas pruebas se realizan desde una visión externa, mediante las pruebas de caja negra. Estas dos técnicas nos ayudarán a definir los casos de prueba para tener la mayor probabilidad de encontrar errores ahorrando esfuerzo y tiempo.

2.5.6. Técnicas De Caja Blanca

La técnica de caja blanca, a veces definida como prueba de "caja de cristal" o "caja transparente", es una técnica de diseño de casos de prueba que usa la estructura de control para obtener los casos de prueba.

Figura 13.*Esquema de caja blanca*

Nota. Adaptado blogspot, técnica de caja blanca (Archivos de blog, 2015)

Dentro de esta estructura de control podemos encontrar la estructura de un componente de software como puede ser sentencias de decisiones, caminos distintos del código, la estructura de una página web, etc. Los métodos de prueba de caja blanca aportan los siguientes puntos:

- Garantizan que todas las rutas del código se revisan al menos una vez.
- Revisan las condiciones lógicas.
- Revisan estructuras de datos (Peño, 2015).

2.5.7. Técnica de Caja Negra

Las técnicas de diseño de caja negra, también llamadas pruebas de comportamiento, son las que utilizan el análisis de la especificación, tanto funcional como no funcional, sin tener en cuenta la estructura interna del programa para diseñar los casos de prueba y, a diferencia de las pruebas de caja blanca, estas pruebas se suelen realizar durante las últimas etapas de la prueba.

Figura 14.*Esquema de caja negra*

Nota. Adaptado de PMOinformatica, técnica de caja negra de (PMOinformatica, 2016)

Con los métodos de caja negra se intenta encontrar los errores: • Funciones incorrectas o faltantes.

- Errores de inicialización y terminación.
- Errores de interfaz.
- Errores en las estructuras (Peño, 2015).

2.5.8. Pruebas de Estrés

Las pruebas de estrés proporcionan información valiosa sobre las características de rendimiento del software. Al detectar cuellos de botella en el rendimiento, ineficiencias y áreas de mejora, las pruebas de estrés ayudan a optimizar el rendimiento del software, lo que se traduce en un sistema más rápido y con mayor capacidad de respuesta (Singureanu, 2022)

2.6. ESTIMACIÓN DE COSTO COCOMO II

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

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

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

2.6.1. Modelo De Composición De Aplicación

La fórmula propuesta de este modelo es la siguiente:

(1). Ecuación de composición de aplicación

$$PM = \frac{NOP}{PROD} \quad (1)$$

Donde:

PM = Meses Persona

NOP = Tamaño del nuevo software a desarrollar expresado en puntos objeto y se calcula de la siguiente manera.

(2). Ecuación de tamaño del nuevo software a desarrollar

$$NOP = \frac{OP * (100 - \%reuso)}{100} \quad (2)$$

OP = Tamaño del software a desarrollar expresado en Puntos Objeto.

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

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

2.6.2. Modelo Diseño Temprano

Este modelo se usa en las etapas tempranas de un proyecto de software, cuando se conoce muy poco del tamaño del producto a ser desarrollado, de la naturaleza de la plataforma, del personal a ser incorporado al proyecto o detalles específicos del proceso a utilizar. Este modelo podría emplearse tanto en productos desarrollados en sectores de Generadores de Aplicación, Sistemas Integrados o Infraestructura. El modelo de Diseño Temprano ajusta el esfuerzo nominal usando siete factores de costo. La fórmula para el cálculo del esfuerzo es la siguiente:

(3). Ecuación de modelo diseño temprano

$$PM_{estimado} = PM_{nominal} * \prod_{i=1}^7 EM_i \quad (3)$$

$PM_{estimado}$ Es el esfuerzo Nominal ajustado por 7 factores que reflejan aspectos propuso del proyecto

EM_i Corresponde a los factores de costo tienen un efecto multiplicativo

(4). Ecuación del esfuerzo nominal

$$PM_{nominal} = A * (KSLOC)^B \quad (4)$$

A es una constante que captura los efectos sobre el esfuerzo (A=2.94).

KSLOC es el tamaño del software expresado en líneas de código.

B es un factor exponencial de escala, toma en cuenta las características relacionadas.

(5). Ecuación del factor exponencial de escala

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

B < 1.0 el proyecto muestra deseconomía de escala.

Si el B = 1.0 las economías y deseconomías de escala de equilibrio.

Si el B > 1.0 el proyecto muestra deseconomía de escala.

2.6.3. Modelo Post – Arquitectura

Es el modelo de estimación más detallado y se aplica cuando la arquitectura del proyecto está completamente definida. Este modelo se aplica durante el desarrollo y mantenimiento de productos de software incluidos en las áreas de Sistemas Integrados, Infraestructura y Generadores de Aplicaciones. El esfuerzo nominal se ajusta usando 17 factores multiplicadores de esfuerzo. El mayor número de multiplicadores permite analizar con más exactitud el conocimiento disponible en las últimas etapas de desarrollo, ajustando el modelo de tal forma que refleje fielmente el producto de software bajo desarrollo. La fórmula para el cálculo del esfuerzo es la siguiente (Gomez, 2021).

(6). Ecuación de modelo post – arquitectura

$$PM_{estimado} = PM_{nominal} * \prod_{i=1}^7 EM_i \quad (6)$$

Fórmula para hallar el factor de complejidad TCF:

(7). Ecuación de factor de complejidad

$$TCF = (0.65 + 0.01 * PF) \quad (7)$$

El procesamiento de datos el punto función se basa en la fórmula siguiente:

(8). Ecuación del punto función

$$PF = Cuenta\ total * TCF \quad (8)$$

Factor LDC/PF se calcula con la fórmula:

(9). Ecuación de factor LDC/PF

$$LDC = PF * Factor \frac{LDC}{PF} \quad (9)$$

Número estimado de líneas de código distribuidas en miles se calcula con:

(10). Ecuación número estimado de líneas de código

$$KLCD = \frac{LDC}{1000} \quad (10)$$

Las ecuaciones de COCOMO básico tiene la siguiente forma:

(11). Ecuación de COCOMO básico

$$E = a_b(KLCD)^{bb} \quad (11)$$

$$D = C_b D^{db}$$

Donde:

E: Esfuerzo aplicado en personas por mes.

D: Tiempo de desarrollo de meses cronológicos.

KLDC: número estimado de líneas de código (en miles).

2.7. Herramientas

2.7.1. Lenguaje De Programación

2.7.1.1. PHP. (PHP: Hypertext Preprocessor) **es un** lenguaje de programación **codigo abierto ampliamente utilizado en el desarrollo web**. En otras palabras, es la tecnología que permite crear sitios web dinámicos, donde el contenido puede cambiar según la interacción del usuario. Es uno de los lenguajes de programación más populares del mundo, y se utiliza para crear una amplia gama de aplicaciones web, desde pequeñas páginas personales hasta grandes sitios web corporativos. Su flexibilidad y facilidad de uso lo han convertido en una opción preferida para construir aplicaciones web dinámicas y sitios interactivos. Sus principales características son:

- **Tipo de lenguaje:** PHP es un lenguaje del lado del servidor. Esto significa que se ejecuta en el servidor web y genera código HTML que se envía al navegador del usuario. El navegador recibe contenido estático, sin acceso directo al código fuente PHP.
- **Uso principal:** Se utiliza principalmente para generar contenido dinámico en el servidor, como páginas web, formularios y manejo de bases de datos. Es ideal

para tareas que requieren acceso a bases de datos y procesamiento en el servidor antes de enviar contenido al navegador.

- **Sintaxis:** Tiene una sintaxis similar a otros lenguajes de programación como C y Perl. Se incorpora directamente en el código HTML utilizando etiquetas especiales `<? php ?>`.
- **Ejecución:** Se ejecuta en el servidor antes de que se entregue una página web al navegador del usuario. El resultado (generalmente HTML) se envía al navegador.
- **Seguridad:** Al ser ejecutado en el servidor, el código PHP no es visible para los usuarios. Sin embargo, se deben tomar precauciones para evitar vulnerabilidades en el código (Arsys, 2022).

2.7.1.2. Java Script. JavaScript es un robusto lenguaje de programación que se puede aplicar a un documento HTML y usarse para crear interactividad dinámica en los sitios web. Fue inventado por Brendan Eich, cofundador del proyecto Mozilla, Mozilla Foundation y la Corporación Mozilla. Puedes hacer casi cualquier cosa con JavaScript. Puedes empezar con pequeñas cosas como carruseles, galerías de imágenes, diseños fluctuantes, y respuestas a las pulsaciones de botones. Con más experiencia, serás capaz de crear juegos, animaciones 2D y gráficos 3D, aplicaciones integradas basadas en bases de datos ¡y mucho más!

JavaScript por sí solo es bastante compacto, aunque muy flexible, y los desarrolladores han escrito gran cantidad de herramientas encima del núcleo del lenguaje JavaScript, desbloqueando una gran cantidad de funcionalidad adicional con un mínimo esfuerzo. Esto incluye:

- Interfaces de Programación de Aplicaciones del Navegador (APIs) — APIs construidas dentro de los navegadores que ofrecen funcionalidades como crear dinámicamente contenido HTML y establecer estilos CSS, hasta capturar y manipular un vídeo desde la cámara web del usuario, o generar gráficos 3D y muestras de sonido.
- APIs de terceros, que permiten a los desarrolladores incorporar funcionalidades en sus sitios de otros proveedores de contenidos como Twitter o Facebook.
- Marcos de trabajo y librerías de terceros que puedes aplicar a tu HTML para que puedas construir y publicar rápidamente sitios y aplicaciones (MDN, 2023).

2.7.2. Base de datos

Una base de datos (database) almacena datos y los conecta en una unidad lógica junto a los metadatos necesarios para su procesamiento. Las bases de datos son instrumentos de gran utilidad para gestionar grandes ficheros y facilitar la consulta de información. En muchas, además, puede definirse un esquema de permisos que establece qué personas o programas pueden acceder a los datos, y a cuáles, con el objetivo de presentar el contenido de forma adecuada y clara.

Los distintos sistemas de bases de datos se diferencian conceptualmente entre sí y tienen, por lo tanto, sus propias ventajas y desventajas. Pero, antes que nada, es conveniente diferenciar entre la base de datos en sí y el sistema que la gestiona. Como base de datos se designa al conjunto de los datos que se ha de ordenar, mientras que el sistema de gestión de la base de datos (SGBD) es responsable de su administración, determinando así su estructura, el orden, los permisos de acceso, las dependencias, etc. Para ello acostumbra a utilizar un compilador propio y un modelo adecuado de base de datos que determina la arquitectura del sistema de base de datos (IONS, 2024).

2.7.2.1. PostgreSQL. PostgreSQL, comúnmente pronunciado "Post-GRES", es una **base de datos de código** abierto que tiene una sólida reputación por su fiabilidad, flexibilidad y soporte de estándares técnicos abiertos. A diferencia de otros RDMBS (sistemas de gestión de bases de datos relacionales), PostgreSQL (enlace externo a ibm.com) soporta tipos de datos relacionales y no relacionales. Esto la convierte en una de las bases de datos relacionales más compatibles, estables y maduras disponibles actualmente.

Su diseño versátil y accesible convierte a PostgreSQL en una solución de "talla única" para muchas empresas que buscan formas rentables y eficientes de mejorar sus sistemas de gestión de bases de datos. Creada como una solución de base de datos de código abierto (enlace externo a ibm.com), PostgreSQL está completamente libre de restricciones de licencia, potencial de bloqueo de proveedores o riesgo de implementación excesiva (IBM, 2021).

2.7.2.2. Navicat. Navicat es un software de gestión y administración de bases de datos. Permite a los usuarios conectarse y administrar múltiples tipos de bases de datos, incluyendo MySQL, MariaDB, SQL Server, PostgreSQL y Oracle, entre otros. Esta herramienta es útil para desarrolladores, administradores de bases de datos y otros profesionales de la tecnología. Cuenta con una interfaz para crear, modificar y gestionar los objetos de la base de datos, lo que facilita su diseño, administración y mantenimiento. Navicat es de gran utilidad en diversas industrias, incluyendo tecnología, finanzas, educación, gobierno y más. Cualquier organización que necesite gestionar grandes cantidades de datos puede beneficiarse de este software (Navicat Shop, 2024).

2.7.3. Framework

Un framework es un conjunto de reglas y convenciones que se usan para desarrollar software de manera más eficiente y rápida. Estos marcos de trabajo se emplean para ahorrar

tiempo y esfuerzo en el desarrollo de aplicaciones, ya que proporcionan una estructura básica que se puede utilizar como punto de partida. Además, los frameworks también ofrecen soluciones a problemas comunes en el desarrollo de software, lo que significa que los desarrolladores pueden centrarse en las funcionalidades específicas de su aplicación en lugar de perder tiempo resolviendo problemas técnicos (Lucena, 2023).

2.7.3.1. Laravel. Laravel es un framework PHP gratis y de código abierto que brinda un conjunto de herramientas y recursos para crear aplicaciones modernas. Posee un ecosistema integral que combina funciones integradas y una variedad de paquetes y extensiones compatibles. Este framework de PHP creció en popularidad rápidamente en los últimos años, y muchos desarrolladores lo adoptaron como su framework de trabajo favorito para lograr un proceso de desarrollo optimizado (Talently, 2022).

2.7.3.2. Bootstrap. Bootstrap es un framework de desarrollo web gratuito y de código abierto. Está diseñado para facilitar el proceso de desarrollo de los sitios web responsivos y orientados a los dispositivos móviles, proporcionando una colección de sintaxis para diseños de plantillas.

En otras palabras, Bootstrap ayuda a los desarrolladores a construir sitios web más rápidamente, ya que no tienen que preocuparse por los comandos y funciones básicos. Consta de scripts basados en HTML, CSS y JS para diversas funciones y componentes relacionados con el diseño web (Deyimar, 2020).

2.7.4. HTML

HTML (Lenguaje de marcas de Hipertexto, del inglés HyperText Markup Language) es el componente más básico de la Web. Define el significado y la estructura del contenido web.

Además de HTML, generalmente se utilizan otras tecnologías para describir la apariencia presentación de una página web (CSS) o la funcionalidad/comportamiento (JavaScript).

"Hipertexto" hace referencia a los enlaces que conectan páginas web entre sí, ya sea dentro de un único sitio web o entre sitios web. Los enlaces son un aspecto fundamental de la Web. Al subir contenido a Internet y vincularlo a las páginas creadas por otras personas, te conviertes en un participante activo en la «World Wide Web» (Red Informática Mundial) (Mdn HTML, 2023).

2.7.5. CSS

Las hojas de estilo en cascada (CSS, cascading style sheets) permiten crear páginas web atractivas. Pero ¿cómo funcionan realmente? En este artículo explicaremos qué es el CSS con un ejemplo de sintaxis sencillo y describiremos algunos términos clave sobre este lenguaje.

En el módulo Introducción al HTML, exponemos qué es el HTML y cómo se usa para definir documentos destinados a leerse en un navegador web. Los títulos se verán más grandes que el texto y los párrafos empezarán en una línea nueva y habrá un espacio entre ellos. Los enlaces aparecerán en un color diferente y subrayados para distinguirlos del resto del texto. Vienen predeterminados por el navegador y, en la práctica, son estilos muy básicos que el navegador aplica al HTML para asegurarse, básicamente, de que sean legibles incluso si el autor de la página no especifica un estilo explícito (MDN CSS, 2023).

2.7.6. Editor de código

2.7.6.1. Visual Studio Code. Visual Studio Code (VS Code) es un editor de código fuente desarrollado por Microsoft. Es software libre y multiplataforma, está disponible para Windows, GNU/Linux y macOS. VS Code tiene una buena integración con Git, cuenta con soporte para depuración de código, y dispone de un sinnúmero de extensiones, que básicamente te da la posibilidad de escribir y ejecutar código en cualquier lenguaje de programación. VS Studio Code incluye una terminal con todas las funciones, la cual se inicia fácilmente en el directorio de trabajo. La terminal integrada puede utilizar cualquier Shell instalado en el equipo, como PowerShell, Bash o cualquier otro. Contar con una terminal en el propio editor es de gran utilidad para ejecutar diferentes comandos necesarios cuando estamos desarrollando (Flores, 2022).

2.7.7. Composer

Composer es el gestor de dependencias para PHP. Sirve para descargar y gestionar las librerías y dependencias de un proyecto PHP de manera automatizada. Cuando se desarrolla una aplicación o sitio web en PHP, a menudo se utilizan diferentes librerías o frameworks de terceros para añadir funcionalidades adicionales. Composer permite gestionar estas dependencias de manera sencilla, descargándolas y manteniéndolas actualizadas de manera automática.

Para utilizar Composer, se crea un archivo llamado "composer.json" en el raíz del proyecto, en el que se especifican las dependencias necesarias y se ejecuta el comando "composer install" para descargar las dependencias y crear un archivo "vendor/autoload.php" que se puede incluir en el código de la aplicación para cargar automáticamente las dependencias (Composer, 2022).

CAPÍTULO III

MARCO APLICATIVO

CAPÍTULO III

3. INTRODUCCIÓN

El capítulo describe el proceso de desarrollo del "Sistema Web para el Seguimiento de Trámites" en el Gobierno Autónomo

Municipal de Huatajata, utilizando un marco de trabajo para estructurar, planificar y controlar el proceso. Se emplea un enfoque de desarrollo de software que incorpora herramientas libres, modelos y técnicas para garantizar la calidad del software. Las etapas del desarrollo del sistema incluyen la obtención de requisitos, análisis y diseño, codificación, pruebas, implantación y mantenimiento, siguiendo la metodología UWE (UML-Based Web Engineering), que se detalla en el capítulo II. El enfoque iterativo e incremental de UWE permite realizar mejoras continuas en el software, asegurando que se adapte a las necesidades cambiantes de los usuarios y del entorno. Los modelos propuestos por UWE, como el Modelo de Requerimientos, Modelo Conceptual, Modelo de Navegación y Modelo de Presentación, proporcionan herramientas efectivas para guiar el desarrollo del sistema y garantizar su calidad.

3.1. METODOLOGÍA UWE – CONSTRUCCIÓN E IMPLEMENTACIÓN

La metodología UWE se basa en el uso de UML para el desarrollo de aplicaciones web. Se divide en fases como análisis, diseño, desarrollo, pruebas, implementación y mantenimiento. Utiliza modelos como el de Requerimientos, Conceptual, Navegación y Presentación. Adopta un enfoque iterativo e incremental, permitiendo entregas tempranas y adaptación a cambios. Además, hace uso de herramientas de modelado UML, frameworks web y prácticas de ingeniería de software para guiar el proceso de desarrollo de manera eficiente y estructurada.

3.1.1. Modelo de Requerimientos

3.1.1.1. Actores. Los actores representan un rol que puede desempeñar alguien o algo que interactúa o que necesita intercambiar información con el sistema. Para identificar a los Actores se permitió conocer a las personas encargadas en el manejo de trámites del Gobierno Autónomo Municipal de Huatajata. En la siguiente tabla se muestra la lista de actores, junto con una descripción de sus actividades relacionadas con el sistema.

Tabla 9.

Tabla de actores

ACTOR	DESCRIPCION DE ACTIVIDADES
USUARIO	Encargado de la gestión de tramites dentro de la alcaldía. Recibe y registra los tramites que provienen de unidades externas, despachan tramites, generan un registro y control de los tramites que ingresan o salen de la alcaldía, y se generan nuevos tramites.
ADMINISTRADORES	Personal responsable de la gestión y mantenimiento del sistema informático de tramites.
FUNCIONARIOS	Empleados del Gobierno Autónomo Municipal de Huatajata encargados de procesar y aprobar los tramites.

3.1.1.2. Obtención de Requisitos. La obtención de requisitos es fundamental para que un sistema sea exitoso para lo cual se realizaron actividades en el que se muestra en la siguiente tabla.

Tabla 10.*Obtención de requisitos*

ACTIVIDAD	DESCRIPCION
Entrevista	Se llevan a cabo reuniones con todas las partes interesadas para identificar sus necesidades y expectativas con respecto al sistema.
Observación De Procesos Actuales	Se observan y analizan los procesos y procedimientos actuales en el Gobierno Autónomo Municipal para identificar áreas de mejora y requisitos del sistema.
Documentación	Se revisan documentos previos, como formularios, regulaciones y políticas, para entender los procesos y requisitos actuales.

3.1.2. Lista de Requerimientos del Sistema

La claridad y la consistencia en la descripción de los requisitos son fundamentales para comprender y definir el comportamiento del sistema de manera efectiva.

3.1.2.1. Requerimientos funcionales. Los requisitos funcionales se describen a continuación en la siguiente tabla, donde se muestran las características que necesita el sistema a partir de la información obtenida como parte de las tareas de obtención de requisitos.

Tabla 11.*Requerimientos funcionales del sistema*

REQUISITO FUNCIONAL	DESCRIPCION
Administrador de Usuarios	Permite la gestión integral de usuarios del sistema, lo que incluye asignar roles y permisos, crear nuevos usuarios, modificar sus detalles, asignarlos a unidades específicas y también eliminar usuarios según sea necesario
Gestión de Tramites	Permite iniciar, seguir y completar trámites municipales, proporcionando documentación necesaria y recibiendo actualizaciones sobre el progreso.
Acceso Restringido por Roles	Limita el acceso a ciertas funciones y datos del sistema según el rol del usuario (ciudadano, funcionario, administrador, autoridad).
Módulo de Tramites	Permite crear, recibir, despachar y finalizar tramites.
Módulo de Seguimiento	Permite realizar seguimiento de tramites a las personas que tengan una cuenta de usuario en el sistema.
Módulo de Consultas De Tramites	Facilita la comunicación y relación con los involucrados para dar respuesta a las hojas de ruta.
Módulo de reportes pdf	Se generará reporte de la hoja de ruta, así mismo se podrá ver el personal que tiene más cantidad de tramites en su bandeja de lo aceptable.
Oculto	Inicio y cierre cesión.

3.1.2.2. Requerimientos No Funcionales. Los requerimientos no funcionales definen los criterios para evaluar la operación de un servicio de tecnología de la información, a diferencia de los requerimientos funcionales que describen los comportamientos específicos del sistema.

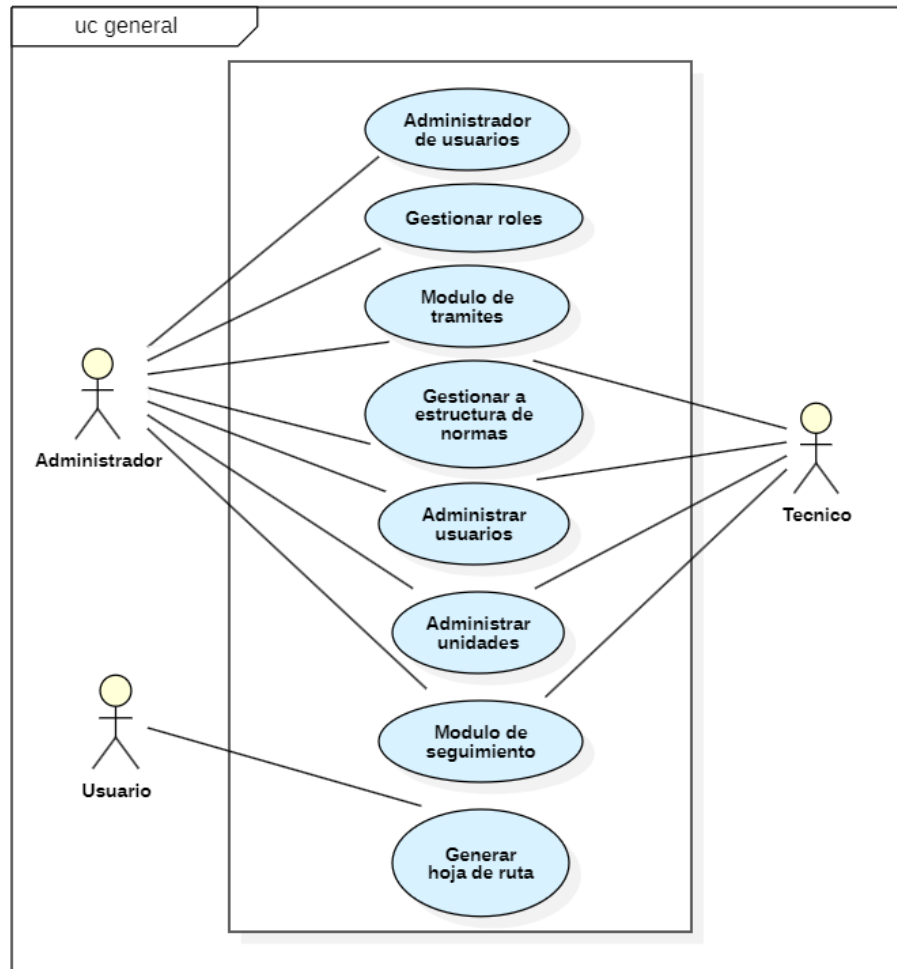
Tabla 12.*Requerimientos no funcionales del sistema*

FUNCIÓN
El sistema debe ser compatible y funcionar correctamente en cualquier navegador web, incluyendo Internet Explorer, Mozilla Firefox, Google Chrome, entre otros.
El sistema debe tener una interfaz gráfica, amigable para que el usuario pueda tener una mejor experiencia.
Se debe implementar un respaldo energético del servidor para garantizar la disponibilidad continua del sistema
Se realizará un soporte y mantenimiento periódico del sistema para garantizar su buen rendimiento a lo largo del tiempo. Esto incluirá actualizaciones de software, parches de seguridad y optimización de recursos según sea necesario.
La sesión tendrá un determinado periodo de tiempo activo.

3.1.2.3. Diagrama de Casos de Uso General. En la siguiente figura se mostrará el diagrama de casos de uso general.

Figura 15.

Diagrama de casos de uso general del sistema



3.1.2.4. Diagramas de casos de uso específicos del sistema

- Diagrama de caso de uso: Gestionar usuarios

Figura 16.

Diagrama de casos de uso - Gestionar usuarios

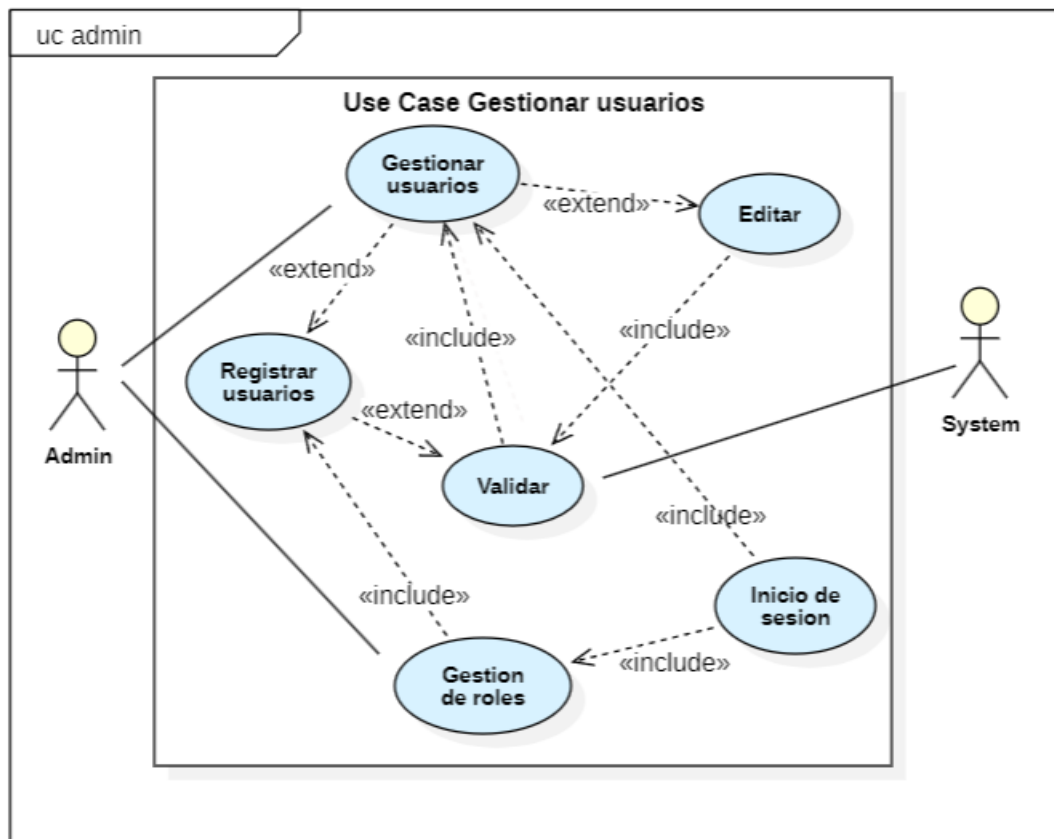


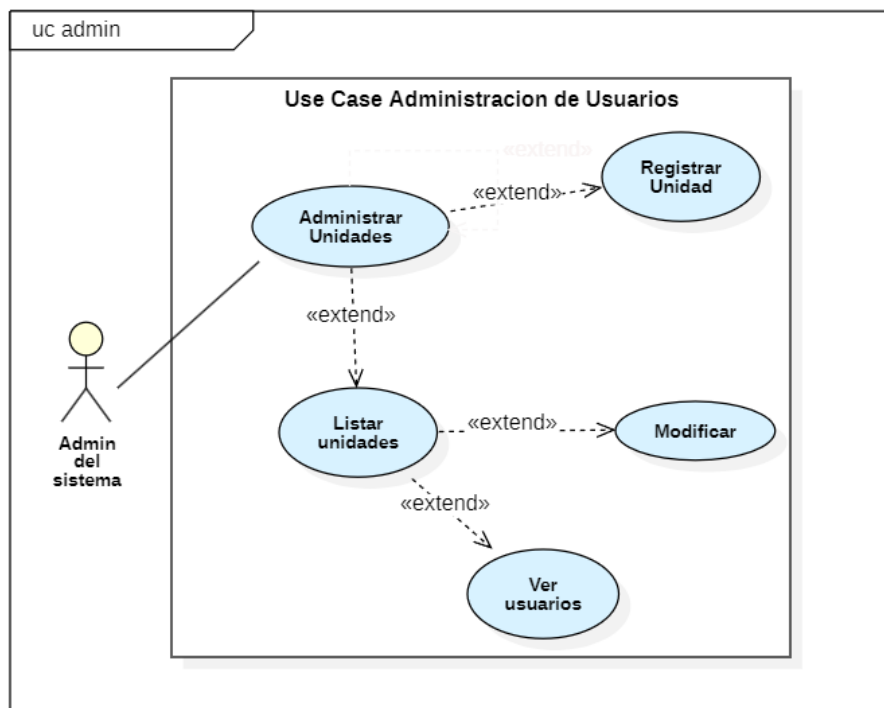
Tabla 13.

Descripción de caso de uso - Gestionar usuarios

Caso de Uso	Gestionar Usuarios
Descripción	El sistema permite gestionar los usuarios registrados en el mismo
Actores	Administrador
Precondiciones	El usuario tiene que estar registrado en el sistema, haber iniciado la sesión con éxito e ingresar al panel de administración de usuarios
Flujo de eventos	<ol style="list-style-type: none"> 1.El sistema despliega una lista de usuarios registrados. 2.El sistema solicita la información para crear, modificar, reset de password, activar o desactivar un usuario. 3.El sistema valida los datos y realiza la acción solicitada.
Flujos alternativos	Si el usuario trata de crear o modificar un usuario con datos ya existentes muestra un mensaje de error "El usuario ya existe"

Figura 17.

Diagrama de casos de uso - Administración de unidades



3.1.2.5. Diagrama de Caso de Uso Específicos – Administración de Unidades

Tabla 14.

Descripción de caso de uso "Administración de unidades"

Caso de Uso	Administración de unidades
Actor	Administrador, Encargado de sistemas
Tipo	Primario
Descripción	El administrador de sistema registra unidades, lista las unidades existentes, modifica la unidad y puede listar los usuarios de la unidad.

3.1.2.6. Diagrama de Caso de Uso Específicos – Administración de Usuarios

Figura 18.

Diagrama de cosos de uso - Administrador de usuarios

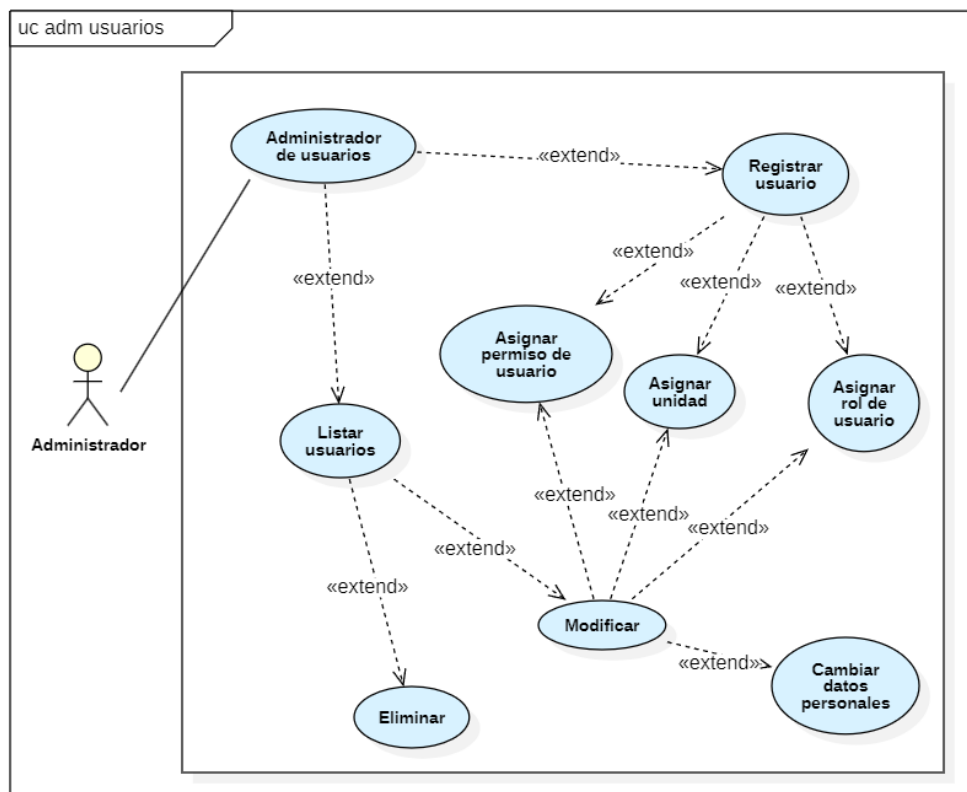


Tabla 15.

Descripción de caso de uso - Administración de usuario

Caso de Uso	Administración de usuarios
Actor	Administrador, Encargado de sistemas
Tipo	Primario
Descripción	El administrador de sistema registra usuarios, existentes, modifica datos personales, asignar permisos, asignar rol, asignar unidad y eliminar usuarios.

3.1.2.7. Diagrama caso de uso hojas de ruta.

Figura 19.

Diagrama de caso de uso hojas de ruta

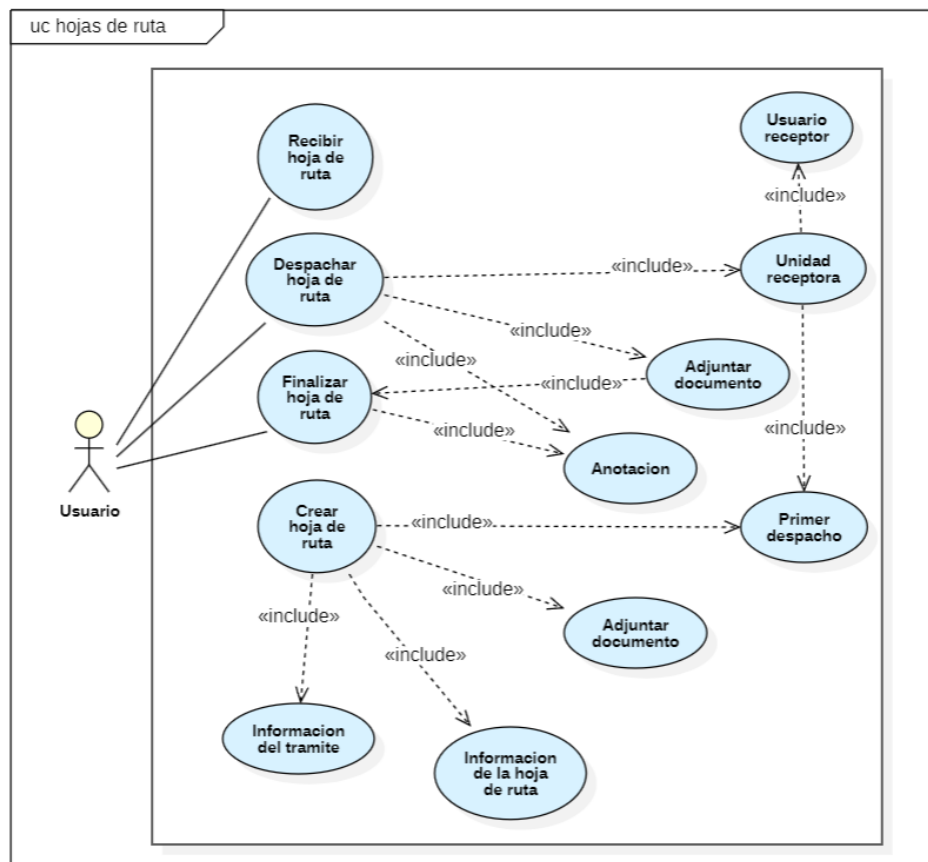


Figura 20.

Diagramas de caso de uso módulo de hoja de ruta

Caso de Uso	Módulo de hojas de ruta
Actor	Usuarios dentro del sistema
Tipo	Primario
Descripción	<p>Todos los usuarios pueden:</p> <ul style="list-style-type: none"> • Recibir hojas de ruta que otros usuarios les hayan enviado • Despachar hojas de ruta a otros usuarios del sistema • Finalizar hoja de ruta • Generar hojas de ruta

3.1.2.8. Diagrama de Caso de Uso Cuenta de Usuario.

Figura 21.

Diagrama de caso de uso cuenta de usuario

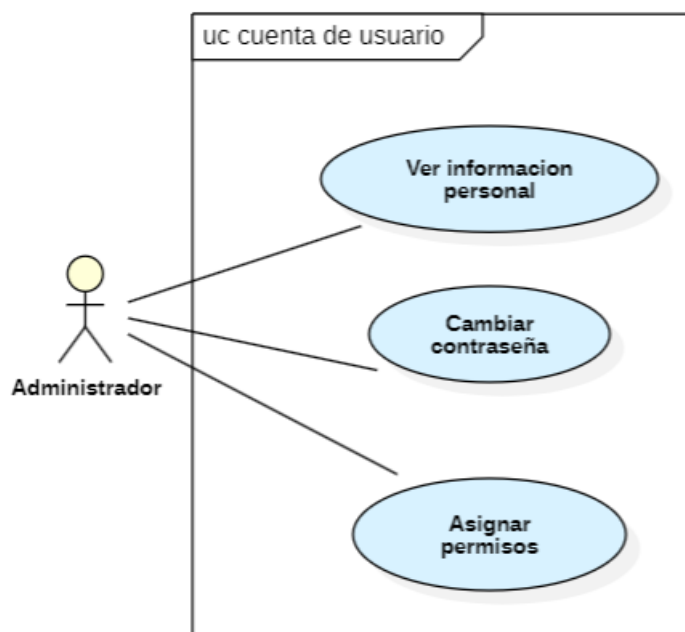


Tabla 16.

Caso de uso cuenta de usuario

Caso de Uso	Módulo de hojas de ruta
Actor	Usuarios dentro del sistema
Tipo	Secundario
Descripción	Todos los usuarios pueden ver su información personal, cambiar su contraseña y cambiar su foto de usuario.

3.1.4. Modelo de Navegación

A continuación, se hace el modelado donde se aprecia la interacción de los usuarios en la navegación del sistema.

Figura 23.

Modelo de navegación - Administrador

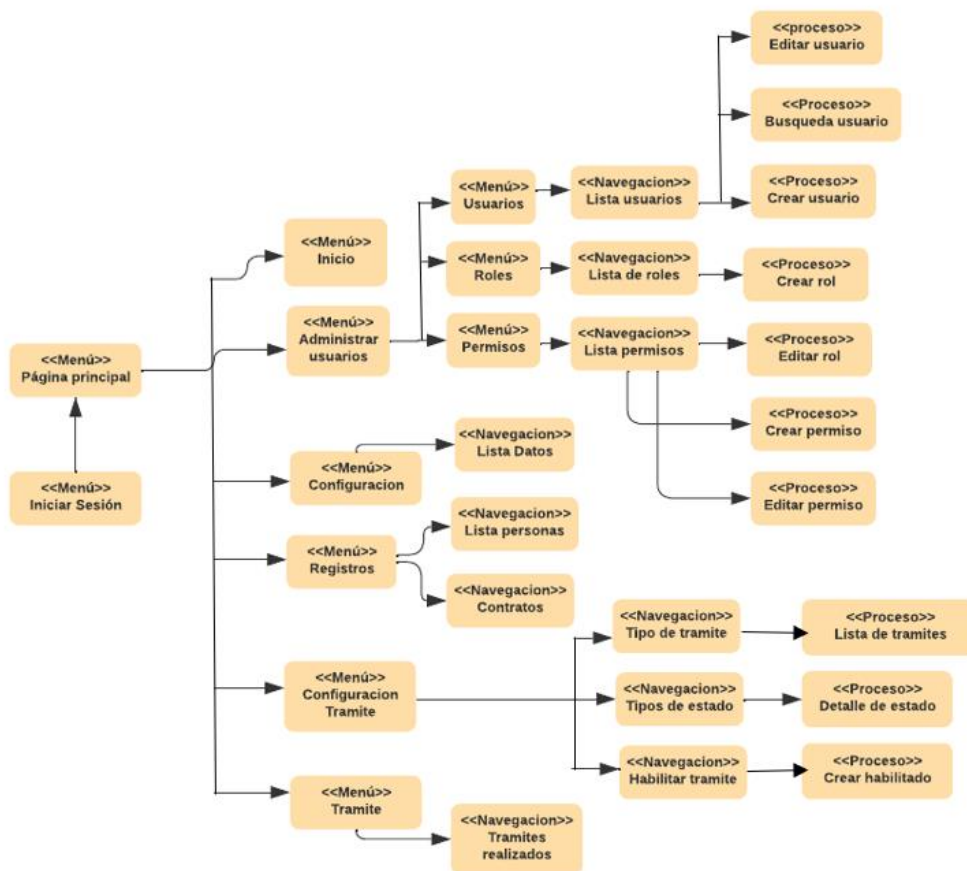


Figura 24.

Modelo de navegación - Técnico

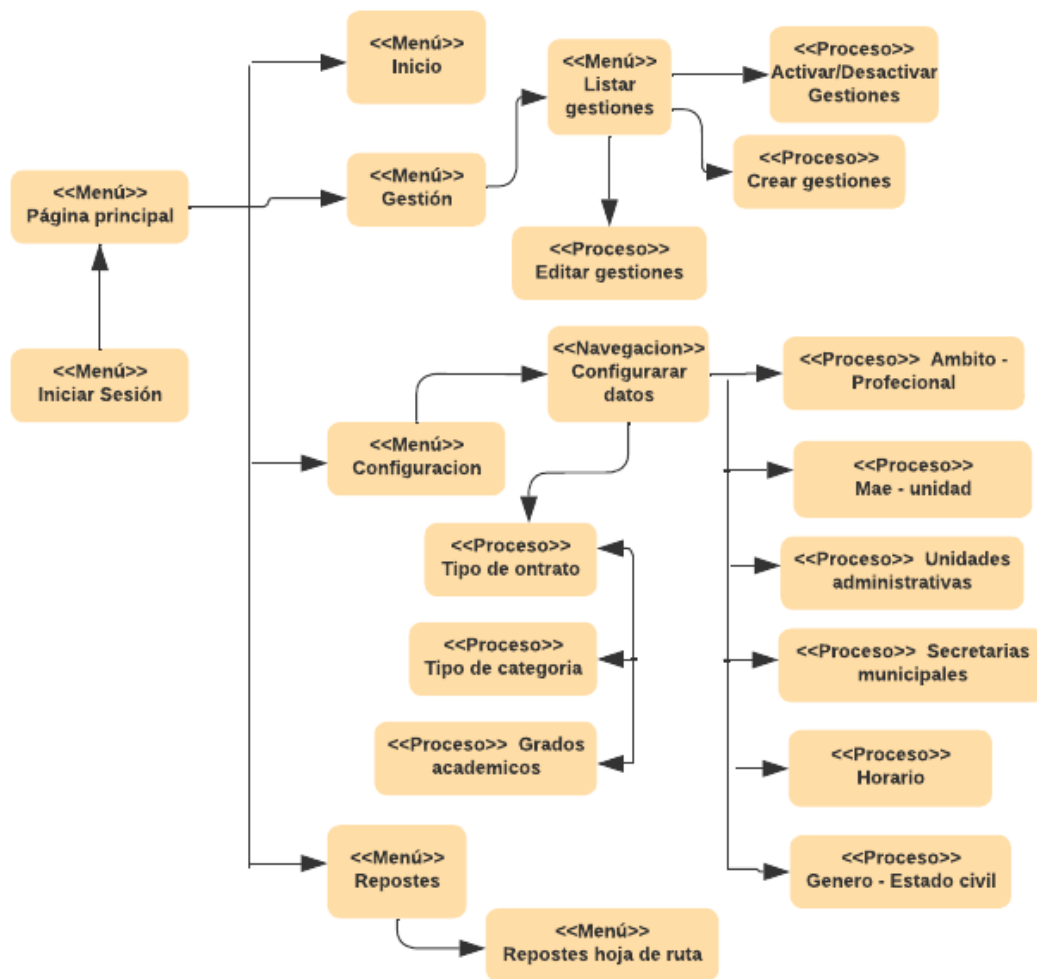
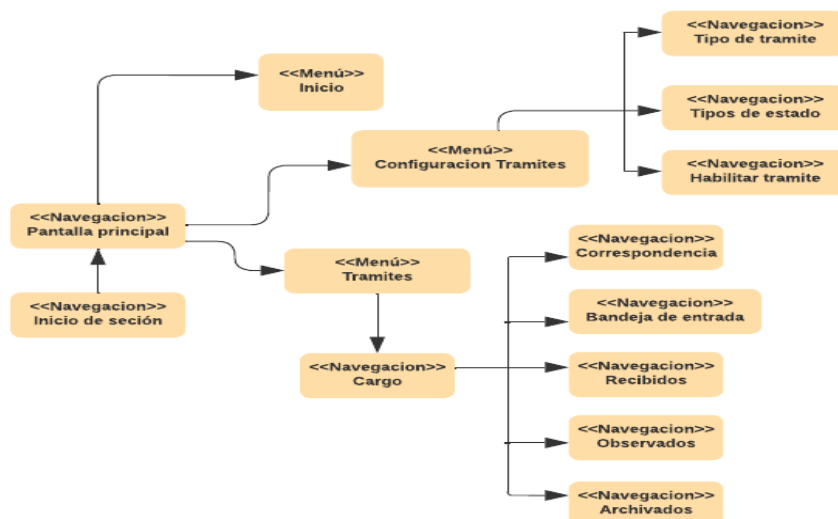


Figura 25.

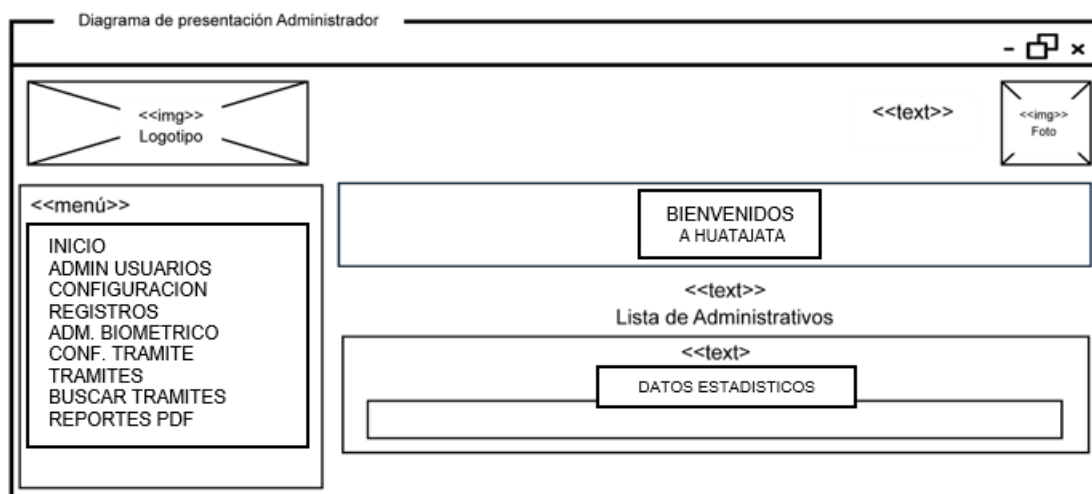
Modelo de navegación - usuario



3.1.5. Modelo de Presentación

Figura 26.

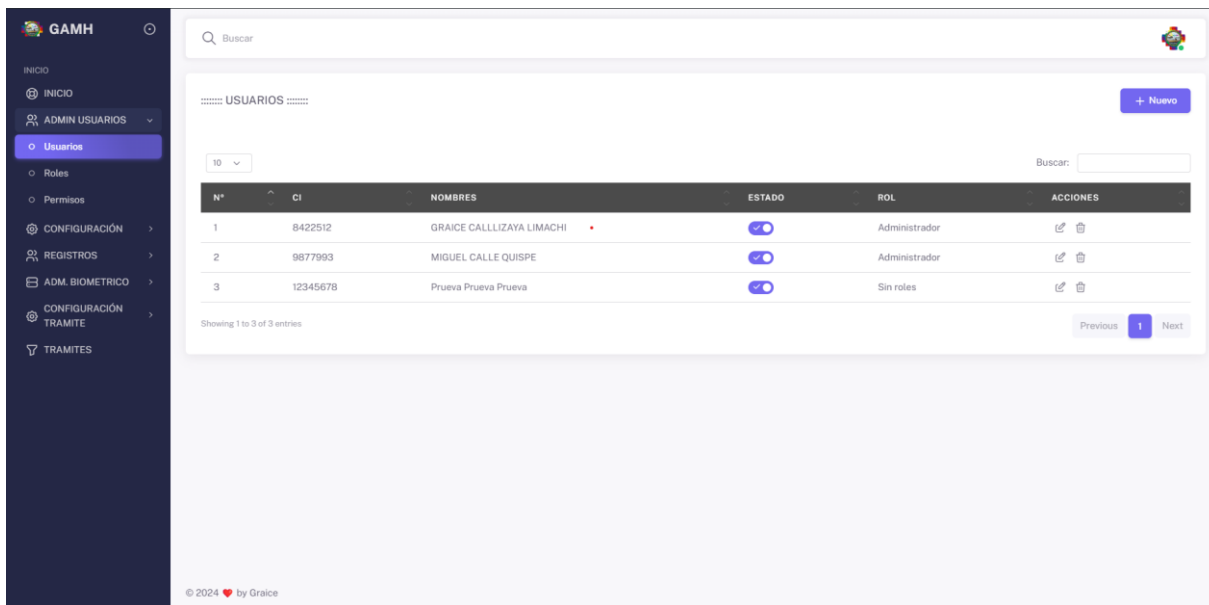
Modelo de presentación



3.1.6. Modelo de Implementación

Figura 27.

Administración de usuarios



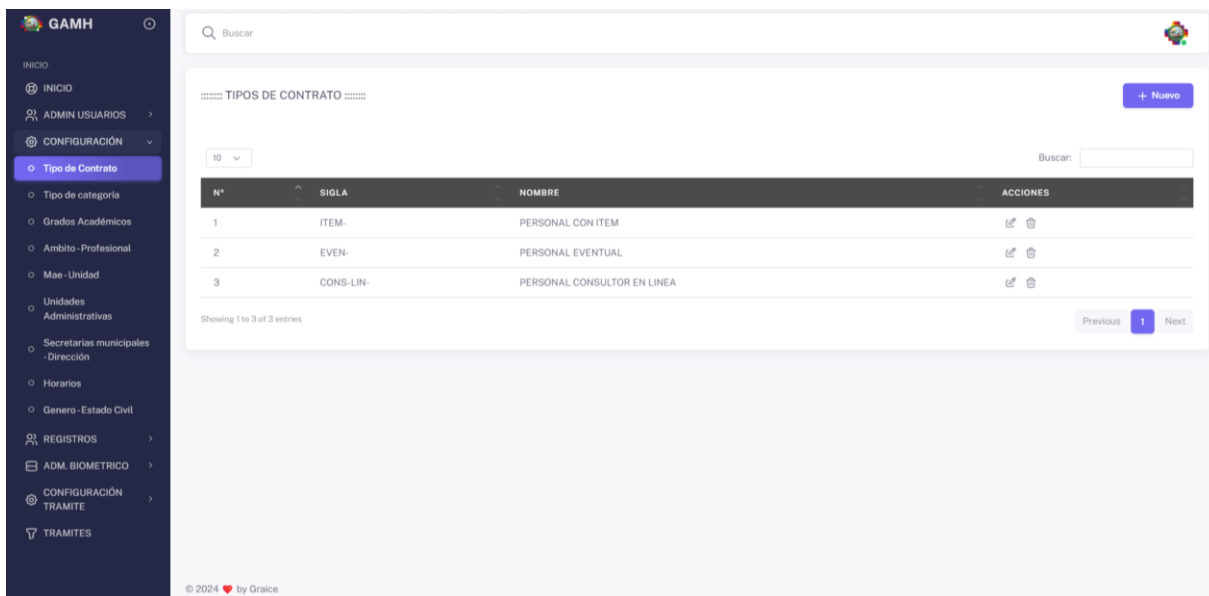
The screenshot displays the 'ADMIN USUARIOS' section of the GAMH system. The left sidebar is active on 'Usuarios'. The main content area shows a table of users with the following data:

N°	CI	NOMBRES	ESTADO	ROL	ACCIONES
1	8422512	GRAICE CALLLIZAYA LIMACHI	<input checked="" type="checkbox"/>	Administrador	✎ ✖
2	9877993	MIGUEL CALLE QUISPE	<input checked="" type="checkbox"/>	Administrador	✎ ✖
3	12345678	Prueba Prueba Prueba	<input checked="" type="checkbox"/>	Sin roles	✎ ✖

Additional interface elements include a search bar, a '+ Nuevo' button, a dropdown menu set to '10', and pagination controls showing 'Showing 1 to 3 of 3 entries'.

Figura 28.

Tipo de contrato



The screenshot displays the 'TIPOS DE CONTRATO' section of the GAMH system. The left sidebar is active on 'Tipo de Contrato'. The main content area shows a table of contract types with the following data:

N°	SIGLA	NOMBRE	ACCIONES
1	ITEM-	PERSONAL CON ITEM	✎ ✖
2	EVEN-	PERSONAL EVENTUAL	✎ ✖
3	CONS-LIN-	PERSONAL CONSULTOR EN LINEA	✎ ✖

Additional interface elements include a search bar, a '+ Nuevo' button, a dropdown menu set to '10', and pagination controls showing 'Showing 1 to 3 of 3 entries'.

Figura 29.

Tipo de categoría

TIPOS DE CATEGORÍA

10

Buscar:

N°	NOMBRE	NIVEL	ACCIONES
1	SUPERIOR	👁	✎ 🗑
2	EJECUTIVO	👁	✎ 🗑
3	OPERATIVO	👁	✎ 🗑

Showing 1 to 3 of 3 entries

Previous 1 Next

© 2024 by Graice

Figura 30.

Grados académicos

GRADOS ACADÉMICOS

10

Buscar:

N°	ABREVIATURA	NOMBRE	ESTADO	ACCIONES
1	SEC.	SECRETARIA	👍	✎ 🗑
2	ARQ.	ARQUITECTO	👍	✎ 🗑
3	TEC. SUP.	TECNICO SUPERIOR	👍	✎ 🗑
4	UNIV.	UNIVERSITARIO	👍	✎ 🗑
5	SRTA.	SEÑORITA	👍	✎ 🗑
6	SRA.	SEÑORA	👍	✎ 🗑
7	SR.	SEÑOR	👍	✎ 🗑
8	TEC.	TÉCNICO MEDIO	👍	✎ 🗑
9	ING.	INGENIERO	👍	✎ 🗑
10	ABG.	ABOGADO	👍	✎ 🗑

Showing 1 to 10 of 11 entries

Previous 1 2 Next

© 2024 by Graice

Figura 31.

Ámbito - profesional

© 2024 by Graice

Figura 32.

Mae - Unidad

N°	NOMBRE	UNIDADES	ACCIONES
1	SECRETARÍA GENERAL		
2	DESPACHO ALCALDE MUNICIPAL		

Showing 1 to 2 of 2 entries

© 2024 by Graice

Figura 33.

Unidades administrativas

UNIDADES ADMINISTRATIVAS

N°	SIGLA	NOMBRE	ESTADO	ACCIONES
1	JSI	JEFATURA DE SERVICIOS INTEGRALES	Activo	
2	URSSM	UNIDAD DE REDES Y SERVICIOS DE SALUD MUNICIPAL	Activo	
3	UADHM	UNIDAD DE ADMINISTRACIÓN DEL HOSPITAL MUNICIPAL	Activo	
4	UOTC	UNIDAD DE ORDENAMIENTO TERRITORIAL Y CATASTRO	Activo	
5	UDU	UNIDAD DE DESARROLLO URBANO	Activo	
6	UDPS	UNIDAD DE PROYECTOS ESPECIALES	Activo	
7	UFDO	UNIDAD DE FISCALIZACIÓN DE OBRAS	Activo	
8	UDGR	UNIDAD DE GESTIÓN DE RIESGOS	Activo	
9	UDC	UNIDAD DE CONTRATACIONES	Activo	
10	RRHH	UNIDAD DE RECURSOS HUMANOS	Activo	

Showing 1 to 10 of 14 entries

© 2024 by Graice

Figura 34.

Secretaría municipal - Dirección

LISTADO DE LAS SECRETARÍAS MUNICIPALES

N°	SIGLA	NOMBRE	ESTADO	DIRECCIONES	ACCIONES
1	SMAF	SECRETARÍA MUNICIPAL ADMINISTRATIVA FINANCIERA	Activo		
2	SMDPMA	SECRETARÍA MUNICIPAL DE DESARROLLO PRODUCTIVO Y MEDIO AMBIENTE	Activo		
3	SMTOP	SECRETARÍA MUNICIPAL TÉCNICA Y DE OBRAS PÚBLICAS	Activo		
4	SMDH	SECRETARÍA MUNICIPAL DE DESARROLLO HUMANO	Activo		

Showing 1 to 4 of 4 entries

© 2024 by Graice

Figura 35.

Horarios

The screenshot shows the 'Horarios' configuration page in the GAMH system. The sidebar menu on the left is expanded to 'Horarios'. The main content area features a search bar at the top right and a '+ Nuevo' button. Below the search bar is a table with the following data:

N°	NOMBRE	DESCRIPCIÓN	ESPECIFICACIÓN HORAS	ACCIONES
1	NORMAL	HORA DE ENTRADA 08:00 SALIDA DE MAÑANA 12:00; ENTRADA TARDE 14:30 Y SALIDA TARDE 18:30	🕒	✏️ 🗑️
2	ESPECIAL CIRECA	HORA DE ENTRADA 08:00 SALIDA DE MAÑANA 12:00; ENTRADA TARDE 14:00 Y SALIDA TARDE 18:00	🕒	✏️ 🗑️

At the bottom of the table, it says 'Showing 1 to 2 of 2 entries' and includes 'Previous' and 'Next' navigation buttons.

Figura 36.

Género - Estado civil

The screenshot shows the 'Género - Estado civil' configuration page in the GAMH system. The sidebar menu on the left is expanded to 'Género - Estado Civil'. The main content area features a search bar at the top right and a '+ Nuevo' button. Below the search bar is a table with the following data:

N°	SIGLA	NOMBRE	ESTADO	ACCIONES
1	M	MASCULINO	🔘	✏️ 🗑️
2	F	FEMENINO	🔘	✏️ 🗑️
3	NS	NO SABE	🔘	✏️ 🗑️
4	PND	PREFIERE NO DECIR	🔘	✏️ 🗑️

At the bottom of the table, it says 'Showing 1 to 4 of 4 entries' and includes 'Previous' and 'Next' navigation buttons.

Figura 37.
Contratos

INFORMACIÓN PERSONAL

Ci: Complemento: NIT:

Fecha de nacimiento: Selección Género: Selección Estado Civil:

Nombres: Apellido Paterno: Apellido Materno:

Gmail: N° de celular: Dirección:

INFORMACIÓN DEL CONTRATO

FECHA DE INICIO: FECHA DE CONCLUSIÓN: SELECCIONE TIPO DE CONTRATO:

NÚMERO DE CONTRATO: HABER BÁSICO: SELECCIONE LA CATEGORÍA:

SELECCIONE EL NIVEL: SELECCIONE EL HORARIO:

Figura 38.
Tipo de tramite

TIPOS DE TRAMITE

10

N°	NOMBRE	SIGLA	ESTADO	ACCIONES
1	TRAMITE INTERNO (TI)	TI	<input checked="" type="checkbox"/>	
2	TRAMITE EXTERNO (TE)	TE	<input checked="" type="checkbox"/>	
3	MEMORANDUMS	ME	<input checked="" type="checkbox"/>	
4	INFORMES	IN	<input checked="" type="checkbox"/>	
5	CIRCULAR	CIR	<input checked="" type="checkbox"/>	
6	HOJA DE PERMISO	HP	<input checked="" type="checkbox"/>	
7	COMUNICACIÓN INTERNA	CI	<input checked="" type="checkbox"/>	
8	CERTIFICACION	CERT	<input checked="" type="checkbox"/>	
9	COURRIER	COU	<input checked="" type="checkbox"/>	
10	FAX	FAX	<input checked="" type="checkbox"/>	

Showing 1 to 10 of 14 entries Previous 1 2 Next

© 2024 by Graice

Figura 39.

Tipos de estado

The screenshot shows the 'Tipos de estado' configuration page in the GAMH application. The sidebar on the left includes options like 'Inicio', 'Admin Usuarios', 'Configuración', 'Registros', 'ADM. BIOMETRICO', and 'Configuración Tramite'. The main content area has a search bar and a table with the following data:

N°	NOMBRE	COLOR	ACCIONES
1	OBSERVADA	OBSERVADA	[Edit] [Delete]
2	ELIMINADO	ELIMINADO	[Edit] [Delete]
3	CONCLUIDA	CONCLUIDA	[Edit] [Delete]
4	RECIBIDA	RECIBIDA	[Edit] [Delete]
5	SIN RECIBIR	SIN RECIBIR	[Edit] [Delete]
6	PENDIENTE	PENDIENTE	[Edit] [Delete]

At the bottom of the table, it says 'Showing 1 to 6 of 6 entries' and includes 'Previous' and 'Next' navigation buttons. The footer shows '© 2024 by Graice'.

Figura 40.

Habilitar tramite

The screenshot shows the 'Habilitar para tramites' configuration page in the GAMH application. The sidebar on the left includes options like 'Inicio', 'Admin Usuarios', 'Configuración', 'Registros', 'ADM. BIOMETRICO', and 'Configuración Tramite'. The main content area has a search bar and a table with the following data:

N°	CI	NOMBRES Y APELLIDOS	CARGOS	ACCIONES
1	8422512	GRAICE CALLIZAYA LIMACHI	ENCARGADO U. DE CONTRATACIONES	Habilitado [Toggle]
2	9877993	MIGUEL CALLE QUISPE	ENCARGADO DE LA U. DE FISCALIZACION	Habilitado [Toggle]
3	12345678	Prueba Prueba Prueba	NO TIENE CARGOS PARA TRAMITES	Habilitar [Toggle] No disponible

At the bottom of the table, it says 'Showing 1 to 3 of 3 entries' and includes 'Previous' and 'Next' navigation buttons. The footer shows '© 2024 by Graice'.

Figura 41.
Buscar tramite

Figura 42.
Reporte tramite

CITE	REFERENCIA	FECHA RECIVIDA	NOMBRE
▶ GAMH/JEF. D ECO./TI	COMRA INVENTARIO	2024-06-19	(JEFE DE PRESUPUESTOS) LIC. MELINA NOEMI ALVAREZ CATARI
▶ GAMH/JEF. PRE./TE	CURSOS TALLERES	2024-06-19	(JEFE DE PRESUPUESTOS) LIC. MELINA NOEMI ALVAREZ CATARI
▶ GAMH/JEF. PRE./TE	REFERENCIA	2024-06-19	(JEFE D ECONTRATACIONES) LIC. GRAICE CALLIZAYA CHAMBI
▶ GAMH/JEF. PRE./TI	REFERENCIA	2024-06-19	(JEFE D ECONTRATACIONES) LIC. GRAICE CALLIZAYA CHAMBI

Showing 1 to 4 of 4 entries

Previous 1 Next

LISTADO DE TRAMITES REALIZADOS POR PERSONA

LIC. MELINA NOEMI ALVAR... 3

CAPÍTULO IV

MÉTRICAS DE

CALIDAD,

SEGURIDAD

Y COSTO

CAPÍTULO IV

4. INTRODUCCIÓN

A continuación, en el siguiente capítulo se muestra las métricas de calidad, costo y seguridad, para sustentar el desarrollo del proyecto de grado en sus diferentes etapas.

4.1. MÉTRICAS DE CALIDAD ISO 25000

En este capítulo se evalúa la calidad del sistema, un aspecto crucial en el desarrollo de software, así como se describe la seguridad del mismo. Se aplicaron métricas de calidad para garantizar el desarrollo adecuado del proyecto, calculando los parámetros que cumple el sistema. Esto permite especificar ordenadamente las características y atributos del sistema.

4.1.1. *Funcionalidad*

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

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

Tabla 17.

Números de entrada de usuarios

ENTRADAS DE USUARIO		N.º
1	Administración de Usuarios	1
2	Administración de Técnicos	1
3	Administración de Usuarios	28
TOTAL		30

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

Tabla 18.*Número de salidas de usuarios*

SALIDA DE USUARIO		N.º
1	Administración de Usuarios	3
2	Administración de Técnicos	3
3	Administración de Usuarios	20
TOTAL		26

- Número de Archivos: Es el número de archivos, se toma en los grupos lógicos de datos, o archivos independientes.
- Número de interfaces externas: Es la suma de todas las interfaces legibles por la máquina.

Tabla 19.*Parámetros de medición de punto de fusión*

PARAMETROS DE MEDICION	CUENTA
Numero de entrada de usuarios	30
Número de salidas de usuario	20
Número de peticiones de usuario	12
Numero de archivos	16
Numero de Interfaz Externa	2

Habiendo realizado el conteo de los parámetros de función en la tabla se hace el cálculo de los puntos de función sin ajustar en la siguiente tabla:

Tabla 20.*Punto de función sin ajustar*

PARAMETRO DE MEDICION	CUENTA	FACTOR	TOTAL
Números de entradas de Usuario	30	6	180
Números de salidas de Usuario	20	7	140
Números de peticiones de Usuario	12	6	72
Números de archivos	16	15	240
Números de interfaz externa	2	10	20
Cuenta total			652

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

Tabla 21.

Factores de complejidad

Importancia	0%	20%	40%	60%	80%	100%
	No influye	Incidental	Moderado	Medio	Significativo	Esencial
Factor	0	1	2	3	4	5
¿Requiere el sistema copias de seguridad y de recuperación fiable?						X 5
¿Se requiere comunicación de datos?					X	4
¿Existe funciones de procesos distribuidos?						X 5
¿Es crítico el rendimiento?					X	4
¿El sistema web será ejecutado el SO es actual?						X 5
¿Se requiere una entrada interactiva para el sistema?					X	4
¿Se requiere que el sistema tenga entradas a datos con múltiples ventanas?					X	4
¿Se actualiza los archivos de forma interactiva?					X	4
¿Son complejas las entradas, salidas, los archivos o las peticiones?						X 5
¿Es complejo el procesamiento interno del sistema?					X	4
¿Se ha diseñado el código para ser reutilizado?						X 5
¿Se ha diseñado el sistema para facilitar al usuario el trabajo y ayudarlos a encontrar la información?						X 5
TOTAL						54

Para el cálculo del punto de función ajustado se utiliza la siguiente formular:

(12). Ecuación de punto de función

$$PF = \text{Conteo Total} [0.65 + 0.01 * \sum fi] \quad (12)$$

Donde:

$$\sum (fi)$$

- Sumatoria de los valores de los factores de ajuste

Se tiene:

$$\sum (fi) = 54$$

$$PF = 652 * [0.65 + 0.01 * 54]$$

$$PF = 775.88$$

Considerando el máximo valor en la sumatoria de los factores de ajuste $\sum(fi) = 70$, se calcula el siguiente valor:

(13). Ecuación de la sumatoria de los factores de ajuste

$$PFmax = Cuenta Total [0.65 + (0.01 * \sum fi)] \quad (13)$$

$$PFmax = 652 * [0.65 + (0.01 * 70)]$$

$$PFmax = 880.2$$

El cálculo de la funcionalidad se obtiene de la relación entre el valor máximo con el valor obtenido del punto de función ajustado:

(14). Ecuación del cálculo de la funcionalidad

$$Funcionalidad = \frac{PF}{PFmax} * 100 \quad (14)$$

$$Funcionalidad = \frac{775.88}{880.2}$$

$$Funcionalidad = 0.8814 * 100\%$$

$$\mathbf{Funcionalidad = 88.14\%}$$

Por lo tanto, la funcionalidad del sistema desarrollado llega a ser del 88.14 %, esto quiere decir que el sistema tiende a funcionar un 88.14 %, sin riesgo de fallar, y un 11.86 % de que el sistema colapse.

4.1.2. Confiabilidad

Es la capacidad de un sistema para mantener su nivel de ejecución bajo condiciones normales en un periodo de prueba establecido, cuyo criterio son la madurez, recuperabilidad, tolerancia a fallos y cumplimiento de fiabilidad Para determinar la confiabilidad del sistema especificamos el tiempo en que empieza a funcionar, a partir de allí se realiza las observaciones hasta un tiempo determinado.

Donde se toma en cuenta que:

$$P(T \leq t) = F(t) \Rightarrow \text{Probabilidad de fallas}$$

$$P(T > t) = 1 - F(t) \Rightarrow \text{Probabilidad de trabajo sin fallas}$$

Para calcular la confiabilidad del sistema se toma en cuenta la siguiente función:

(15). Ecuación de la confiabilidad

$$R(t) = f * e^{-\mu * t} \tag{15}$$

Donde:

- f = funcionalidad del sistema
- μ = Probabilidad de error del sistema
- t = tiempo de prueba del sistema

En un periodo de 20 días como tiempo de prueba se define de cada 10 ejecuciones 1 falla.

$$F(t) = 0.88 * e^{-\frac{1}{10} * 20}$$

$$F(t) = 0.1191 * 100\%$$

$$F(t) = 11.91 \%$$

Reemplazando:

$$P(T \leq t) = 0.1191 = 11.91\%$$

$$P(T \leq t) = 1 - 0.1191$$

$$P(T \leq t) = 0.8809 * 100\%$$

$$P(T \leq t) = \mathbf{88\%}$$

Por lo tanto, la confiabilidad del sistema desarrollado es del 88.09%. en un periodo de 20 días como el tiempo de prueba.

4.1.3. Usabilidad

Para calcular la usabilidad de un sistema se toma en cuenta el factor humano, para conocer si el sistema cumple con los requisitos establecidos para el usuario se realiza la evaluación del sistema mediante encuestas a los usuarios del sistema. Para determinar la usabilidad del sistema, se utiliza la siguiente formular:

Formula:

(16). Ecuación de la usabilidad

$$FU = \left[\frac{\sum x_i}{n} * 100 \right] \quad (16)$$

Donde:

- x_i = Valores de evaluacion de cada pregunta

- $n = \text{Numero de preguntas}$

Se toma en cuenta la siguiente tabla:

Tabla 22.

Parámetros de medición de usabilidad

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

Tabla 23.

Usabilidad del sistema

N	Pregunta	Si	No	Resultado xi
1	¿Puede utilizar con facilidad el sistema?	9	1	0.9
2	¿Puede controlar las operaciones que el sistema solicita?	8	2	0.8
3	¿Las Respuestas del sistema son complicadas?	2	8	0.8
4	¿El Sistema permitió la retroalimentación de información?	9	1	0.9
5	¿El sistema cuenta con interfaz agradable a la vista?	9	1	0.9
6	¿La respuesta del sistema es satisfactoria?	9	1	0.9
7	¿Le parece complicada las funciones del sistema?	8	2	0.8
8	¿Se hace difícil o dificultoso aprender a manejar el sistema?	1	9	0.9
9	¿Los resultados que proporciona el sistema facilitan el trabajo?	10	0	1
10	¿Durante el uso del sistema se produjeron errores?	2	8	0.8
TOTAL				8.7

Con los resultados obtenidos del cuestionario obtenemos calculamos la usabilidad:

$$FU = \left[\frac{8.7}{10} * 100 \right]$$

$$FU = [0.87 * 100]$$

$$FU = 87\%$$

Con esto concluimos la usabilidad del sistema es del 87%.

4.1.4. Eficiencia

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

Tabla 24

Escala de valores de eficiencia

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

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

Tabla 25.

Valoración de la eficiencia del sistema

N ^a	Pregunta	Porcentaje
1	¿La distribución y estilo de la interfaz permite que un usuario introduzca con eficiencia las operaciones y la información?	5
2	¿Una secuencia de operaciones (o entrada de datos) puede realizarse con facilidad de movimientos?	5
3	¿Los datos de salida están presentados de modo que se entienden de inmediato?	4
4	¿Las operaciones jerárquicas están organizadas de manera que minimizan la navegación del usuario para hacer que alguna se ejecute?	5
5	¿Procesa y responde adecuadamente cuando realiza alguna consulta o búsqueda?	5
TOTAL		24

Para calcular la eficiencia teneos la siguiente formula:

(17). *Ecuación de la eficiencia*

$$E = \frac{\sum x_i}{n} * \frac{100}{n} \quad (17)$$

Donde:

$$\sum x_i = \text{Sumatoria de los valores de eficiencia}$$

$n = \text{Numero de preguntas}$

Reemplazando se tiene lo siguiente:

$$E = \frac{22}{5} * \frac{100}{5}$$

$$E = 88\%$$

Con esto concluimos que la eficiencia del sistema es del 88%.

4.1.5. Mantenibilidad

Permite medir el esfuerzo necesario para realizar modificaciones del sistema, ya sea por corrección de errores o por el incremento de módulos o funciones dentro del sistema. El estándar IEE94 sugiere la implementación del índice de madurez del sistema que proporciona un indicador de estabilidad del sistema desarrollado.

De este modo tenemos que la siguiente ecuación para el índice de madurez del sistema:

(18). Ecuación de la mantenibilidad

$$IMS = \frac{[Mt - (Fa + Fc + Fd)]}{Mt} \quad (18)$$

Donde:

- *Mt: Numero de modulos total de la version actual.*
- *Fa: Numero de modulos de la version actual que se añadieron.*
- *Fc: Numero de modulos de la version actual que se cambiaron.*
- *Fd: Numero de modulos de la version anterior que se eliminaron en la*

version actual.

Tabla 26.*Valores para la mantenibilidad*

Descripción	Valor
Mt	5
Fa	0
Fc	1
Fd	0

Reemplazando en la ecuación se tiene:

$$IMS = \frac{[5 - (0 + 1 + 0)]}{5}$$

$$IMS = 0.8 * 100\%$$

$$IMS = 80\%$$

Se concluye que el índice de la mantenibilidad del sistema de desarrollo es del 80%.

4.1.6. Portabilidad

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

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

El sistema está desarrollado para ser implementado en cualquier servidor que tenga instalado los servicios de apache, y las herramientas de PHP y MySql. En el caso del sistema

desarrollado, al estar bajo el marco de trabajo Laravel, se cuenta con la facilidad de configurar el software para casi la mayoría de los ambientes.

En cuanto a la ejecución, al estarse aplicando una tecnología web, el sistema puede ser ejecutado en cualquier dispositivo que cuente con conexión a internet y un navegador web.

Para obtener la portabilidad, se tiene la siguiente formula:

(19). Ecuación de la portabilidad

$$\mathbf{Portabilidad} = 1 - \left(\frac{\text{Numero de dia para portar el sistema}}{\text{Numero de dias para implementar el sistema}} \right) \quad \mathbf{(19)}$$

Reemplazando la formula se obtiene el siguiente resultado:

$$\mathbf{Poratabiliad} = 1 - \left(\frac{1}{7} \right)$$

$$\mathbf{Portabiliad} = 0.85 * 100\%$$

$$\mathbf{Portabilidad} = 86\%$$

Resultados:

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

Tabla 27.
Resultados

Características	Resultados
Funcionalidad	88.14%
Confiabilidad	88.09%
Usabilidad	87%
Eficiencia	88%
Mantenibilidad	80%
Portabilidad	86%
Evaluación total	86.2%

Evaluación de calidad total del sistema es de un 86.2%.

4.2. SEGURIDAD

Uno de los aspectos más importantes a tomar en cuenta en este proyecto, es la implementación de normas de seguridad. La norma ISO 27001 evalúa y rectifica el cumplimiento de las normas, así como la mejora continua en base a un conjunto de controles que permiten reducir el riesgo de sufrir incidentes de seguridad en el funcionamiento del sistema dentro de la institución, para ello se toman en cuenta los siguientes tipos de seguridad:

4.2.1. Seguridad Lógica

Para la seguridad del sistema se consideran las siguientes precauciones:

- **Autenticación:** Para la seguridad de datos del sistema se tienen registrado de nombre de usuario y contraseña de acceso encriptadas, según su nivel de acceso pueda realizar actividades en el sistema. Para asegurar la seguridad de la información dentro del sistema se aplicaron los controles mediante el uso de sesiones y su verificación. En el sistema se tiene 4 niveles de usuario, cada uno con diferentes privilegios. También la implementación del código captcha.
- **Encriptación:** El sistema permite encriptar las rutas y las contraseñas de los usuarios.

- **Base de datos:** En cuanto a la forma de resguardo se realizó los siguientes puntos:
 - Una vez que se accedió a los registros o se realizó las diferentes acciones dentro del sistema se hace la desconexión de la base de datos para que no se tenga ningún problema con terceros.
 - Los respaldos o back-up de la base de datos se realizará de manera automática. Porque la información en una institución es muy valiosa, por tanto, su resguardo es fundamental, la conexión a la base de datos y el cierre de la conexión es de forma automática.

4.2.2. Seguridad Física

- **Seguridad física y del entorno:** Se prevé que los equipos donde se almacenan los datos y el sistema se encuentran resguardados en lugares seguros, con las condiciones necesarias para funcionar con el menor número de riesgos con seguridad de acceso.
- **Equipamiento:** Se tiene una adecuada protección física y mantenimiento de los servidores donde se almacenan los sistemas de la alcaldía, y el completo resguardo de las instalaciones donde se encuentran, como el monitoreo de los servidores.
- **Acceso físico:** Se restringe el acceso físico a las áreas críticas a toda persona no autorizada, para reducir el riesgo de accidentes fraudulentos y robos de contraseña.
 - Cuentan con sistemas de vigilancia por cámara de seguridad
 - Restricción del acceso del personal autorizado.

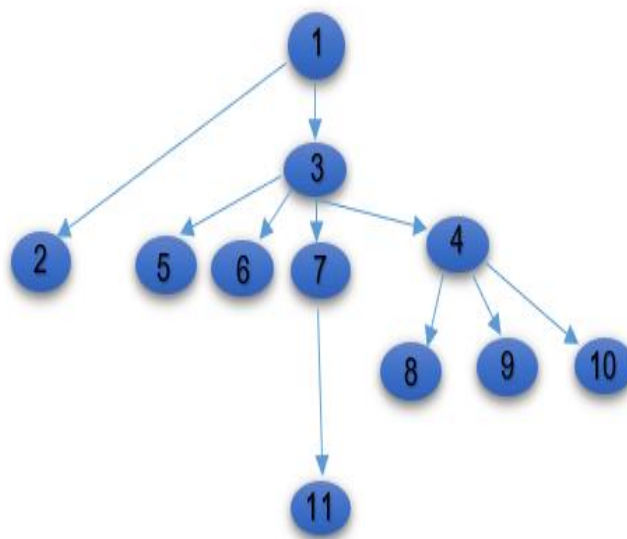
4.3. PRUEBAS

4.3.1. Pruebas de caja blanca

Esta prueba se basa en el cálculo de las regiones que deben ser consideradas como caminos independientes del sistema, y estableciendo cuales con las entradas para que se ejecuten cada una de las regiones. A continuación de forma general se muestra los caminos independientes del sistema.

Tabla 28.

Caja blanca



Donde:

1. Inicio del Sistema
2. Seguimiento de tramites
3. Página principal de usuario
4. Administración de tramites
5. Cuenta de usuario
6. Administrar usuario
7. Administrar unidades

8. Creación de tramites
9. Correspondencia de tramites
10. Reporte de tramite
11. Fin del sistema

Después de realizar el gráfico se procede a determinar la complejidad ciclomática del grafo mediante la siguiente fórmula:

(20). Ecuación de la complejidad ciclomática del grafo

$$V(G) = A - N + 2 \quad (20)$$

Donde:

$$A = 16(\text{Aristas})$$

$$N = 11(\text{Nodos})$$

Por tanto:

$$V(G) = 16 - 11 + 2$$

$$V(G) = 7$$

Determinar los caminos linealmente independientes los caminos que deben ser probados.

Camino 1: 1-2-11

Camino 2: 1-3-7-11

Camino 3: 1-3-6-11

Camino 4: 1-3-5-11

Camino 5: 1-3-4-10-11

Camino 6: 1-3-4-9-11

Camino 7: 1-3-4-8-11

Presentar los casos de prueba para forzar ejecución de cada camino. Esta última condición establece que para la ejecución de ciertos caminos.

- Camino 1: El sistema recibe los datos necesarios para consultar el estado de un trámite y muestra su seguimiento y estado actual.
- Camino 2: Se presenta un listado de unidades existentes en el sistema, así como la opción de crear nuevas unidades si es necesario.
- Camino 3: Se muestra un listado de usuarios existentes en el sistema, junto con la opción de crear nuevos usuarios según sea necesario.
- Camino 4: Se visualizan los permisos del usuario, permitiendo la opción de cambiar su foto de perfil y de actualizar su contraseña si es requerido.
- Camino 5: Se ejecutan los reportes relacionados con los trámites de la unidad según la solicitud recibida.
- Camino 6: Se presenta un listado completo de todos los trámites que han sido registrados, ya sea ingresando o saliendo de la unidad.
- Camino 7: Se despliega un formulario que permite registrar un nuevo trámite en el sistema.

4.3.2. Pruebas de Caja Negra

La prueba de Caja Negra, es una técnica de pruebas de software en la cual la funcionalidad se verifica sin tomar en cuenta la estructura interna de código, detalles de implementación o escenarios de ejecución internos en el software.

Para realizarla prueba de caja negra se tomará en cuenta los casos de ingreso al sistema, registro de trámites y seguimiento de trámites.

Tabla 29.

Caso de prueba "ingreso al sistema"

Caso de prueba ingreso al sistema	
Descripción de la prueba	Ingresar al sistema mediante la autenticación del usuario y contraseña, tras completar la validación necesaria.
Condición de ejecución	El usuario debe encontrarse habilitado previamente
Entradas	El sistema muestra la pantalla de autenticación, donde pide usuario y contraseña para ingresar.
Resultados esperados	Después de validar los datos ingresados, el sistema autoriza el acceso del usuario en función de su rol asignado.
Resultados obtenidos	Se realiza el ingreso al sistema con la verificación previa, en caso de que la verificación falle, el usuario es devuelto a la página de inicio.

1.1.1. Prueba de estrés

Las pruebas de estrés se enfocan en someter la plataforma a situaciones que excedan su capacidad normal de procesamiento, con el propósito de identificar posibles debilidades, errores o fallos en el sistema. Estas pruebas son fundamentales para determinar si la plataforma puede manejar volúmenes de usuarios concurrentes, cargas de trabajo intensas o picos repentinos de tráfico.

Para esta prueba se utilizó un servidor con algunas variaciones en el Sistema Operativo donde se obtuvieron resultados que van con el rendimiento óptimo que exigen a la plataforma.

- Usuarios establecidos: 50
- Tiempo de prueba: 5 min
- Solicitudes realizadas: 987
- Fallas HTTP: 0
- Pico de solicitud: 8 (Solicitudes por segundo)
- Tiempo de respuesta media: 87ms.

4.4. ESTIMACION DE COSTOS – COCOMO II

Para realizar el presente proyecto es necesario planificar y estimar los costos durante y hasta la finalización del mismo. Es útil para estimar el costo total del sistema de registro se tomarán en cuenta los siguientes costos:

Para calcular el esfuerzo, necesitaremos hallar la variable KLDC (kilo-líneas de código). Este proyecto se implementa 5769 Líneas de Código en el lenguaje PHP. Aplicando Conversiones se tiene.

$$LCD = 5769$$

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

$$KLCD = \frac{5769}{1000}$$

$$KLCD = 5.769$$

Tabla 30.

Coefficientes a, b, c, d, de Cocomo II

Proyecto de software	A	B	C	D
Orgánico	3.2	1.05	2.5	0.38
Semiacoplado	3.0	1.12	2.5	0.35
Empotrado	2.8	1.20	2.5	0.32

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

Tabla 31.

Ecuaciones de Cocomo II

Variable	Ecuación	Tipo / Unidad
Esfuerzo requerido por el proyecto	$E = a * (KLDC)^B * FAE$	Personas / mes
Tiempo requerido por el proyecto	$T = c * (E)^d$	Mes
Número de personas requeridas para el proyecto	$NP = \frac{E}{T}$	Personas
Costo Total	$CT = SueldoMes * NP * T$	\$us

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

Tabla 32.

Calculo de atributos FAE

Atributos	Valor					
	Muy bajo	Bajo	Nominal	Alto	Muy Alto	Extra Alto
Atributos de software						
Fiabilidad	0,75	0,88	1,00	1,15	1,40	
Tamaño de Base de Datos		0,94	1,00	1,08	1,16	
Complejidad	0,70	0,85	1,00	1,15	1,30	1,65
Atributos de Hardware						
Restricciones de Tiempo de ejecución			1,00	1,11	1,30	1,66
Restricciones de memoria virtual			1,00	1,06	1,21	1,56
Volatilidad de la máquina virtual		0,87	1,00	1,15	1,30	
Tiempo de respuesta		0,87	1,00	1,07	1,15	
Atributos de personal						
Capacidad de análisis	1,46	1,19	1,00	0,86	0,71	
Experiencia en la aplicación	1,29	1,13	1,00	0,91	0,82	
Calidad de los programadores	1,42	1,17	1,00	0,86	0,70	
Experiencia en la máquina virtual	1,21	1,10	1,00	0,90		
Experiencia en el lenguaje	1,14	1,07	1,00	0,95		
Atributos del proyecto						
Técnicas actualizadas de programación	1,24	1,10	1,00	0,91	0,82	
Utilización de herramientas de software	1,24	1,10	1,00	0,91	0,83	
Restricciones de tiempo de desarrollo	1,22	1,08	1,00	1,04	1,10	

Por tanto, nuestro factor de ajuste será:

$$FAE = 1.00 * 1.08 * 1.30 * 1.00 * 1.06 * 1.15 * 1.00 * 0.86 * 0.91 * 1.00 * 0.90 * 0.95 * 0.82$$

$$* 0.91 * 1.04 = 0.8887$$

Aplicando y reemplazando valores a la fórmula de esfuerzo, se tiene:

$$E = a * KLCD^b * FAE(persona/mes)$$

$$E = 3.2 * 5.769^{1.05} * 0.89(persona/mes)$$

$$E = 17.44(persona/mes)$$

Calculo del tiempo:

$$T = c * Esfuerzo^d(meses)$$

$$T = 2.5 * 17^{0.38}(meses)$$

$$T = 7(\text{meses})$$

Calculo del personal requerido:

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

$$NP = \frac{17}{7} \text{ persona}$$

$$NP = 2.42 \cong 2 \text{ Personas}$$

Calcula el costo total:

$$CT = \text{SueldoMes} * NP * T$$

$$CT = 350 * 2 * 7$$

$$CT = 4900 \text{ \$us}$$

En resumen, se requiere 2 personas estimando un trabajo de 7 meses y con costo total de 4900 \$us, Equivalente en bolivianos a 34000 Bs.

Tabla 33.

Resumen de costo

Esfuerzo	17 personas
Tiempo	7 meses
Cantidad de personas por mes	2 personas
Costo Total	34006

Nota. Es decir que es costo estimado del proyecto es 34006.

CAPÍTULO V

CONCLUSIONES Y

RECOMENDACIONES

CAPÍTULO V

5. CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

Finalmente se detalla las conclusiones alcanzadas tras el desarrollo y despliegue del "Sistema Web de Gestión y Control para el Seguimiento de Hojas de Ruta" para el Gobierno Autónomo Municipal de Huatajata. Este sistema se diseñó para cumplir con los requerimientos específicos de la institución y proporcionar acceso eficiente a información crucial sobre hojas de ruta.

- Se logró diseñar una interfaz amigable y de fácil manipulación para el usuario
- El sistema agiliza los tiempos de acceso a la información.
- Se logró un control sobre el registró y manejo de hojas de ruta.
- Se logró realizar el manual de usuario correspondiente al sistema.
- Se logró disminuir los tiempos en generar reportes.

El desarrollo y despliegue del Sistema Web ha permitido alcanzar el objetivo principal de proporcionar acceso a información confiable y oportuna sobre el estado y avance de las hojas de ruta. Además, ha facilitado al personal administrativo la tarea de manejar y registrar dichas hojas de ruta de manera eficiente.

5.2. RECOMENDACIONES

A partir del presente trabajo se propone las siguientes recomendaciones, con el objetivo de optimizar el sistema:

- Se recomienda capacitar sobre el sistema a los diferentes funcionarios.
- Se recomienda delegar Hojas de ruta simples por completo al sistema y dejar la documentación física.

- Se recomienda hacer el mantenimiento del software en determinados periodos de tiempo tomando en cuenta las normas preestablecidas.

BIBLIOGRAFÍA

BIBLIOGRAFÍA

- 7Graus. (27 de 12 de 2023). *Significado, Equipo*. Obtenido de Significado, Equipo:
<https://www.significados.com/control/>
- Adrian, Y. (12 de 08 de 2023). *ConceptoDefinición*. Obtenido de
<https://conceptodefinicion.de/seguimiento/>
- Alegsa, L. (2016). Obtenido de <https://www.alegsa.com.ar/Dic/informacion.php>
- Alegsa, L. (2016). <https://www.alegsa.com.ar/Dic/informacion.php>. Obtenido de
<https://www.alegsa.com.ar/Dic/informacion.php>
- Alvarez, S. C. (2021). Implementacion ISO 27001.
- Andres, U. M. (2017). *repositorio institucional*. Obtenido de repositorio institucional:
<http://repositorio.umsa.bo/xmlui/handle/123456789/16349>
- Andres, U. M. (2020). *repositorios latinoamericanos* . Obtenido de
<http://repositorio.umsa.bo/xmlui/handle/123456789/7875>
- Bembibre, C. (Abril de 2010). *Significado.com*. Obtenido de <https://significado.com/hoja-de-ruta/>
- Cabaleiro, D. (13 de Octubre de 2015). *Linkedin*. Obtenido de
<https://es.linkedin.com/pulse/definici%C3%B3n-de-mantenimiento-ideas-para-la-mejora-y-cabaleiro-sab%C3%ADn>
- Carlemany. (22 de Marzo de 2024). *Universitat Carlemany*. Obtenido de
<https://www.universitatcarlemany.com/actualidad/blog/metodologias-de-desarrollo-de-software>
- Composer, C. (23 de Diciembre de 2022). *Cursosdesarrolloweb*. Obtenido de
<https://www.cursosdesarrolloweb.es/blog/composer-gestor-dependencias-php>
- Danielthewolf1993. (25 de Junio de 2015). *Metodologia UWE*. Obtenido de Metodologia UWE:
<https://metodologiauwe.wordpress.com/2015/06/25/hello-world/>
- Danielthewolf1993. (25 de Junio de 2015). *Metodologia UWE*. Obtenido de
<https://metodologiauwe.wordpress.com/2015/06/25/hello-world/>
- Deyimar. (28 de Agosto de 2020). *Hostinger*. Obtenido de
<https://www.hostinger.es/tutoriales/que-es-bootstrap>
- Ecured. (2019). *Ecured*. Obtenido de
https://www.ecured.cu/Metricas_para_la_calidad_del_software
- estres, P. d. (08 de Julio de 2022). *Testing iT*. Obtenido de Testing iT:
<https://www.testingit.com.mx/blog/pruebas-de-estres-de-software>

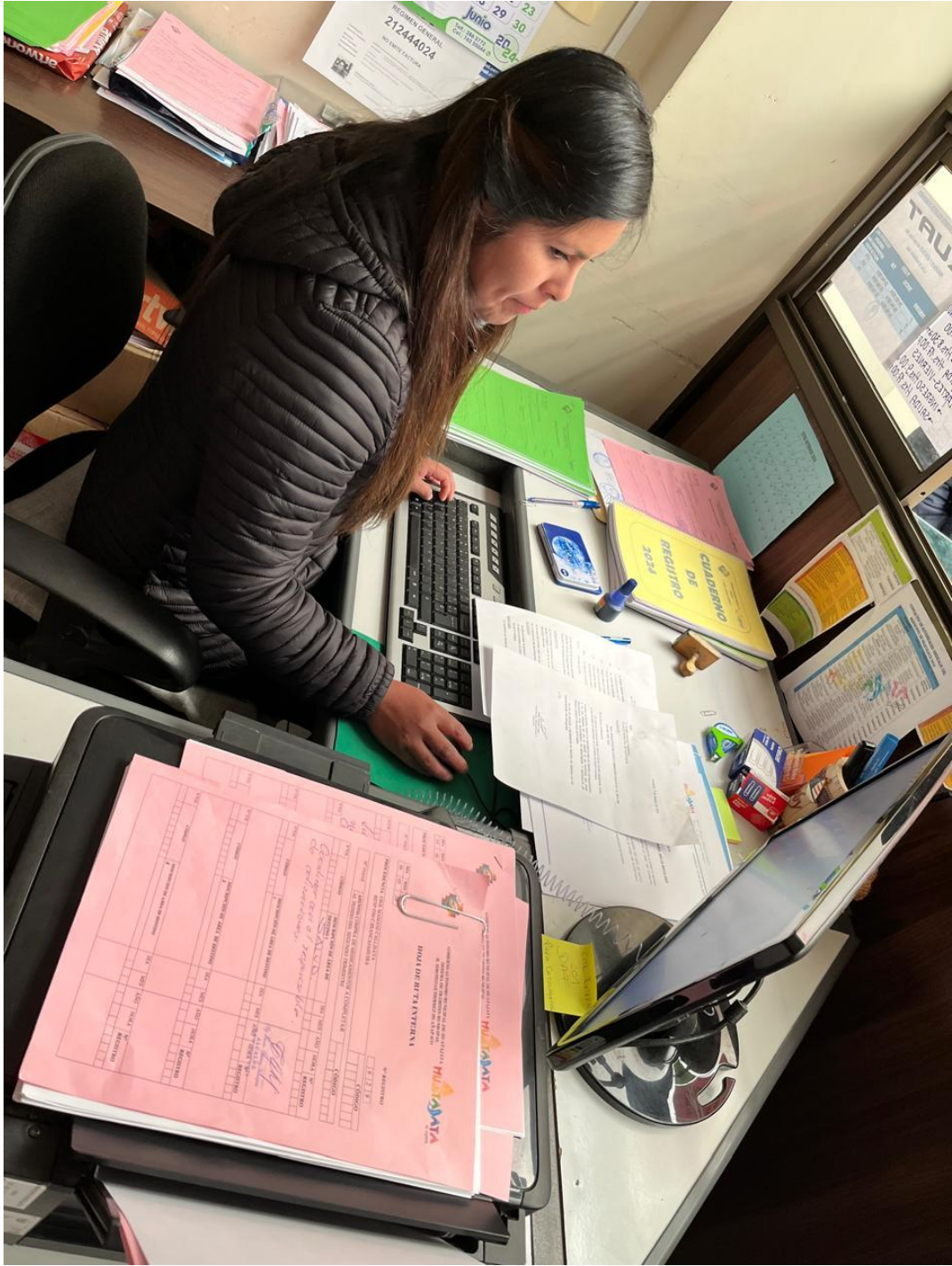
- Etecè. (22 de 10 de 2021). Obtenido de <https://concepto.de/sistema/#ixzz8W0WCcgCq>
- Flores, F. (22 de Julio de 2022). *OpenWebinars*. Obtenido de <https://openwebinars.net/blog/que-es-visual-studio-code-y-que-ventajas-ofrece/>
- Gauchat, J. D. (2012). *El gran libro de HTML5 CSS3 y JavaScript*.
- GlobalSuite ISO 27001, S. (28 de Septiembre de 2023). *GlobalSuite*. Obtenido de <https://www.globalsuitesolutions.com/es/que-son-normas-iso>
- Gomez, A. (2021). UN MODELO DE ESTIMACION DE PROYECTOS DE SOFTWARE. *blogadmin1*, 68.
- Gracia. (02 de Julio de 2012). *Modelos de estimación*. Obtenido de Modelos de estimación: <https://unpocodejava.com/2012/02/07/modelos-de-estimacion-un-poco-sobre-cocomo-ii/>
- IBM, j. e. (2021). *IBM*. Obtenido de <https://www.ibm.com/mx-es/topics/postgresql>
- IONS, b. d. (12 de abril de 2024). *IONS Digital Guide*. Obtenido de <https://www.ionos.es/digitalguide/hosting/cuestiones-tecnicas/bases-de-datos/>
- ISO 25012, N. (2024). Obtenido de <https://iso25000.com/index.php/normas-iso-25000/iso-25012>
- ISO, N. (2024). *Normas ISO 25000*. Obtenido de <https://iso25000.com/index.php/normas-iso-25000>
- ISO/IEC 2502n, D. d. (2024). *Norma ISO/IEC 2502n*. Obtenido de <https://iso25000.com/index.php/normas-iso-25000/8-iso-iec-2502n>
- ISO/IEC 2503n, D. d. (2024). *Normas ISO/IEC 2503n*. Obtenido de <https://iso25000.com/index.php/normas-iso-25000/10-iso-iec-2503n>
- ISO/IEC 2504n, D. d. (2024). *Normas ISO/IEC 2504n*. Obtenido de <https://iso25000.com/index.php/normas-iso-25000/11-iso-iec-2504n>
- Madrid, U. E. (2020). *repositorio academico upc*. Obtenido de repositorio academico upc: <https://repositorioacademico.upc.edu.pe/handle/10757/648599>
- Matich, D. J. (Marzo de 2001). Redes Neuronales: Conceptos Basicos y Aplicaciones. *Revista de Universidad Tecnologica Nacional - Facultad Regional Rosario*, 6-7.
- MDN CSS, j. e. (02 de Agosto de 2023). *Mmdn web docs*. Obtenido de https://developer.mozilla.org/es/docs/Learn/CSS/First_steps/What_is_CSS
- MDN, D. w. (18 de Julio de 2023). *Mmdn wed docs*. Obtenido de https://developer.mozilla.org/es/docs/Learn/Getting_started_with_the_web/JavaScript_basics
- Mmdn HTML, d. (24 de Julio de 2023). *Mmdn web docs*. Obtenido de

- <https://developer.mozilla.org/es/docs/Web/HTML>
- Navicat. (2024). *Navicat Premium*. Obtenido de Navicat Premium:
<https://www.navicat.com/es/products/navicat-premium?igr=intercambios3d>
- Navicat Shop, S. (2024). *Software Shop*. Obtenido de <https://www.software-shop.com/producto/navicat>
- NORMAS ISO 25000. (2022). *Iso25000.com*. Obtenido de <https://iso25000.com/index.php/normas-iso-25000>
- Peño, P. d. (2015).
https://oa.upm.es/40012/1/PFC_JOSE_MANUEL_SANCHEZ_PENO_3.pdf.
- PostgreSQL, Q. e. (14 de Abril de 2024). *IBM*. Obtenido de IBM: <https://www.ibm.com/mx-es/topics/postgresql>
- Pressman, R. S. (2010). *Ingeniería del Software Un Enfoque Practico*. México, D. F.: McGRAW-HILL INTERAMERICANA EDITORES, S.A.
- Rica, U. N. (2017). *una*. Obtenido de una: <http://hdl.handle.net/11056/14151>
- Rosas Guevara, S. J., & Estrada Correa, J. E. (2018). *DISEÑO DE UN SISTEMA DE GESTION DOCUMENTAL Y PROTOTIPO DEL MODULO DE MIGRACION*.
- S.L.U., I. C. (12 de Julio de 2021). *Dgital Guide Ions*. Obtenido de Dgital Guide Ions:
<https://www.ionos.es/digitalguide/paginas-web/disenio-web/que-es-css/>
- Singureanu, P. d. (2022). *Zaptest*. Obtenido de <https://www.zaptest.com/es/pruebas-de-estres-en-pruebas-de-software-que-es-tipos-procesos-enfoques-herramientas-mas>
- System. (22 de Septiembre de 2022). *CreaSystem*. Obtenido de <https://www.creasystem.net/posts/que-es-un-sistema-web>
- System, B. G. (28 de Enero de 2020). *Armetrics*. Obtenido de Armetrics:
<https://www.armetrics.com/glosario-digital/bootstrap>
- Talently, ¿. e. (26 de Julio de 2022). *Talently*. Obtenido de <https://talently.tech/blog/que-es-laravel/>
- upea, r. (2020). *repositorioupea.bo*. Obtenido de [repositorioupea.bo](http://repositorio.upea.bo/jspui/handle/123456789/157):
<http://repositorio.upea.bo/jspui/handle/123456789/157>
- Vaswani, V. (2019). *Fundamentos PHP*.
- Vistatec. (2024). *Vistatec*. Obtenido de <https://vistatec.com/es/services/engineering-and-testing/functionality-testing/>
- Westreicher. (14 de febrero de 2020). *Economipedia.com*. Obtenido de <https://economipedia.com/definiciones/gestion.html>

ANEXOS

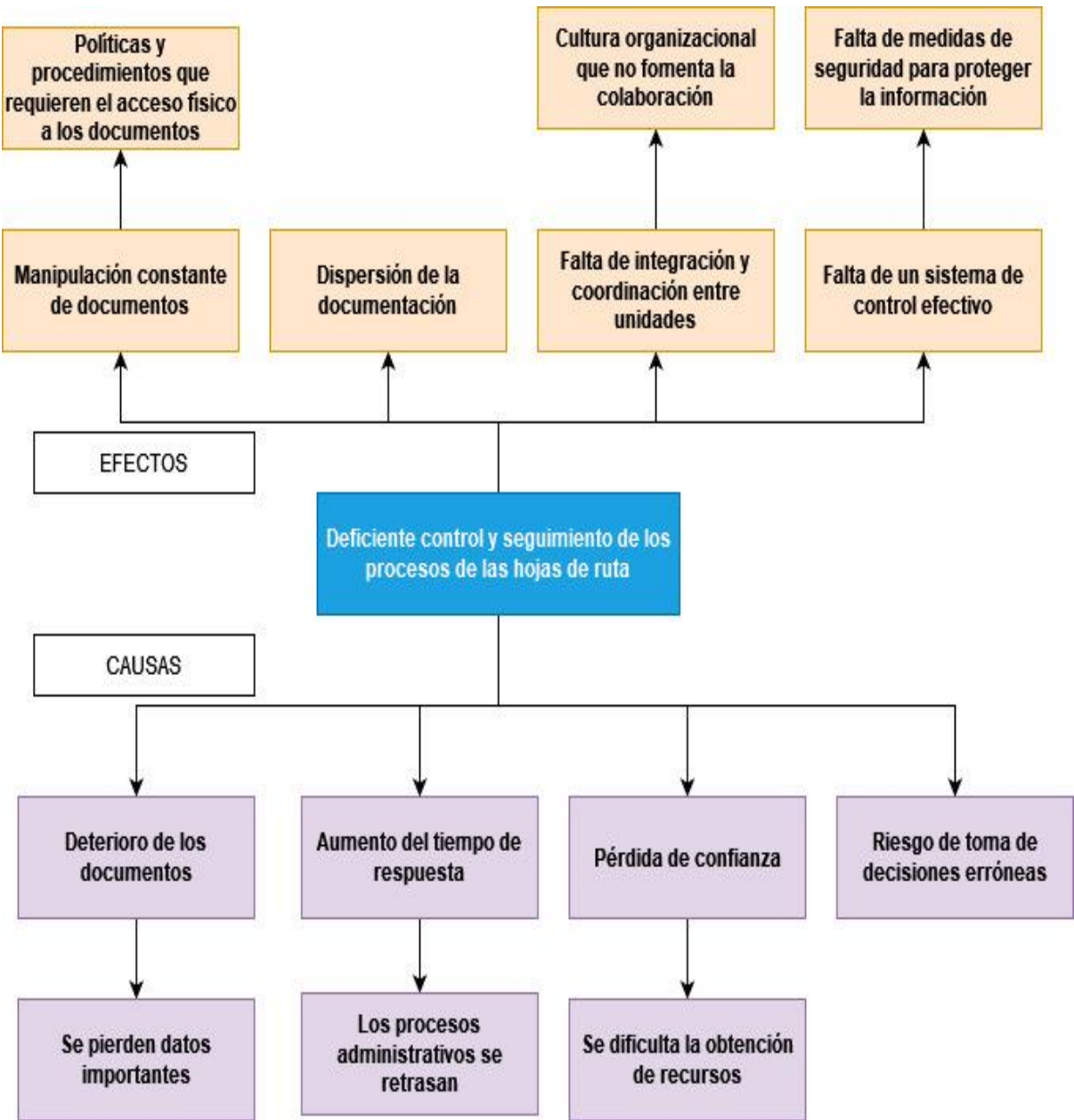
ANEXOS A



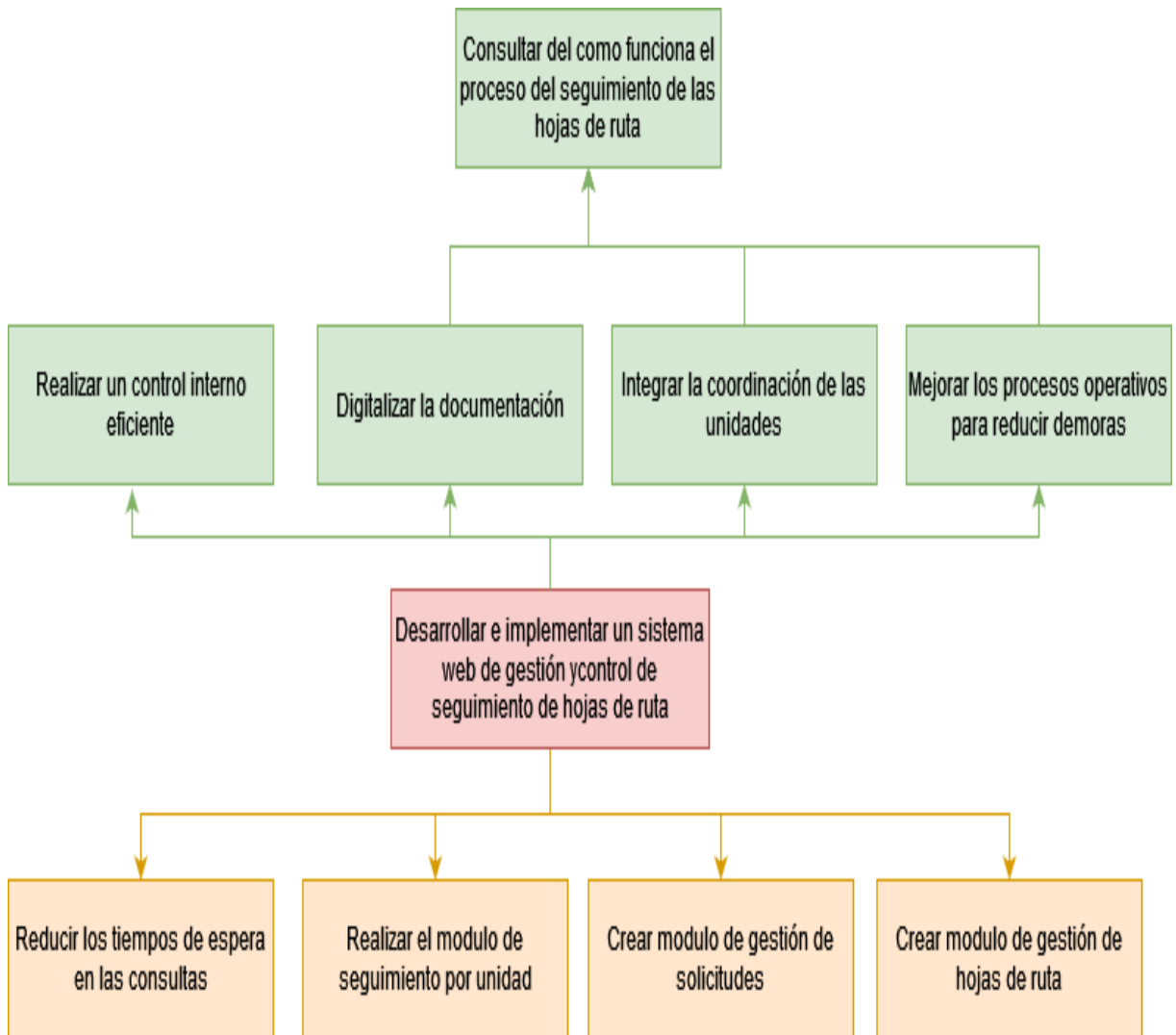


ANEXOS B

ÁRBOL DE PROBLEMAS



ÁRBOL DE OBJETIVOS



ANEXOS C



MANUAL DE ADMINISTRACIÓN

Versión 1.0

**“SISTEMA WEB DE GESTIÓN
Y CONTROL PARA EL
SEGUIMIENTO
DE HOJAS DE RUTA”**

GESTIÓN 2024

MANUAL DE ADMINISTRACIÓN

✓ INTRODUCCION

El presente manual de administración pretende servir de guía para el uso adecuado del sistema web de gestión y control para el seguimiento de hojas de ruta del Gobierno Autónomo Municipal de Huatajata, para la parte de administración.

✓ OBJETIVO

Brindar información necesaria para poder realizar la administración y configuración del de los usuarios del sistema.

✓ GUIA DEL SISTEMA



The image shows a login interface with the following elements:

- USUARIO**: Input field with placeholder text "Ingrese usuario".
- CONTRASEÑA**: Input field with placeholder text "Ingrese su contraseña".
- INGRESAR**: A prominent blue button for logging in.
- SEGUIMIENTO**: A smaller button located below the login button, likely for tracking or monitoring.

- 1) Ingresar el usuario
- 2) Ingresar la contraseña

MODULO DE ADMINISTRACIÓN DEL SISTEMA

3

4

5

6

N°	CI	NOMBRES	ESTADO	ROL	ACCIONES
1	9877991	JHONNY CHOQUE CONDORI	<input checked="" type="checkbox"/>	Administrador	
2	9874993	YESENIA FLORES APAZA	<input checked="" type="checkbox"/>	Administrador	
3	8422516	SANTOS CLAURE FLORES	<input checked="" type="checkbox"/>	Administrador	
4	9877998	MELINA CALLE QUISPE	<input checked="" type="checkbox"/>	Administrador	
5	8422513	NOEMI CALLIZAYA CATARI	<input checked="" type="checkbox"/>	Administrador	
6	123456	RAMIRO KANTUTA LIMACHI	<input checked="" type="checkbox"/>	Administrador	
7	8422512	GRAICE CALLIZAYA LIMACHI	<input checked="" type="checkbox"/>	Administrador	
8	9877993	MIGUEL CALLE QUISPE	<input checked="" type="checkbox"/>	Administrador	
9	12345678	Prueba Prueba Prueba	<input checked="" type="checkbox"/>	Sin roles	

Showing 1 to 9 of 9 entries

Previous 1 Next

© 2024 by Grace

3) Usuarios

4) Crear un nuevo usuario

Nuevo usuario

Persona

[SELECCIONE UNA PERSONA]

Rol

[SELECCIONE UN ROL]

Usuario

Ingrese el usuario

Contraseña

Ingrese la contraseña

Guardar Cerrar

5) Estado activo – inactivo

6) Editar usuario

Editar Usuario

Rol

Administrador

Usuario

9877998

Contraseña

Ingrese la contraseña

Guardar Cerrar

MODULO ADMINISTRACIÓN DE ROLES

Buscar

ROLES

7

+ Nuevo Rol

8

Administrador

Super Administrador

7) Roles

8) Crear nuevo Rol

Nuevo Rol ✕

Rol

Permisos

Seleccionar todos ⌵

admin_usuario_menu

usuario_submenu

usuario_nuevo

usuario_estado

usuario_editar

usuario_eliminar

roles_submenu

roles_nuevo

roles_editar

roles_eliminar

roles_vizualizar

permisos_submenu

permisos_nuevo

permisos_editar

MODULO ADMINISTRACION DE PERMISOS

Permisos

9

PERMISOS

10 ▾ Buscar:

N°	NOMBRE	ACCIONES
1	boletas_generar_licencia_constancia	<input type="checkbox"/> <input type="checkbox"/>
2	boletas_generar_licencia_aprobado	<input type="checkbox"/> <input type="checkbox"/>
3	boletas_generar_licencia_pdf	<input type="checkbox"/> <input type="checkbox"/>
4	boletas_generar_licencia_eliminar	<input type="checkbox"/> <input type="checkbox"/>

9) Crear nuevo permiso

Nuevo Permiso ✕

Permiso



MANUAL DE USUARIO

Versión 1.0

**“SISTEMA WEB DE GESTIÓN
Y CONTROL PARA EL
SEGUIMIENTO
DE HOJAS DE RUTA”**

GESTIÓN 2024

MANUAL DE USUARIO

✓ INTRODUCCION

El presente manual de administración pretende servir de guía para el uso adecuado del sistema web de gestión y control para el seguimiento de hojas de ruta del Gobierno Autónomo Municipal de Huatajata, para la parte de administración y registro de las hojas de ruta.

✓ OBJETIVO DEL SISTEMA

El objetivo del manual, es permitir al usuario final, que pueda tener la administración adecuada del "SISTEMA WEB DE GESTIÓN Y CONTROL PARA EL SEGUIMIENTO DE HOJAS DE RUTA".

✓ GUIA DEL SISTEMA

Inicio de sesión para el ingreso al sistema

Para ingresar al panel de administración del sistema primeramente debe iniciar sesión.



The image shows a login interface with the following elements:

- LOGIN**: Title at the top center.
- USUARIO**: Label for the username field.
- : Username input field.
- CONTRASEÑA**: Label for the password field.
- : Password input field.
- INGRESAR**: Large blue button to submit the login information.
- SEGUIMIENTO**: A smaller blue button located below the login button.

CONFIGURACIÓN

Tipo de contrato

The screenshot shows the 'TIPOS DE CONTRATO' configuration page. The left sidebar has 'Tipo de Contrato' selected. The main area displays a table with 3 entries. A green box labeled 'Crear un nuevo tipo de contrato' points to the '+ Nuevo' button. Another green box labeled 'editar' points to the edit icon in the 'ACCIONES' column of the first row. A third green box labeled 'eliminar' points to the delete icon in the 'ACCIONES' column of the first row.

N°	SIGLA	NOMBRE	ACCIONES
1	ITEM-	PERSONAL CON ITEM	[editar] [eliminar]
2	EVEN-	PERSONAL EVENTUAL	[editar] [eliminar]
3	CONS-LIN-	PERSONAL CONSULTOR EN LINEA	[editar] [eliminar]

The 'Editar Tipo de Contrato' form contains two input fields: 'Sigla' with the value 'ITEM-' and 'Nombre' with the value 'PERSONAL CON ITEM'. There are 'Guardar' and 'Cerrar' buttons at the bottom.

A confirmation dialog with a large orange exclamation mark icon. The text reads: '¿Estás seguro de eliminar?' followed by '¡No podrás revertir esto!'. At the bottom are 'Si, eliminarlo' and 'Cancelar' buttons.

Tipo de contrato

The screenshot shows the 'TIPOS DE CATEGORÍA' configuration page. The left sidebar has 'Tipo de categoría' selected. The main area displays a table with 3 entries. A green box labeled 'Crear un nuevo tipo de categoría' points to the '+ Nuevo' button. An orange arrow points to the 'N°' column header.

N°	NOMBRE	NIVEL	ACCIONES
1	SUPERIOR	[icon]	[editar] [eliminar]
2	EJECUTIVO	[icon]	[editar] [eliminar]
3	OPERATIVO	[icon]	[editar] [eliminar]

Grados académicos

Buscar

GRADOS ACADÉMICOS

10

Buscar:

N°	ABREVIATURA	NOMBRE	ESTADO	ACCIONES
1	SEC.	SECRETARIA	✓	
2	ARQ.	ARQUITECTO	✓	
3	TEC. SUP.	TECNICO SUPERIOR	✓	
4	UNIV.	UNIVERSITARIO	✓	
5	SRTA.	SEÑORITA	✓	
6	SRA.	SEÑORA	✓	

Crear un nuevo grado académico

+ Nuevo

Ámbito profesional

Buscar

ÁMBITOS PROFESIONALES

Crear un nuevo ámbito profesional

+ Nuevo

- ÁMBITO SOCIAL Y SERVICIOS COMUNITARIOS
- ÁMBITO AMBIENTAL
- ÁMBITO CREATIVO Y ARTES
- ÁMBITO EDUCATIVO
- ÁMBITO EMPRESARIAL
- ÁMBITO DE LA TECNOLOGÍA
- ÁMBITO LEGAL
- ÁMBITO DE LA SALUD

Mae – Unidad

Buscar

MAE

10

Buscar:

N°	NOMBRE	UNIDADES	ACCIONES
1	SECRETARÍA GENERAL		
2	DESPACHO ALCALDE MUNICIPAL		

Showing 1 to 2 of 2 entries

Previous 1 Next

Crear un nuevo Mae Unidad

+ Nuevo

Unidades administrativas

Crear un nuevo Unidad administrativa

N°	SIGLA	NOMBRE	ESTADO	ACCIONES
1	JSI	JEFATURA DE SERVICIOS INTEGRALES	<input checked="" type="checkbox"/>	
2	URSSM	UNIDAD DE REDES Y SERVICIOS DE SALUD MUNICIPAL	<input checked="" type="checkbox"/>	
3	UADHM	UNIDAD DE ADMINISTRACIÓN DEL HOSPITAL MUNICIPAL	<input checked="" type="checkbox"/>	
4	UOTC	UNIDAD DE ORDENAMIENTO TERRITORIAL Y CATASTRO	<input checked="" type="checkbox"/>	
5	UDU	UNIDAD DE DESARROLLO URBANO	<input checked="" type="checkbox"/>	
6	UDPS	UNIDAD DE PROYECTOS ESPECIALES	<input checked="" type="checkbox"/>	
7	UFDO	UNIDAD DE FISCALIZACION DE OBRAS	<input checked="" type="checkbox"/>	
8	UDGR	UNIDAD DE GESTIÓN DE RIESGOS	<input checked="" type="checkbox"/>	

Secretarías municipales – dirección

Crear un nuevo secretarías municipales dirección

N°	SIGLA	NOMBRE	ESTADO	DIRECCIONES	ACCIONES
1	SMAF	SECRETARÍA MUNICIPAL ADMINISTRATIVA FINANCIERA	<input checked="" type="checkbox"/>		
2	SMDPMA	SECRETARÍA MUNICIPAL DE DESARROLLO PRODUCTIVO Y MEDIO AMBIENTE	<input checked="" type="checkbox"/>		
3	SMTOP	SECRETARÍA MUNICIPAL TÉCNICA Y DE OBRAS PÚBLICAS	<input checked="" type="checkbox"/>		
4	SMDH	SECRETARÍA MUNICIPAL DE DESARROLLO HUMANO	<input checked="" type="checkbox"/>		

Showing 1 to 4 of 4 entries

Horarios

Crear un nuevo horario

N°	NOMBRE	DESCRIPCIÓN
1	NORMAL	HORA DE ENTRADA 08:00 SALIDA DE MAÑANA 12:00; ENTRADA TARDE 14:30 Y SALIDA TARDE 18:30
ESPECIFICACIÓN HORAS		
ACCIONES		
2	ESPECIAL CIRECA	HORA DE ENTRADA 08:00 SALIDA DE MAÑANA 12:00; ENTRADA TARDE 14:00 Y SALIDA TARDE 18:00

Showing 1 to 2 of 2 entries

Género - estado civil

Administración de Género

Administración de Estado Civil

Crear un nuevo género

+ Nuevo

N°	SIGLA	NOMBRE	ESTADO	ACCIONES
1	M	MASCULINO	✓	
2	F	FEMENINO	✓	

Showing 1 to 2 of 2 entries

Previous 1 Next

Administración de Género

Administración de Estado Civil

Crear un nuevo estado civil

+ Nuevo

N°	NOMBRE	ESTADO	ACCIONES
1	SOLTERO/A	✓	
2	CASADO/A	✓	
3	VIUDO/A	✓	
4	DIVORCIADO/A	✓	
5	CONCUVINO/A	✓	

Showing 1 to 5 of 5 entries

Previous 1 Next

REGISTROS

Personas

Administración de Género

Administración de Estado Civil

Ingresar cedula de identidad

+ Nuevo

Busqueda por CI

8422512

CI	Genera información del funcionario	: 8422512
NIT		:
NOMBRES Y APELLIDOS		: GRAICE CALLIZAYA LIMACHI
FECHA DE NACIMIENTO		: MIÉRCOLES 9 DE JUNIO DE 1999
GMAIL		: GRAICE@GMAIL.COM
CELULAR		: 69826073
DIRECCIÓN		: AVENIDA 13 DE JUNIO
GÉNERO		: FEMENINO
ESTADO CIVIL		: SOLTERO/A
ESTADO		: ACTIVO

[Editar](#)
[Eliminar](#)
[Listar Contratos](#)

CONTRATOS

GAMH

- INICIO
- ADMIN USUARIOS
- CONFIGURACIÓN
- REGISTROS
 - Personas
 - Contratos**
- ADM. BIOMETRICO
- CONFIGURACIÓN TRAMITE
- TRAMITES
- BUSCAR TRAMITES

Buscar

INFORMACIÓN PERSONAL

CI Ingrese el CI	Complemento Ingrese complemento	NIT Ingrese NIT
Fecha de nacimiento 1999-10-25 Ingrese la fecha de nacimiento	Selección Género [SELECCIONE GÉNERO]	Selección Estado Civil [SELECCIONE ESTADO CIVIL]
Nombres Ingrese nombres	Apellido Paterno Ingrese apellido paterno	Apellido Materno Ingrese apellido materno
Gmail Ingrese su correo electrónico	Nº de celular Ingrese su número de celular	Dirección Ingrese la dirección

INFORMACIÓN DEL CONTRATO

Ingresamos datos

FECHA DE INICIO Seleccione la fecha de inicio	FECHA DE CONCLUSIÓN Seleccione la fecha de inicio	SELECCIONE TIPO DE CONTRATO [SELECCIONE TIPO DE CONTRATO]
NÚMERO DE CONTRATO ----- Ingrese el número de contrato	HABER BÁSICO Bs Ingrese el haber básico	SELECCIONE LA CATEGORÍA [SELECCIONE LA CATEGORÍA]
SELECCIONE EL NIVEL [SELECCIONE EL NIVEL]	SELECCIONE EL HORARIO [SELECCIONE EL HORARIO]	

INFORMACIÓN ACADÉMICA

Seleccionamos datos

SELECCIONE ÁMBITO PROFESIONAL [SELECCIONE EL ÁMBITO PROFESIONAL]	SELECCIONE PROFESIÓN [SELECCIONE PROFESIÓN]	SELECCIONE EL GRADO ACADÉMICO [SELECCIONE EL GRADO ACADÉMICO]
---	--	--


TRABAJO  Seleccionamos datos

[SELECCIONE]

[SELECCIONE] 

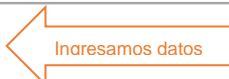
MAE  Ingresamos datos

SELECCIONE MAE SELECCIONE UNIDAD SELECCIONE CARGO




[SELECCIONE MAE]  [SELECCIONE UNIDAD]  [SELECCIONE CARGO] 

INGRESE SU CARGO


Ingrese el cargo correspondiente

SM  Ingresamos datos

SELECCIONE SECRETARIA SELECCIONE LA DIRECCIÓN *SELECCIONE UNIDAD ADMINISTRATIVA O JEFATURA SI CORRESPONDE *

[SELECCIONE LA SECRETARIA MUNICIPAL]  [SELECCIONE LA DIRECCIÓN]  [SELECCIONE UNIDAD ADMINISTRATIVA] 

SELECCIONE UN CARGO * INGRESE SU CARGO

[SELECCIONE CARGO]  Ingrese el cargo correspondiente

Registrar datos!!  Registramos datos

CONFIGURACIÓN DE TRAMITE

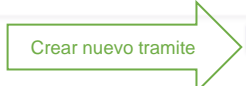
Tipo de tramite

GAMH





















INICIO

- INICIO
- ADMIN USUARIOS >
- CONFIGURACIÓN >
- REGISTROS >
- ADM. BIOMETRICO >
- CONFIGURACIÓN TRAMITE
 - Tipo de Tramite**
 - Tipos de estado
 - Habilitar a tramite
- TRAMITES
- BUSCAR TRAMITES

Buscar

 **Crear nuevo tramite** + Nuevo

10 Buscar:

N°	NOMBRE	SIGLA	ESTADO	ACCIONES
1	TRAMITE INTERNO (TI)	TI	<input checked="" type="checkbox"/>	 
2	TRAMITE EXTERNO(TE)	TE	<input checked="" type="checkbox"/> 	 
3	MEMORANDUMS	ME	<input checked="" type="checkbox"/>	 
4	INFORMES	IN	<input checked="" type="checkbox"/>	 
5	CIRCULAR	CIR	<input checked="" type="checkbox"/>	 
6	HOJA DE PERMISO	HP	<input checked="" type="checkbox"/>	 
7	COMUNICACIÓN INTERNA	CI	<input checked="" type="checkbox"/>	 
8	CERTIFICACION	CERT	<input checked="" type="checkbox"/>	 
9	COURRIER	COU	<input type="checkbox"/> 	 

Tipos de estado

Buscar

Crear nuevo estado → + Nuevo

10 Buscar:

N°	NOMBRE	COLOR	ACCIONES
1	OBSERVADA	OBSERVADA	
2	ELIMINADO	ELIMINADO	
3	CONCLUIDA	CONCLUIDA	
4	RECIBIDA	RECIBIDA	
5	SIN RECIBIR	SIN RECIBIR	
6	PENDIENTE	PENDIENTE	

Modulo tramites, so observa el cargo del funcionario en caso tuviese dos o tres cargos se podrá visualizar.

Buscar

..... CARGOS

CARGO : ENCARGADO U. DE CONTRATACIONES

4 Bandeja de entrada

3 Recibidos

Ingresamos → INGRESAR

..... CARGO :: ENCARGADO U. DE CONTRATACIONES

<p>6</p> <p>CORRESPONDENCIA</p>	<p>0</p> <p>BANDEJA DE ENTRADA</p>	<p>0</p> <p>RECIBIDOS</p>	<p>0</p> <p>OBSERVADOS</p>	<p>0</p> <p>ARCHIVADOS</p>
--	---	----------------------------------	-----------------------------------	-----------------------------------

Indresamos a correspondencia

Se visualiza todos los los tramites realizados con Cite y datos de origen.

..... CORRESPONDENCIA

Crear nuevo tramite + Nuevo

10 ▼ Buscar:

ACCIONES	N° UNICO ESTADO	CITE	DATOS ORIGEN
<p>Reporte pdf Urgente 249</p>		GAMH/ENC. U. CON./TI TRAMITE INTERNO (TI) TI	Remite: LIC. GRAICE CALLIZAYA LIMACHI Destinatario: LIC. JHONNY CHOQUE CONDORI Referencia: COMPRA DE IMPRESORAS Salida: 2024-06-17
<p>Visualizar el tramite Alta 47</p>		GAMH/ENC. U. CON./TI TRAMITE INTERNO (TI) TI	Remite: LIC. GRAICE CALLIZAYA LIMACHI Destinatario: ABG. ALVARO FERNANDEZ TINTAYA Referencia: VERIFICAR EL ESTADO DE CONTRADOS Salida: 2024-06-16
<p>Eliminar tramite si es 1ro Alta 246</p>		GAMH/ENC. U. CON./TI TRAMITE INTERNO (TI) TI	Remite: LIC. GRAICE CALLIZAYA LIMACHI Destinatario: TEC. SANTOS CLAURE FLORES Referencia: COMPRA DE COMPUTADORAS Salida: 2024-06-16
<p>Editar tramite Urgente 245</p>		GAMH/ENC. U. CON./TI TRAMITE INTERNO (TI) TI	Remite: LIC. GRAICE CALLIZAYA LIMACHI Destinatario: LIC. MELINA CALLE QUISPE Referencia: COMPRA DE MAQUINAS Salida: 2024-06-16

Nueva Correspondencia Ingresa datos de la correspondencia

Tipo trámite	Número de hojas	Número de anexos
<input type="text" value="[TE] TRAMITE EXTERNO(TE)"/>	<input type="text" value="3"/>	<input type="text" value="2"/>
Cite	N°	Gestión
<input type="text" value="GAMH/ENC. U. CON./TE"/>	<input type="text" value="06"/>	<input type="text" value="2024"/>
Remite	Prioridad	
<input type="text" value="Sr. EUGENIO CALLE"/>	<input type="text" value="Alta"/>	
Cargo del Remite	<div style="border: 1px solid gray; padding: 5px;"> <p>[SELECCIONE PRIORIDAD]</p> <p>Alta</p> <p>Media</p> <p>Baja</p> <p>Urgente</p> </div>	
Referencia	<p><input type="text" value="ALCALDE DE ACHACAHU"/></p>	
Destinatario	<p><input type="text" value="TERRENOS VALDIOS"/></p>	
Instructivo	<p><input type="text" value="[ENCARGADO DE LA U. DE FISCALIZACION] MIGUEL CALLE QUISPE"/></p>	
<p><input type="text" value="PARA SU CONOCIMIENTO"/></p>		
<p> <input type="button" value="Guardar"/> <input type="button" value="Cerrar"/> Guardar-Cerrar </p>		



MANUAL TÉCNICO

Versión 1.0

**“SISTEMA WEB DE GESTIÓN
Y CONTROL PARA EL
SEGUIMIENTO
DE HOJAS DE RUTA”**

GESTIÓN 2024

✓ INTRODUCCIÓN

Este manual técnico proporciona una guía detallada sobre los controladores y modelos utilizados en el sistema de contratos. Está diseñado para desarrolladores y técnicos que necesiten entender y mantener el código.

Sección 1: Controlador de Contrato

Ubicación

- **Archivo:** \app\Http\Controllers\Registro\Controlador_contrato

Funcionalidades

- **Guardar Nuevo Contrato:** Esta función permite crear y almacenar un nuevo contrato en el sistema.

```
use App\Models\Configuracion\Nivel;
use App\Models\Configuracion\Profesion;
use App\Models\Configuracion\Secretaria_municipal;
use App\Models\Configuracion\Tipo_baja;
use App\Models\Configuracion\Tipo_categoria;
use App\Models\Configuracion\Tipo_contrato;
use App\Models\Configuracion\Unidad_mae;
use App\Models\Configuracion\Unidades_administrativas;
use App\Models\Registro\Contrato;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\Validator;

class Controlador_contrato extends Controller
{
    /**
     * @version 1.0
     * @author Graice Callizaya Chambi <graicecallizaya1234@gmail.com>
     * @param Controlador Administrar la parte de la administracion de los contratos
     * ¡Muchas gracias por preferirnos! Esperamos poder servirte nuevamente
     */
    private $menu_registro = 'registro';
    /**
     * PARA LA ADMINISTRACION DEL CONTRATO
     */
    //para ver la vista de los contrato
    public function contrato(){
        $data['menu'] = '14';
        $data['lis_genero'] = Genero::where('estado','activo')->orderBy('id','asc')->get();
        $data['lis_estado_civil'] = Estado_civil::orderBy('id','asc')->get();
        $data['lis_tipo_contrato'] = Tipo_contrato::orderBy('id','asc')->get();
        $data['lis_categoria'] = Tipo_categoria::orderBy('id','asc')->get();
        $data['lis_ambito'] = Ambito::orderBy('id','asc')->get();
        $data['lis_horario'] = Horario::orderBy('id','asc')->get();
        $data['grado_academico'] = Grado_academico::orderBy('id','asc')->get();
        $data['listar_mae'] = Mae::orderBy('id','asc')->get();
        $data['secretaria_municipal'] = Secretaria_municipal::where('estado','activo')->orderBy('id','asc')->get();
        $data['unidad_administrativa'] = Unidades_administrativas::where('estado','activo')->orderBy('id','asc')->get();
        return view('administrador.contrato.contrato',$data);
    }
}
```

```

//PARA GUARDAR EL CONTRATO
public function guardar_contrato(Request $request){
    $validar = Validator::make($request->all(),[
        'fecha_nacimiento' => 'required',
        'genero'           => 'required',
        'estado_civil'    => 'required',
        'nombres'         => 'required',
        'apellido_paterno' => 'required',
        'apellido_materno' => 'required',
        'gmail'           => 'required',
        'numero_celular'  => 'required',
        'direccion'       => 'required',
        //desde aqio informacion para el contrato
        'fecha_inicio'    => 'required',
        //'fecha_conclusion' => 'required',
        'tipo_contrato'   => 'required',
        'numero_contrato' => 'required',
        'haber_basico'    => 'required',
        'categoria'       => 'required',
        'nivel'           => 'required',
        'horario'         => 'required',
        //INFORMACIÓN ACADEMICA
        'ambito_profesional'=> 'required',
        'profesion'       => 'required',
        'grado_academico' => 'required',
    ]);
    if($validar->fails()){
        $data = mensaje_mostrar('errores', $validar->errors());
    }else{
        //primero verificamos si la creamos persona o no
        if($request->id_persona != '' || $request->id_persona != null){
            $persona = Persona::find($request->id_persona);
        }else{
            $persona = new Persona();
            $persona->ci = $request->ci;
            $persona->nombres = $request->nombres;
            $persona->ap_paterno = $request->apellido_paterno;
            $persona->ap_materno = $request->apellido_materno;
            $persona->fecha_nacimiento = $request->fecha_nacimiento;

```

- **Modelo en Relación:** Describe cómo este controlador interactúa con otros modelos relacionados en la base de datos.

```
//relacion reversa con nivel
public function nivel(){
    return $this->belongsTo(Nivel::class, 'id_nivel', 'id');
}
//relacion reversa con tipo de contrato
public function tipo_contrato(){
    return $this->belongsTo(Tipo_contrato::class, 'id_tipo_contrato', 'id');
}
//relacion con persona
public function persona() {
    return $this->belongsTo(Persona::class, 'id_persona', 'id');
}
//relacion reversa con cargo mae
public function cargo_mae(){
    return $this->belongsTo(Cargo_mae::class, 'id_cargo_mae', 'id');
}
//relacion reversa con cargo_sm
public function cargo_sm(){
    return $this->belongsTo(Cargo_sm::class, 'id_cargo_sm', 'id');
}
//relacion reversa con id_profesion
public function profesion(){
    return $this->belongsTo(Profesion::class, 'id_profesion', 'id');
}
//relacion reversa con grado academico
public function grado_academico(){
    return $this->belongsTo(Grado_academico::class, 'id_grado_academico', 'id');
}
//relacion con el horario
public function horario(){
    return $this->belongsTo(Horario::class, 'id_horario', 'id');
}
//relacion con el usuario
public function usuario(){
    return $this->belongsTo(User::class, 'id_usuario', 'id');
}
//relacion reversa con baja
public function baja(){
    return $this->belongsTo(Baja::class, 'id_baja', 'id');
}
```

Sección 2: Controlador de Registro del Personal

Ubicación

- **Archivo:** \app\Http\Controllers\Registro\Controlador_persona

Funcionalidades

- **Registro del Personal:** Explica cómo se maneja la creación y actualización de registros de personal.

```
class Controlador_persona extends Controller
{
    /**
     * @version 1.0
     * @author Graice Callizaya Chambi <graicecallizaya1234@gmail.com>
     * @param Controlador Administrar la parte de LA ADMINISTRACION DE LAS PERSONAS QUE ESTEN REGISTRADAS
     * ¡Muchas gracias por preferirnos! Esperamos poder servirte nuevamente
     */
    /**
     * ADMINISTRACION DEL LAS PERSONAS
     */
    public function persona(){
        $data['menu'] = '13';
        $data['lis_genero'] = Genero::where('estado','activo')->orderBy('id','asc')->get();
        $data['lis_estado_civil'] = Estado_civil::orderBy('id','asc')->get();
        return view('administrador.persona.persona', $data);
    }
    //para la busqueda de la persona con por ci
    public function persona_buscar(Request $request){
        $persona = Persona::with(['genero','estado_civil'])->where('ci', $request->ci)->get();
        if(!$persona->isEmpty()){
            $data['listar_persona'] = $persona;
        }else{
            $data['listar_persona'] = null;
        }
        return view('administrador.persona.listado_persona', $data);
    }
    //para realizar el registro de la nueva persona
    public function persona_nuevo(Request $request){
        $validar = Validator::make($request->all(),[
            'ci' => 'required|unique:rl_persona,ci',
            'fecha_nacimiento' => 'required|date|before_or_equal:' . now()->format('Y-m-d'),
            'genero' => 'required',
            'estado_civil' => 'required',
            'nombres' => 'required',
            'apellido_paterno' => 'required',
            'gmail' => 'required',
            'numero_celular' => 'required',
            'direccion' => 'required'
        ],
        [
            'fecha_nacimiento.before_or_equal' => 'La fecha de nacimiento no puede ser en el futuro.',
        ]
    );
    }
}
```

```

class Persona extends Model
{
    use HasFactory;
    protected $table = 'rl_persona';
    protected $fillable=[
        'ci',
        'complemento',
        'nit',
        'nombres',
        'ap_paterno',
        'ap_materno',
        'fecha_nacimiento',
        'gmail',
        'celular',
        'direccion',
        'estado',
        'id_genero',
        'id_estado_civil',
        'id_usuario',
    ];
    const CREATED_AT = 'creado_el';
    const UPDATED_AT = 'editado_el';

    //relacion reversa con rl_genero
    public function genero(){
        return $this->belongsTo(Genero::class, 'id_genero', 'id');
    }

    //relacion reversa con rl_estado_civil
    public function estado_civil(){
        return $this->belongsTo(Estado_civil::class, 'id_estado_civil', 'id');
    }

    //relacion de uno a muchos con el contrato
    public function contrato(){
        return $this->hasMany(Contrato::class, 'id_persona', 'id');
    }
}

```


Sección 3: Controlador de Trámite

Ubicación

- **Archivo:** \app\Http\Controllers\Tramite\Controlador_tramite

Funcionalidades

- **Control de Trámite:** Descripción de las operaciones de gestión de trámites, incluyendo la relación entre los diferentes modelos de trámite.

```
class Controlador_tramite extends Controller{
    /**
     * @version 1.0
     * @author Graice Callizaya Chambi <graicecallizaya1234@gmail.com>
     * @param Controlador Administrar la parte de LA ADMINISTRACION DE LOS TRAMITES QUE SE REALIZARAN
     * ¡Muchas gracias por preferirnos! Esperamos poder servirte nuevamente
     */
    //Para la administracion de los tramites
    public function vizualizar_cargos_tramite() {
        $data['menu'] = 60;

        $listar_cargos = User_cargo_tramite::with([
            'tramite' => function ($us_tramite) {
                $us_tramite->with([
                    'hojas_ruta',
                    'tipo_tramite'
                ])->withCount([
                    'hojas_ruta as bandeja_entrada_count' => function ($query) {
                        $query->where('estado_id', 2);
                    },
                    'hojas_ruta as recibidos_count' => function ($query) {
                        $query->where('estado_id', 3);
                    }
                ]);
            },
            'cargo_sm',
            'cargo_mae'
        ])
        ->where('estado', true)
        ->where('id_usuario', Auth::user()->id)
        ->get();

        $data['listar_cargos'] = $listar_cargos;
        return view('administrador.tramite.vista_tramite', $data);
    }
}
```

```

class Tramite extends Model
{
    use HasFactory;
    protected $table = 'r1_tramite';
    protected $fillable=[
        'fecha_creada',
        'hora_creada',
        'fecha_hora_creada',
        'cite',
        'cite_texto',
        'numero_hojas',
        'numero_anexos',
        'referencia',
        'remitente_nombre',
        'remitente_cargo',
        'remitente_sigla',
        'remitente_dni'
    ];
}

```

```

//relacion de uno amuchos
public function hojas_ruta(){
    return $this->hasMany(Hojas_ruta::class, 'tramite_id', 'id');
}

//relacion tipo_prioridad
public function tipo_prioridad(){
    return $this->belongsTo(Tipo_prioridad::class, 'id_tipo_prioridad', 'id');
}

//relacion tipo_tramite
public function tipo_tramite(){
    return $this->belongsTo(Tipo_tramite::class, 'id_tipo_tramite', 'id');
}

//relacion con tipo_estado
public function estado_tipo(){
    return $this->belongsTo(Tipo_estado::class, 'id_estado', 'id');
}

//relacion con remitente user_cargo_tram
public function remitente_user(){
    return $this->belongsTo(User_cargo_tramite::class, 'remitente_id', 'id');
}

//relacion con destinatario user_cargo_tram
public function destinatario_user(){
    return $this->belongsTo(User_cargo_tramite::class, 'destinatario_id', 'id');
}

//relacion con cargo user_cargo_tram
public function user_cargo_tramite(){
    return $this->belongsTo(User_cargo_tramite::class, 'user_cargo_id', 'id');
}
}

```

Un resumen de las capacidades del sistema y pautas para su mantenimiento y actualización.