

# UNIVERSIDAD PÚBLICA DE EL ALTO

## CARRERA INGENIERÍA DE SISTEMAS



## PROYECTO DE GRADO

**“SISTEMA WEB PARA LA ADMINISTRACIÓN DE INFORMACIÓN Y CONTROL DE LA RECAUDACIÓN DE RECURSOS DEL SUMINISTRO DE AGUA POTABLE”**

**CASO: GOBIERNO AUTÓNOMO MUNICIPAL DE CHULUMANI**

**Para Optar al Título de Licenciatura en Ingeniería de Sistemas**

**MENCIÓN: GESTIÓN Y PRODUCCIÓN**

**Postulante: Univ. Noemi Liz Solarez Chico**

**Tutor Metodológico: Lic. Ing. Dionicio Henry Pacheco Rios**

**Tutor Revisor: M. Sc. Lic. Ing. Juan Fernando Chambi Guachalla**

**Tutor Especialista: Ing. Rolando Ulo Mollo**

**EL ALTO – BOLIVIA**

**2023**

## **DECLARACIÓN JURADA DE AUTENTICIDAD Y RESPONSABILIDAD**

Yo, **Noemi Liz Solarez Chico** estudiante con **C.I. 8431010 LP** mediante la presente **declaro** de manera pública que la propuesta del **PROYECTO DE GRADO** titulada **“SISTEMA WEB PARA LA ADMINISTRACIÓN DE INFORMACIÓN Y CONTROL DE LA RECAUDACIÓN DE RECURSOS DEL SUMINISTRO DE AGUA POTABLE”** 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 **PROYECTO 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 a las normas establecidas y vigentes de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

**El Alto, diciembre del 2023**

  
**Noemi Liz Solarez Chico**

**C.I. 8431010 LP**

***noemilizsolarez@gmail.com***

## *DEDICATORIA*

El presente proyecto de grado va dedicado a mi familia, a mi madre Rebeca Lea Chico Carrillo por brindarme esa educación y sabiduría, nacer en mí, esa inspiración a ser una mejor persona. A mi señor padre Martin Solarez Mamani por su emotivo apoyo incondicional, a mis hermanas por el impulso a seguir un buen ejemplo para ellas.

A todos mis amigas y amigos, por la ayuda constante en el transcurso de nuestro tiempo en la carrera.

# *AGRADECIMIENTO*

Agradezco a Dios por guiarme, por darme salud y fuerza para finalizar este largo camino de trayectoria.

A toda mi familia por estar ahí presente en cada paso de mi vida.

Al Ing. Dionicio Henry Pacheco Ríos por su enseñanza y paciencia en el desarrollo del presente proyecto de grado.

Al Ing. Rolando Ulo Mollo por su tiempo y paciencia en la revisión del trabajo de grado.

Al Ing. Juan Fernando Chambi Guachalla por su tiempo, enseñanza y paciencia en la inspección del documento de proyecto de grado.

*Noemi Liz Solarez Chico*

## ÍNDICE

CAPÍTULO I.....	1
1. MARCO PRELIMINAR .....	1
1.1. INTRODUCCIÓN.....	1
1.2. ANTECEDENTES.....	1
1.2.1. Estructura Organizacional.....	3
1.2.2. Antecedentes Internacionales.....	4
1.2.3. Antecedentes Nacionales .....	5
1.2.4. Antecedentes locales.....	6
1.3. PLANTEAMIENTO DEL PROBLEMA.....	6
1.3.1. Problema principal .....	7
1.3.2. Problemas específicos.....	7
1.3.3. Formulación del problema .....	8
1.4. OBJETIVOS .....	8
1.4.1. Objetivo General.....	8
1.4.2. Objetivos Específicos .....	8
1.5. JUSTIFICACIÓN .....	9
1.5.1. Justificación Técnica.....	9
1.5.2. Justificación Económica.....	9
1.5.3. Justificación Social .....	10
1.6. METODOLOGÍA.....	10
1.6.1. UWE .....	10
1.6.2. Método de Ingeniería.....	10
1.6.2.1. Metodología UWE (UML-BASED WEB ENGINEERING).....	10
1.6.2.2. Características UWE .....	11
1.7. METRICAS DE CALIDAD.....	11
1.7.1. Control de seguridad ISO/IEC 27001 .....	11
1.8. HERRAMIENTAS .....	12
1.9. LIMITES Y ALCANCES .....	14
1.9.1. Límites.....	14
1.9.2. Alcances.....	14
1.10. APORTES .....	14
CAPÍTULO II.....	16
2. MARCO TEÓRICO .....	16

2.1.	INTRODUCCIÓN.....	16
2.2.	DEFINICIÓN DE TÉRMINOS .....	16
2.2.1.	Sistema .....	16
2.2.2.	Sistema Web .....	17
2.2.3.	Administración .....	17
2.2.4.	Sistema Web de administración.....	18
2.2.5.	Información.....	18
2.2.6.	Administración de la Información .....	18
2.2.7.	Control.....	19
2.2.8.	Recaudación.....	19
2.2.9.	Suministro .....	20
2.2.10.	Recaudación del Suministro .....	20
2.2.11.	Suministro Agua Potable .....	20
2.3.	INGENIERÍA DE REQUERIMIENTOS.....	21
2.3.1.	Actividades de la Ingeniería de Requerimientos .....	21
2.3.2.	Requerimientos funcionales.....	22
2.3.3.	Requerimientos no funcionales.....	23
2.4.	MODELO ITERATIVO E INCREMENTAL.....	24
2.4.1.	Modelo Iterativo e Incremental.....	24
2.4.2.	Modelo Incremental .....	24
2.4.3.	Modelo Iterativo .....	25
2.5.	INGENIERÍA WEB.....	26
2.5.1.1.	Fases de la UWE.....	28
2.5.2.	Captura, análisis y especificación de requisitos .....	29
2.5.3.	Diseño del sistema .....	30
2.5.3.1.	Diagrama de Casos de Usos. ....	30
2.5.3.2.	Modelo Conceptual.....	31
2.5.3.3.	Modelo de navegación.....	32
2.5.3.4.	Modelo de presentación .....	33
2.5.3.5.	Modelo de procesos .....	34
2.5.4.	Codificación del sistema .....	35
2.5.5.	Pruebas .....	35
2.5.6.	Instalación o fase de Implementación.....	36
2.5.7.	Mantenimiento .....	36

2.6.	LENGUAJE UNIFICADO DE MODELADO UML.....	36
2.7.	ARQUITECTURA CLIENTE/SERVIDOR .....	37
2.7.1.	Características del modelo cliente servidor.....	37
2.7.2.	Componentes del modelo cliente servidor .....	39
2.7.3.	Diferencia entre cliente y servidor.....	39
2.7.4.	Tipos de cliente servidor .....	41
2.8.	MÉTRICAS DE CALIDAD.....	41
2.8.1.	Norma ISO/IEC 25000.....	42
2.8.2.	Funcionalidad .....	42
2.8.3.	Fiabilidad .....	43
2.8.4.	Usabilidad.....	44
2.8.5.	Eficiencia .....	45
2.8.6.	Mantenibilidad .....	45
2.8.7.	Portabilidad.....	46
2.9.	SEGURIDAD DE SOFTWARE ISO – 27001.....	46
2.9.1.	Estructura de la ISO 27001.....	47
2.10.	PRUEBAS DE ACEPTACIÓN.....	49
2.10.1.	Prueba de caja blanca .....	49
2.10.2.	Prueba de Ruta básica .....	50
2.10.3.	Prueba de caja negra.....	52
2.10.4.	Prueba de estrés .....	53
2.11.	ESTIMACIÓN DE COSTO.....	53
2.11.1.	Características de COCOMO II.....	55
2.11.2.	Modelos de COCOMO II.....	56
2.11.3.	Estimación de esfuerzo.....	57
2.11.4.	Estimación del cronograma.....	58
2.12.	HERRAMIENTAS .....	58
2.12.1.	Lenguaje de programación .....	58
2.12.1.1.	PHP .....	58
2.12.1.2.	JavaScript .....	59
2.12.2.	Base de Datos .....	59
2.12.2.1.	PostgreSQL.....	59
2.12.3.	Framework .....	60
2.12.3.1.	Backend .....	60

2.12.3.2.	Laravel.....	60
2.12.3.3.	Frontend.....	61
2.12.4.	HTML.....	61
2.12.5.	Css.....	61
CAPÍTULO III.....		62
3.	MARCO APLICATIVO.....	62
3.1.	INTRODUCCIÓN.....	62
3.2.	DESARROLLO MODELADO MEDIANTE UWE.....	62
3.2.1.	Captura, análisis y especificación de requisitos.....	62
3.2.2.	Requerimientos Funcionales.....	63
3.2.3.	Requerimientos No Funcionales.....	65
3.2.4.	Diseño del sistema.....	66
3.2.4.1.	Descripción de actores.....	66
3.2.4.2.	Diagrama de caso de uso general.....	68
3.2.4.3.	Especificaciones de casos de usos.....	69
3.2.4.4.	Diseño conceptual.....	82
3.2.4.5.	Modelo de navegación.....	84
3.2.4.6.	Modelo de presentación.....	86
3.2.4.7.	Modelo de implementación.....	94
3.2.5.	Codificación.....	103
CAPÍTULO IV.....		105
4.	PRUEBAS DE EVALUACIÓN Y RESULTADOS.....	105
4.1.	MÉTRICAS DE CALIDAD ISO 25000.....	105
4.1.1.	Funcionalidad.....	105
4.1.2.	Confiabilidad.....	110
4.1.3.	Usabilidad.....	111
4.1.4.	Eficiencia.....	113
4.1.5.	Mantenibilidad.....	115
4.1.6.	Portabilidad.....	116
4.1.7.	Seguridad del software ISO - 27001.....	117
4.1.8.	Seguridad Lógica.....	117
4.1.9.	Seguridad Física.....	118
4.2.	PRUEBAS DE FUNCIONAMIENTO.....	118
4.2.1.	Prueba de caja blanca.....	119

4.2.2.	Prueba de caja negra .....	121
4.2.3.	Pruebas de estrés.....	123
CAPÍTULO V .....		125
5.	ESTIMACIÓN DE COSTO.....	125
5.1.	COCOMO II.....	125
CAPÍTULO VI .....		129
6.	CONCLUSIONES Y RECOMENDACIONES .....	129
6.1.	CONCLUSIONES.....	129
6.2.	RECOMENDACIONES.....	130
7.	BIBLIOGRAFÍA.....	131
ANEXOS.....		137

## ÍNDICE DE TABLAS

<b>Tabla 1.</b> Actividades de la ir para diferentes modelos de procesos de ingeniería de software .....	22
<b>Tabla 2.</b> Estimación de Esfuerzo .....	57
<b>Tabla 3.</b> Captura, análisis y especificación de requisitos .....	62
<b>Tabla 4.</b> Requerimientos funcionales.....	63
<b>Tabla 5.</b> Requerimientos no funcionales.....	65
<b>Tabla 6.</b> Descripción de actores .....	67
<b>Tabla 7.</b> Descripción de caso de uso - Ingreso al sistema .....	70
<b>Tabla 8.</b> Descripción de caso de uso de afiliados .....	72
<b>Tabla 9.</b> Descripción de caso de uso - Registro de propiedad de afiliado .....	74
<b>Tabla 10.</b> Descripción de caso de uso - Instalación del servicio .....	76
<b>Tabla 11.</b> Descripción del caso de uso - Control del servicio .....	80
<b>Tabla 12.</b> Descripción de caso de uso - reportes administrativas del suministro del servicio	81
<b>Tabla 13.</b> Número de entradas de usuario.....	105
<b>Tabla 14.</b> Número de salidas de usuario .....	106
<b>Tabla 15.</b> Peticiones de usuario .....	106
<b>Tabla 16.</b> Punto función sin ajustar.....	107
<b>Tabla 17.</b> Factores de Complejidad.....	108
<b>Tabla 18.</b> Escala de Calificación.....	111
<b>Tabla 19.</b> Cuestionario de evaluación.....	112
<b>Tabla 20.</b> Escala de valores de eficiencia.....	113
<b>Tabla 21.</b> Valoración en la eficiencia del sistema .....	114
<b>Tabla 22.</b> Resultados.....	117
<b>Tabla 23.</b> Caso de prueba - ingreso al sistema.....	121
<b>Tabla 24.</b> Caso de prueba - Administrar persona natural y jurídica.....	122
<b>Tabla 25.</b> Caso de prueba Administrar cobros.....	122
<b>Tabla 26.</b> Auditoria de desempeño.....	124
<b>Tabla 27.</b> Coeficientes de a, b, c, d de COCOMO II .....	125
<b>Tabla 28.</b> Ecuaciones de Cocomo II.....	125
<b>Tabla 29.</b> Cálculo de Atributos FAE.....	126
<b>Tabla 30.</b> Resumen de costo.....	128

## ÍNDICE DE FIGURAS

<b>Figura 1.</b> Organigrama .....	3
<b>Figura 2.</b> Clasificación de requerimientos no funcionales .....	23
<b>Figura 3.</b> Modelo incremental.....	25
<b>Figura 4.</b> Desarrollo iterativo e incremental .....	26
<b>Figura 5.</b> Vista General de las Transformaciones entre modelos de UWE. ....	28
<b>Figura 6.</b> Fases de la Metodología UWE .....	29
<b>Figura 7.</b> Diagrama de Casos de Uso. ....	31
<b>Figura 8.</b> Modelo de Contenido (UWE): MHC-PMS.....	32
<b>Figura 9.</b> Modelo de Navegación (UWE): MHC-PMS. ....	33
<b>Figura 10.</b> Modelo de Presentación (UWE): MHC-PMS. ....	34
<b>Figura 11.</b> Modelo de Estructura del Proceso (UWE): MHC-PMS. ....	35
<b>Figura 12.</b> Modelo cliente-servidor .....	37
<b>Figura 13.</b> Estructura de una aplicación Cliente Servidor .....	40
<b>Figura 14.</b> División de la norma ISO/IEC 25000 .....	42
<b>Figura 15.</b> Caja blanca .....	50
<b>Figura 16.</b> Prueba del camino básico .....	51
<b>Figura 17.</b> Prueba de Caja Negra.....	52
<b>Figura 18.</b> Modelo arquitectura COCOMO II .....	56
<b>Figura 19.</b> Diagrama de caso de uso general.....	69
<b>Figura 20.</b> Diagrama de caso de uso - Ingreso al sistema.....	70
<b>Figura 21.</b> Caso de uso – Gestión de administración información afiliación de usuarios .....	72
<b>Figura 22.</b> Diagrama de caso de uso - Registro de propiedad al afiliado.....	74
<b>Figura 23.</b> Caso de uso instalación del servicio.....	76
<b>Figura 24.</b> Caso de uso – Categorización de cobro del servicio .....	78
<b>Figura 25.</b> Descripción del caso de uso - cobro del servicio .....	78
<b>Figura 26.</b> Caso de uso - control del servicio .....	80
<b>Figura 27.</b> Diseño conceptual.....	83
<b>Figura 28.</b> Modelo de navegación – Administrador .....	84
<b>Figura 29.</b> Modelo de navegación - técnico.....	85
<b>Figura 30.</b> Modelos de navegación - cajero.....	86
<b>Figura 31.</b> Modelo de presentación - Login .....	87
<b>Figura 32.</b> Modelo de presentación - inicio.....	87

<b>Figura 33.</b> Modelo de presentación - sistemas .....	88
<b>Figura 34.</b> Modelo de presentación - recaudaciones .....	88
<b>Figura 35.</b> Categoría de cobro .....	89
<b>Figura 36.</b> Registro de Zonas.....	89
<b>Figura 37.</b> Registro de nuevas zonas.....	90
<b>Figura 38.</b> Registro datos profesiones.....	90
<b>Figura 39.</b> Registro tipo de empresa .....	91
<b>Figura 40.</b> Registro tipo de propiedad .....	91
<b>Figura 41.</b> Afiliación persona natural .....	92
<b>Figura 42.</b> Afiliación persona jurídica .....	92
<b>Figura 43.</b> Vista unidades de trabajo.....	93
<b>Figura 44.</b> Vista para adicionar personal de trabajo en las áreas .....	93
<b>Figura 45.</b> Instalación de servicio .....	94
<b>Figura 46.</b> Login .....	94
<b>Figura 47.</b> Seleccionar acceso .....	95
<b>Figura 48.</b> Administrar usuarios .....	95
<b>Figura 49.</b> Administrar recaudaciones.....	96
<b>Figura 50.</b> Configuración de categorías.....	96
<b>Figura 51.</b> Tipo de zonas .....	97
<b>Figura 52.</b> Administración de zonas .....	97
<b>Figura 53.</b> Administrar profesiones.....	98
<b>Figura 54.</b> Tipo de empresas .....	98
<b>Figura 55.</b> Tipo de propiedad .....	99
<b>Figura 56.</b> Registro de persona natural .....	99
<b>Figura 57.</b> Registro de persona jurídica.....	100
<b>Figura 58.</b> Registro de unidades .....	100
<b>Figura 59.</b> Registro de cargos .....	101
<b>Figura 60.</b> Registro de personal .....	101
<b>Figura 61.</b> Registro del personal .....	102
<b>Figura 62.</b> Registrar instalación.....	102
<b>Figura 63.</b> Codificación principal del software. ....	103
<b>Figura 64.</b> Codificación de datos mediante el ORM.....	103
<b>Figura 65.</b> Codificación parte de la administración de usuarios.....	104
<b>Figura 66.</b> Flujograma de recaudaciones .....	119

## RESUMEN

El presente proyecto de grado está enfocado en la administración y control de los siguientes procesos del agua potable, lo cual tiene como objetivo asegurar y brindar una consolidación específica del control del servicio hacia a los usuarios.

El proyecto se utilizó las tecnologías emergentes como ser, PostgreSQL para el almacenamiento de datos a grandes escalas, que de tal manera se puede distribuir los datos en varios nodos para mejorar el rendimiento.

El proyecto se basó en las metodologías UWE, metodología orientada a la programación de aplicaciones Web, También se buscó garantizar una buena calidad del sistema adecuado al estándar del ISO 25000, esta que nos proporcionó realizar el funcionamiento de asegurar al cumplimiento de los estándares de calidad establecidas del.

El proyecto se desarrolló con una interfaz dinámica y amigable para tener una buena experiencia intuitiva y fácil de usar para los usuarios que tienen acceso al sistema.

**Palabras Clave:** ISO 25000, UWE, Administración, Control, PostgreSQL.

## **ABSTRACT**

This degree project is focused on the administration and control of the following drinking water processes, which aims to ensure and provide a specific consolidation of the control of the service to users.

The project used emerging technologies such as PostgreSQL for large-scale data storage, which can distribute data across multiple nodes to improve performance.

The project was based on the UWE methodologies, a methodology oriented to the programming of Web applications. It also sought to guarantee good quality of the system appropriate to the ISO 25000 standard, which allowed us to carry out the operation of ensuring compliance with quality standards. established of the.

The project was developed with a dynamic and friendly interface to have a good intuitive and easy-to-use experience for users who have access to the system.

**Keywords:** ISO 25000, UWE, Administration, Control, PostgreSQL.

## LISTADO DE ABREVIATURAS

**OTBS:** Organización Territorial de Base.

**UWE:** Metodología basada en el proceso Unificado.

**HTML:** Lenguaje de Modelado de Hiper Texto.

**UML:** Lenguaje Unificado de Modelado.

**MVC:** Modelo, Vista y Controlador.

**HTTP:** Protocolo de Transferencia de Hipertexto.

**MII:** Modelo Iterativo Incremental

**RF:** Requerimientos Funcionales.

**RNF:** Requerimientos no Funcionales.

**CSS:** Cascading Style Sheets.

**PHP:** Personal Home Page (Hypertext Preprocessor).

**COCOMO:** Modelo Constructivo de Costos (COnstructive COst MOdel).

# **CAPÍTULO I**

## **MARCO PRELIMINAR**

## **1. MARCO PRELIMINAR**

### **1.1. INTRODUCCIÓN**

Hoy en día, los sistemas accesibles, integrados y de fácil manejo son una necesidad dentro de las empresas e instituciones ya sean públicas o privadas, la razón principal implica la cantidad de información que se maneja, hace que el tratamiento automático de la información sea realmente útil y necesario, una solución a esta demanda es el desarrollo de sistemas Web, que para su manejo solo requiere de un navegador web por parte del cliente.

El desarrollo de este proyecto web tiene como finalidad brindar información a los clientes y funcionarios del Gobierno Autónomo Municipal de Chulumani para la administración eficiente de la información generada de la recaudación de recursos del suministro de agua potable. El gobierno Autónomo Municipal de Chulumani actualmente cuenta con formularios impresos y libros foliados para la administración de la recaudación del servicio de agua, este proceso es deficiente y no tiene el control de los materiales y las herramientas que se utilizan para las instalaciones, reparaciones y corte del servicio de agua potable, generando así pérdidas económicas en El Gobierno Autónomo Municipal de Chulumani.

Por lo tanto, el sistema web para la administración de información se desarrolló utilizando una metodología ágil de desarrollo, y para el desarrollo Web, Laravel usa lenguaje de programación PHP considerado uno de los más utilizados para el desarrollo de aplicaciones Web ofreciendo estabilidad y seguridad en el funcionamiento, PostgreSQL como gestor de base de datos, los mismos que permiten satisfacer los requerimientos de la mencionada institución.

### **1.2. ANTECEDENTES**

El Gobierno Autónomo Municipal de Chulumani, Provincia Sud Yungas del departamento de La Paz, representado legalmente para este acto por el Señor. Ramiro Navia Avendaño, con Cedula de Identidad N° 5951593 LP., nombrada mediante ACTA DE JURAMENTO DE POSESION DE FECHA 03 DE MAYO DE 2021, ministrada por el Juez Público Mixto Civil, Comercial y de familia, con asiento en la ciudad de

Chulumani de la provincia de Sud Yungas del Departamento de La Paz en calidad de ALCALDE MUNICIPAL "TITULAR" del Municipio de CHULUMANI, que para fines legales en el presente documento en adelante se denominara LA ENTIDAD.

Actualmente el Gobierno Autónomo Municipal de Chulumani realiza el proceso de registro de recaudaciones utilizando formularios impresos donde el llenado se hace de manera manual, de la misma forma se realiza el registro de cobros, pagos y del proceso de atención a instalaciones y reparaciones, es decir que las solicitudes de instalación, reparación o corte del servicio de agua potable son asignados aleatoriamente al personal encargado, generando inconformidad de los clientes por el tiempo de respuesta a las solicitudes.

- **Misión**

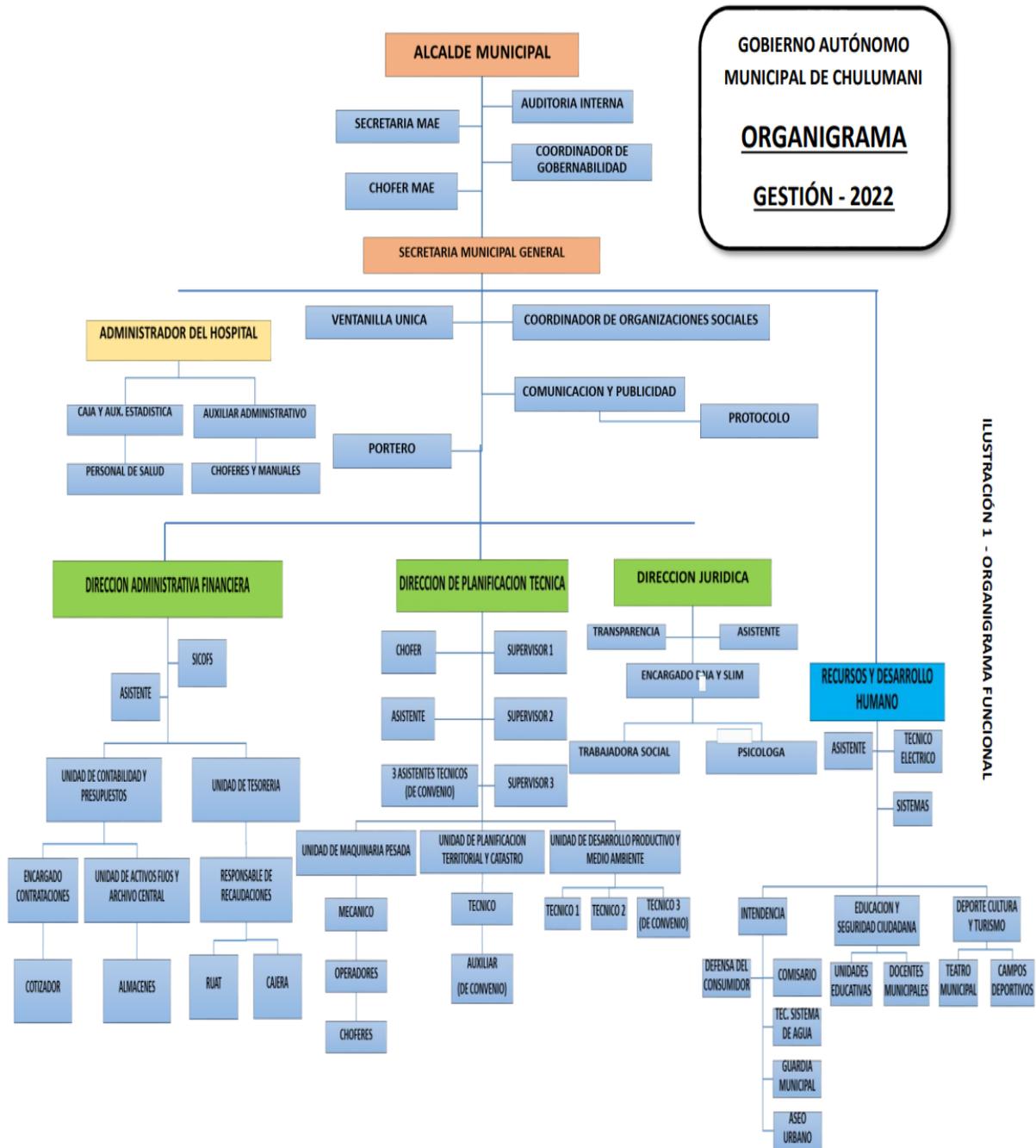
Somos una institución pública Municipal Autónoma, cuya misión es de mejorar la calidad de vida de las comunidades y pobladores del municipio de Chulumani generando y ejecutando políticas de desarrollo integral en corresponsabilidad con su comunidad, administrando su territorio y prestando servicios con transparencia, equidad y calidad con sus funcionarios públicos motivados, comprometidos con solvencia técnica de favor de los habitantes del municipio.

- **Visión**

Resolver problemas, satisfacer necesidades y/o disminuir o eliminar un déficit de presente a través de ejecución de planes, políticas, programas, proyectos y actividades que beneficie de la sociedad, sujetos a los elementos de eficacia, eficiencia, transparencia, responsabilidad y capacidad administrativa coadyuvando con el nivel departamental y nacional enmarcados en la normativa vigente que rige nuestro país.

### 1.2.1. Estructura Organizacional

Figura 1.  
Organigrama



Nota. Organigrama del Gobierno Autónomo Municipal de Chulumani.

### **1.2.2. Antecedentes Internacionales**

- León y Guerrero, (2016) “SISTEMA DE CONTROL PARA COOPERATIVA DE AGUA POTABLE.” implementar un sistema de control para una cooperativa de agua potable, que tendrá funciones orientadas a la gerencia, administración y clientes de la Cooperativa de Agua Potable. La metodología de desarrollo de software define el marco de trabajo para estructurar, planificar y controlar el desarrollo de los sistemas de información. Para este proyecto el paradigma que se utilizó es la del modelo iterativo incremental, ya que éste se adecua de mejor manera a las características y tipo de software que se desarrolló. (Pontificia Universidad Católica De Valparaíso-Chile).
- Muñoz, (2015) “SISTEMA INFORMÁTICO PARA LA FACTURACIÓN Y CONTROL DE USUARIOS DE LA JUNTA ADMINISTRADORA DE AGUA POTABLE Y ALCANTARILLADO SANTANDEREANA DE LA PROVINCIA DE BOLÍVAR”. Diseñar un sistema informático de facturación y control de usuarios para la Junta Administradora de Agua Potable y Alcantarillado “Santandereana”- Provincial de Bolívar. Para el presente sistema se ha procedido a utilizar la Metodología de desarrollo de UML del mismo modo contiene construcciones organizativas para agrupar los modelos en paquetes, lo que permite a los equipos de software dividir grandes sistemas en piezas de trabajo, para entender y controlar las dependencias entre paquetes, y para gestionar las versiones de las unidades de modelo, en un ambiente de desarrollo complejo. (Universidad Técnica De Babahoyo-Ecuador).
- Mesías, (2012) “SISTEMA DE GESTIÓN UTILIZANDO SOFTWARE LIBRE PARA COBROS Y REGISTROS DE USUARIOS DE LA JUNTA DE AGUAS CHACÓN SEVILLA.” Desarrollar un sistema de gestión utilizando software libre para cobros y registros de usuarios de la Junta de Aguas Chacón Sevilla. Para el presente sistema se ha procedido a utilizar la metodología de desarrollo de software denominado Modelo de Cascada o Ciclo de Vida Clásico, dicho paradigma consiste en descomponer la actividad global en fases que se van desarrollando en forma lineal, es decir una detrás de la otra de tal forma que el

inicio de cada etapa debe esperar a la finalización de la inmediatamente anterior; esta metodología permite tener una mejor administración y llegar al objetivo propuesto. Para el análisis del sistema se utilizó UML (Lenguaje Unificado de Modelado) el cual es un conjunto de notaciones y diagramas para modelar sistemas orientados a objetos; dando diferentes perspectivas a un sistema. (Universidad Técnica De Ambato-Ecuador).

### **1.2.3. Antecedentes Nacionales**

- Rivas, (2019) “DISEÑO DE UN PROTOTIPO ELECTRÓNICO DE MEDICIÓN, CONTROL Y REGISTRO DE CONSUMO DE AGUA POTABLE PARA APLICACIÓN DOMÉSTICA”. El proyecto aporta el diseño de un prototipo electrónico que permite la lectura del consumo del servicio de agua potable. Para la elaboración del presente proyecto de grado se analizarán circuitos digitales, protocolos de comunicaciones, lenguajes de programación. Lo que nos induce a realizar diseños analíticos, así como pruebas experimentales, buscando que la parte teórica se relacione con la práctica. (Universidad Mayor De San Andrés)
- Aramayo, (2009) “SISTEMA DE FACTURACIÓN DE CONSUMO DE AGUA POTABLE PARA LA PROVINCIA INQUISIVI.” Desarrollar e implementar un sistema de facturación, para determinar de manera eficiente, oportuna y eficaz el pago por consumo de agua potable, esto se logrará por medio de metodologías de diseño y tecnologías de programación que estén de acuerdo con este propósito. Para el análisis y diseño del sistema se empleará la metodología OMT (Objet Modeling Technique) descrito por Rumbaugh y la herramienta de modelo UML como un lenguaje de apoyo (Universidad Mayor De San Andrés).
- Galarza, (2008) “SISTEMA DE COBRO DE AGUA POTABLE Y ALCANTARILLADO EN EPSA-ASOAPAL – Libre.” Desarrollar un sistema computacional de cobro de Agua Potable y Alcantarillado en EPSA-ASOAPAL para verificar que los usuarios cumplan con el pago respectivo por los servicios prestados utilizando metodología UP.

El manejo y control comprende el hecho de verificar si cada uno de los usuarios está al tanto con el pago por el servicio prestado, sin embargo, todo este proceso se lo realiza de forma manual es por esta razón que se requiere de un sistema informático que pueda automatizar tal proceso de manera que se pueda administrar de una forma más eficiente y eficaz. (Universidad Mayor De San Simón).

#### **1.2.4. Antecedentes locales**

- Ticona, (2020) “DESARROLLO DE UN SISTEMA WEB CONTROL DE REGISTRO DE ARCHIVOS NOTARIALES.” Diseñar e implementar un Sistema orientado a la Web que permita controlar y almacenar los registros de archivos y hacer un seguimiento de la información almacenada en la notaria número quince de la ciudad de El Alto. Para el desarrollo del software, análisis, implementación y documentación del presente proyecto se utilizará la metodología UWE. (Universidad Pública de El Alto).
- Huanca, (2020) “SISTEMA DE ADMINISTRACIÓN – FACTURACIÓN Y CONTROL DE INVENTARIO” Desarrollar un sistema de administración - facturación y control de inventario que permita la optimización del proceso de control de las entradas y salidas como el registro ventas facturadas de los productos que coadyuven a la eficiencia, logrando que el flujo de información sea rápido y oportuno. Para el desarrollo del sistema de administración, control de inventarios y facturación el recurso a utilizar en cuanto a software es PHP como lenguaje de programación MYSQL como gestor de base de datos (Universidad Pública de El Alto).

### **1.3. PLANTEAMIENTO DEL PROBLEMA**

El Gobierno Autónomo Municipal de Chulumani - Sud Yungas a experimentando un crecimiento poblacional, lo que ha significado un incremento de clientes en el servicio de agua potable del Municipio, de tal manera que la carencia de herramientas tecnológicas en la administración de información del cobro, pago como también en el proceso de atención a instalaciones, reparaciones y corte del servicio de agua potable, que de acuerdo a un análisis preliminar y amplio al manejo de información de

usuario/cliente en el Gobierno Autónomo Municipal de Chulumani se percibe que los procesos de: administración de información, facturación y registro se realizan de forma manual y sin ningún control, utilizando únicamente libros foliados.

Esta situación provoca conflictos, pérdidas económicas y descontento por la lentitud de los procesos, la relación con el cliente no es agradable ya que dentro de los requerimientos de la institución no se utilizan métodos de ingeniería de requerimientos.

### **1.3.1. Problema principal**

Debido al proceso manual de la administración de información y control de las recaudaciones del servicio de agua potable, el municipio posee una gran cantidad de libros con los registros de sus clientes y de las cuentas de ellos, además del registro de los trabajos que realizan los trabajadores en terreno como ser en el proceso de atención a instalaciones, reparaciones y corte, lo que dificulta es la comprensión y orden de la información. Por lo dicho anteriormente el Municipio presenta pérdidas de información en los registros de administración y control de pagos de agua generando demora en la elaboración de informes y reportes, lo cual ocasiona que la asignación de control y supervisión de los recibos por parte de la Unidad de Recaudaciones no sea adecuada.

### **1.3.2. Problemas específicos**

- El control y registro de recaudaciones del servicio de agua potable se elaboran en hojas de escritorio.
- Pérdida de documentos en el manejo inadecuado de los datos.
- Los registros y actualizaciones de los datos de los contribuyentes se guardan en planillas de Excel, como también en libros foliados, esto ocasionando pérdida de tiempo al momento de acceder y realizar consultas a las mismas.
- La información se encuentra descentralizada ocasionando desconocimiento del estado de los procesos de gestiones pasadas.

- Inexistencia de clasificación de información de los cobros, pagos y de proceso de atención a instalaciones, reparaciones y corte del suministro de Agua Potable.
- Carencia de un sistema de administración y control para el almacenamiento de información.

### **1.3.3. Formulación del problema**

¿De qué manera se podría mejorar la administración y controlar las recaudaciones de los recursos del suministro de Agua Potable y como acelerar los procesos para su óptimo funcionamiento y atención oportuna a los usuarios/clientes?

## **1.4. OBJETIVOS**

### **1.4.1. Objetivo General**

Desarrollar un “Sistema Web para la Administración de Información y Control de la Recaudación de Recursos del Suministro de Agua Potable”, para el Gobierno Autónomo Municipal de Chulumani, Unidad Dirección de Recaudaciones con el propósito de acelerar los procesos administrativos, brindar información actualizada y mejorar la calidad de servicio hacia el cliente.

### **1.4.2. Objetivos Específicos**

- Realizar el análisis y requerimientos de los procesos para el funcionamiento del sistema como ser: flujos de trabajo, roles y procesos principales.
- Desarrollar una Base de Datos única que permita almacenar datos e información en base a los requerimientos obtenidos de la Institución.
- Diseñar el sistema de control del servicio del agua potable conforme a las necesidades y requerimientos importantes según el diagnóstico.
- Generar reporte dinámico, para mejorar la atención a los clientes con la información correcta que se almacenara en un gestor de base de datos.

- Elaborar pruebas necesarias con el fin de verificar que el sistema cumpla con las expectativas deseadas.

## **1.5. JUSTIFICACIÓN**

### ***1.5.1. Justificación Técnica***

El proyecto ha sido desarrollado para Alcaldía del Gobierno Autónomo Municipal de Chulumani-Sud Yungas, se realizó por las necesidades que tiene la unidad Recaudaciones al no contar con ningún sistema de información para la ejecución de las tareas encomendadas por la Institución, optimizando así los servicios que presta la misma.

Para el presente proyecto, se dispone de distintas tecnologías que son necesarias como ser en la parte de Hardware: equipos de computación con una buena capacidad de memoria, procesamiento, conexiones de red, servicio de internet y ambiente adecuado, para poder llevar a cabo el desarrollo del sistema, y en la parte de Software se tienen instalado el sistema operativo Windows, con programas como: Office 2007, PDF, antivirus y otros programas.

### ***1.5.2. Justificación Económica***

El sistema reduce gastos por que la información se maneja en grandes volúmenes, que implica ahorro de tiempo, reducción de material de escritorio y reducción del trabajo manual.

Para la implementación del Sistema Web de Administración de Información y Control de la Recaudación de Recursos del Suministro de Agua Potable se augura un ahorro económico ya que el costo se reduce a la capacitación del personal para el manejo del sistema tomando en cuenta que el sistema ha sido desarrollado utilizando software libre, que implica la no adquisición de licencias.

### **1.5.3. Justificación Social**

La unidad de Recaudaciones del Gobierno Autónomo Municipal de Chulumani, recibe cada mes y año a nuevos clientes para la instalación, cobro y pago del agua potable, los datos de los mismo son manipulados en forma manual, almacenadas en material de escritorio, hojas de cálculo Excel, en este sentido el sistema coadyuvara al Usuario/Servidor en la administración y control, procesamiento y almacenamiento de una forma más efectiva para agilizar los procesos de cobros y lograr un control integral de los clientes, es decir proporcionará una mejor efectividad al flujo y procesamiento de los datos, mostrando confiabilidad en el manejo de datos e información, reducirá el trabajo excesivo del personal, reducirá los tiempos de respuesta en la emisión de consultas y reportes.

## **1.6. METODOLOGÍA**

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

### **1.6.1. UWE**

Es una metodología que permite especificar de mejor manera una aplicación Web en su proceso de creación mantiene una notación estándar basada en el uso de UML (Unified Modeling Lenguaje) para sus modelos y sus métodos, lo que facilita la transición. La metodología define claramente la construcción de cada uno de los elementos del modelo. En su implementación se deben contemplar las siguientes etapas y modelos München, (2016).

### **1.6.2. Método de Ingeniería**

#### **1.6.2.1. Metodología UWE (UML-BASED WEB ENGINEERING)**

UWE (UML-Based Web Engineering) es una metodología basada en UML y en el proceso unificado para el desarrollo de aplicaciones web. Esta propuesta está formada por una anotación para especificar el dominio (basada en UML) y un modelo para llevar a cabo el desarrollo del proceso de modelado. Los sistemas adaptativos y la sistematización son dos aspectos sobre los que se enfoca UWE.

### 1.6.2.2. Características UWE

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

- **Fases de la Metodología UWE:** UWE es una metodología dirigida o enfocada al modelado de aplicaciones Web, ya que está basada estrictamente en UML, esta metodología nos garantiza que sus modelos sean fáciles de entender para los que manejan UML.
- **Fase de análisis de Requisitos:** La Fase de Análisis de Requerimientos realiza la captura de los mismos mediante diagramas de casos de uso acompañado de documentación que detallada.
- **Fase Conceptual:** Caracterizado por un modelo de dominio, que utiliza los requisitos que se detallan en los casos de uso.
- **Fase de Navegación:** Este modelo se destaca en el marco de UWE como el más importante, ya que representa elementos estáticos, a la vez que se pueden incorporar lineamiento semántico de referencia para las funcionalidades dinámicas de una aplicación Web.

## 1.7. METRICAS DE CALIDAD

Existen numerosas publicaciones, artículos y conferencias sobre métricas del software donde se abordan los aspectos técnicos de la medición; pero no existen guías claras para la implementación de programas de métricas en las empresas. Definir métricas que se quieran implementar en un proceso de medición para conseguir medir la calidad de un producto o proceso de software, no es algo trivial. Varas y Pradenas, (2000)

### 1.7.1. Control de seguridad ISO/IEC 27001

ISO/IEC 27001, en su apartado 0.3 Compatibilidad con otros sistemas de gestión, asegura que esta norma internacional sigue las pautas marcadas en las normas ISO 9001:2000 e ISO 14001:2004 para asegurar una implementación

integrada y consistente con las mencionadas normas de gestión. Mesquida, ( p. 25 y 34).

## 1.8. HERRAMIENTAS

- **Base de Datos**

- **PostgreSQL** es un potente sistema de base de datos relacional de objetos de código abierto que utiliza y amplía el lenguaje SQL combinado con muchas características que almacenan y escalan de forma segura las cargas de trabajo de datos más complicadas. PostgreSQL se ha ganado una sólida reputación por su arquitectura probada, confiabilidad, integridad de datos, conjunto de características sólidas, extensibilidad y la dedicación de la comunidad de código abierto detrás del software para ofrecer constantemente soluciones innovadoras y de alto rendimiento.

- **Editor de Base de Datos**

- **Navicat** es una herramienta de desarrollo de bases de datos que le permite conectarse simultáneamente a bases de datos MySQL, MariaDB, MongoDB, SQL Server, Oracle, **PostgreSQL** y SQLite desde una sola aplicación.

- **Framework de desarrollo**

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

- **Editor de texto**

- **VSCode** es un editor de código fuente que permite trabajar con diversos lenguajes de programación, admite gestionar tus propios atajos de teclado y

refactorizar el código. Es gratuito, de código abierto y nos proporciona una utilidad para descargar y gestionar extensiones con las que podemos personalizar y potenciar esta herramienta. Y sus extensiones nos otorgan infinidad de opciones, como colorear tabulaciones, etiquetas o recomendaciones de autocompletado. También hay extensiones que nos ayudan con el lenguaje de programación que vayamos a usar, como por ejemplo para Python, C / C++, JavaScript y demás.

- **Bootstrap** es uno de los framework más populares para el desarrollo del frontend de páginas web. La herramienta proporciona plantillas para CSS y HTML que facilitan la colocación y el diseño de la página.
- **JQuery**
  - **JQuery** también conocido como **JavaScript**, es un software libre y de código abierto que posee un doble licenciamiento bajo la licencia MIT y la Licencia Publica General de GNUv2, permitiendo su uso en proyectos libres y privativos. Es compatible con los navegadores Mozilla Firefox 2.0+, Internet Explorer 6+, Safari 3, Opera 10.6+ y Google Chrome 8+.
- **PHP**
  - **PHP** (acrónimo de PHP: Hypertext Preprocessor), es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor. Es un lenguaje multiplataforma. Completamente orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una Base de Datos. Es libre, por lo que se presenta como una alternativa de fácil acceso para todos y permite aplicar técnicas de programación orientada a objetos.
  - **Servidor Apache** WampServer de Windows es un entorno de desarrollo Web. Le permite crear aplicaciones web con Apache, PHP y la base de datos MySQL. También viene con SQLiteManager PHP MyAdmin para que administre sus más fácilmente sus bases de datos.
  - **JavaScript** es el lenguaje de programación encargado de dotar de mayor interactividad y dinamismo a las páginas web. Cuando JavaScript se ejecuta en el navegador, no necesita de un compilador. El navegador lee directamente el código, sin necesidad de terceros. Por tanto, se le reconoce como uno de los

tres lenguajes nativos de la web junto a HTML (contenido y su estructura) y a CSS (diseño del contenido y su estructura).

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

## **1.9. LIMITES Y ALCANCES**

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

### **1.9.1. Límites**

El Presente proyecto analizará la situación actual y se desarrollará de acuerdo a las necesidades, especificadas y requeridas dentro la unidad de Recaudaciones del Municipio.

- El sistema no contempla el control del personal.
- No contempla áreas contables.
- No contempla a otras Unidades e Direcciones de la Institución, como ser Dirección Administrativa Financiera, Archivos y Almacenes.
- Los pagos no se realizan en línea a través de una cuenta de banco.

### **1.9.2. Alcances**

El sistema tiene los siguientes módulos:

- Módulo Administrador, se registrará los datos del personal, se otorgarán permisos de acceso a los usuarios.
- Módulo de registro de Instalación del servicio de agua potable.
- Módulo de control de Cobro y Pago de agua.
- Módulo de gestión de Información.
- Módulo de Reportes.

## **1.10. APORTES**

El aporte que brinda el sistema informático a la administración de información y control de la recaudación de recursos del suministro de agua, es en automatizar los procesos

rutinarios y optimizar los tiempos de ejecución generando información correcta y centralizada en la base de datos, esto con el propósito de coadyuvar a la dirección de Recaudaciones del Municipio de Chulumani

# **CAPÍTULO II**

## **MARCO TEÓRICO**

## **2. MARCO TEÓRICO**

### **2.1. INTRODUCCIÓN**

En el desarrollo de los sistemas han evolucionado tanto en su uso con la automatización de los procesos operativos de las organizaciones, como apoyo al nivel operativo, así proporcionándose la información como una base para el proceso de toma de decisiones, como todos sabemos la tecnología es parte del desarrollo, porque transforma las empresas y cambia completamente su estructura.

Asimismo, para su completo entendimiento, se describe el desarrollo del sistema, por los conceptos más relevantes sobre las metodologías, métodos y herramientas utilizadas, con una descripción detallada de cada uno de los elementos de la teoría que será directamente utilizados en el desarrollo del proyecto de grado.

Para comprender el título del proyecto, tenemos como referencia los siguientes conceptos y definiciones.

### **2.2. DEFINICIÓN DE TÉRMINOS**

#### **2.2.1. Sistema**

Sistema es un elemento completo donde cada uno de sus componentes se vincula con otro componente, tanto de forma material como conceptual. Los métodos tienen composición, organización y dominio, pero únicamente los sistemas materiales tienen mecanismos y solamente algunos de ellos tienen figura o configuración. Martínez (2022)

Sistema es un conjunto de elementos (partes u órganos componentes del sistema) dinámicamente relacionados, en interacción que desarrollan una actividad (operación o proceso del sistema) para lograr un objetivo o propósito (finalidad del sistema), operando con datos, energía o materia (insumos o entradas de recursos necesarios para poner en marcha el sistema), unidos al ambiente que rodea el sistema (con el cual se relaciona dinámicamente), y para suministrar información, energía o materia (salidas o resultados de la actividad del sistema). Chiavenato I. , (1999)

Un sistema es una colección de componentes interrelacionados que trabajan conjuntamente para cumplir algún objetivo, Sommerville , (2005).

### **2.2.2. Sistema Web**

Según el autor Barners, (1990) es un sistema de distribución de información basado en hipertexto o hipermedios enlazados y accesibles a través de Internet.

Según Moran, ( 2001) Programación en Internet: Clientes WEB, menciona que una aplicación web (web-based-application) es un tipo especial de aplicación cliente/servidor, donde tanto el cliente (el navegador, explorador o visualizador) como el servidor (el servidor web) y el protocolo mediante el que se comunica (Hyper Text Transfer Protocol (HTTP) están estandarizados y no han de ser creados por el programador de aplicaciones.

Se denomina así a un conjunto de elementos orientados a Internet, ordenadamente relacionados entre sí que aporta a la organización en su funcionamiento, dando así la información necesaria para el cumplimiento de sus fines u objetivos, obteniéndola con procesar y almacenar información dinámicamente.

### **2.2.3. Administración**

La Administración es el conjunto de las funciones o procesos básicos planificar, organizar, dirigir y controlar el uso de los recursos y las actividades de trabajo con el propósito de lograr los objetivos o metas propuestas de la organización de manera eficiente y eficaz. Permite implementar nuevas estrategias para el logro de las metas para tener éxito en la empresa. Periañez, (2004)

La administración como el proceso de plantear, organizar, liderar y controlar el trabajo de los miembros de la organización y de utilizar todos los recursos disponibles de la empresa para alcanzar los objetivos organizacionales establecidos. Stoner y Freeman R, (1995)

Por lo tanto, la administración financiera es una disciplina que optimiza los recursos financieros para el logro de los objetivos de la organización con mayor eficiencia y rentabilidad. Robles , (2012)

#### **2.2.4. Sistema Web de administración**

Un sistema de información se destaca por su diseño, facilidad de uso, flexibilidad, mantenimiento automático de los registros, apoyo en toma de decisiones críticas y mantener el anonimato en informaciones no relevantes. Chen, (2019)

#### **2.2.5. Información**

Es un conjunto de datos con un significado, o sea, que reduce la incertidumbre o que aumenta el conocimiento de algo. En verdad, la información es un mensaje con significado en un determinado contexto, disponible para uso inmediato y que proporciona orientación a las acciones por el hecho de reducir el margen de incertidumbre con respecto a nuestras decisiones. Chiavenato, (2006, p. 110)

La información está constituida por un grupo de datos ya supervisados y ordenados, que sirven para construir un mensaje basado en un cierto fenómeno o ente. La información permite resolver problemas y tomar decisiones, ya que su aprovechamiento racional es la base del conocimiento. Perez Porto, (2012)

Es un fenómeno que proporciona significado o sentido a las cosas. En sentido general la información es un conjunto organizado de datos procesados, que constituyen un mensaje sobre un determinado ente o fenómeno. Perez y Gardey, (2012)

#### **2.2.6. Administración de la Información**

los sistemas de información administrativa (SIA). La definición de SIA varía. Aquí se considera como un sistema formal de recopilación, procesamiento y comparación, análisis y divulgación de información interna y externa a la empresa, de una manera oportuna, eficaz y eficiente para apoyar a los administradores en sus trabajos. El SIA tiene que ser preparado de acuerdo con necesidades específicas y comprende información de rutina, como informes mensuales, información que señale excepciones, en especial en los puntos críticos, e información necesaria para predecir el futuro. Koontz et al., (1991)

Sistema de información administrativa Sistema formal de recopilación, procesamiento, comparación, análisis y divulgación de información interna y externa a la empresa de una manera oportuna, eficaz y eficiente para apoyar a los administradores en sus trabajos. Koontz y Weihrich, (2013)

### **2.2.7. Control**

El Control es una función de administración que se elabora para asegurar que los hechos concuerden con los planes. Para que sea eficaz, se debe enfocar al presente, se debe centrar en la corrección y no en el error, debe, asimismo, ser específico, de tal forma que se concentre en los factores claves que afecten los resultados. es universal y abarca todas las fases de la empresa. Scanlan, (1987)

El control es un elemento de proceso administrativo; este incluye todas las actividades empleadas para garantizar que las operaciones reales coincidan con las operaciones planificadas. Robbins, (1996)

El control es un elemento del proceso administrativo que incluye todas las actividades que se emprenden para garantizar que las operaciones reales coincidan con las operaciones planificadas. Todos los gerentes de una 31 organización tienen la obligación de controlar; Por ejemplo, tienen que realizar evaluaciones de los resultados y tomar las medidas necesarias para minimizar las ineficiencias. De tal manera, el control es un elemento clave en la administración. Suárez, (2009)

### **2.2.8. Recaudación**

En un concepto genérico que engloba todas las actividades que realiza el estado para sufragar sus gastos, los tributos constituyen prestaciones exigidas obligatoriamente por él, esto en virtud de sus potestad de imperio, para atender a sus necesidades y llevar a cabo sus fines, para ello se vale de la actividad financiera denominada recaudación, la cual consiste en haber valer las normas jurídicas impositivas y exigir a la población su contribución para sufragar el gasto necesario para dar cumplimiento a las metas por las cuales fue creado el estado. Guillermina, (2003)

### **2.2.9. Suministro**

Define a la Cadena de Suministros como una red de organizaciones que están relacionadas a través de las conexiones downstream o corriente abajo (en la dirección de creación de un producto o servicio) y un corriente arriba (en la dirección de los proveedores) en los diferentes procesos y actividades que producen valor en la forma de productos y servicios finales ofrecidos al consumidor final. Cristobal, (1998)

Cadena de suministro es la porción de la cadena de valor que se enfoca principalmente en el desplazamiento físico de bienes y materiales y en el soporte de los flujos de información y transacciones financieras a través de los procesos de suministro, producción y distribución. Allen Collier y Evans , (2019)

El suministro es la dotación de un bien, el cual debe pasar por todo un proceso para ir desde el productor hasta el consumidor o beneficiario final.

#### **2.2.10. Recaudación del Suministro**

A efectos de analizar la evolución de la recaudación se procede a clasificar la misma de acuerdo a diferentes criterios de agregación, realizando comparaciones interanuales y con información externa. Con periodicidad trimestral se monitorea la evolución de los agregados macroeconómicos y sus estimaciones. Salim, (2009)

La importancia de recaudar impuestos para satisfacer las exigencias de la colectividad ha estado presente en el progreso en todas las formas en el país. Desde el punto de vista de la conciencia, no es idóneo recaudar sin una dirección clara, desde la perspectiva legal, es indispensable recaudar los impuestos con una aprobación voluntaria por parte de los contribuyentes.

#### **2.2.11. Suministro Agua Potable**

El suministro de agua potable consiste en la distribución del agua desde las estaciones de tratamiento de agua potable o desde los depósitos de servicio a los usuarios, a través de las redes de abastecimiento público. El agua suministrada a la red se clasifica en agua registrada y agua no registrada. Anonimo, (s.f.)

Además, el suministro de agua dulce de que dispone la humanidad se está reduciendo a raíz de la creciente contaminación de muchos de esos recursos hídricos. En algunos países los lagos y ríos se han transformado en receptáculos de una variedad abominable de desechos, inclusive aguas negras municipales parcialmente tratadas, efluentes industriales tóxicos y sustancias químicas de las actividades agrícolas en las aguas de superficie y freáticas. Pincolini, (2012, p. 12)

### **2.3. INGENIERÍA DE REQUERIMIENTOS**

La ingeniería de requerimientos (IR) cumple un papel primordial en el proceso de producción del software, ya que se enfoca en un área fundamental, la definición de lo que se desea producir, su principal tarea consiste en la generación de especificaciones correctas que describan con claridad, sin antigüedades, en la forma consistente y compacta las necesidades de los usuarios o clientes de esta manera se pretende minimizar los problemas relacionados por mala gestión de los requerimientos en el desarrollo de sistemas. Los requerimientos de software pueden dividirse en dos categorías requerimientos funcionales y no funcionales. Chavez, (2005, p. 3)

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

En otras palabras, es el proceso de recopilar, analizar y verificar las necesidades del cliente o usuario para un software es llamado ingeniería de requerimientos. La meta de la ingeniería de requerimientos es entregar una especificación de requisitos de software correcta y completa.

#### **2.3.1. Actividades de la Ingeniería de Requerimientos**

Las actividades que se llevan a cabo en la ingeniería de requerimientos se lleva un modelo de proceso de software utilizado para el ciclo de desarrollo, estas varían según el número como nombres, la Tabla 1 muestra algunos ejemplos de las actividades identificadas para cada proceso.

**Tabla 1.**

*Actividades de la ir para diferentes modelos de procesos de ingeniería de software*

MODELO	Oliver and Steiner 1996	EIA / IS-632	IEEE Std 1220- 1994	CMM nivel Repetitivo (2)	RUP
<b>Actividades</b>	Evaluar la información disponible	Análisis de requerimientos	Análisis de Requerimientos	Identificación de requerimientos	Análisis del problema
	Definir métricas efectivas	Análisis funcional	Estudio de los requerimientos	Identificación de restricciones del sistema a desarrollar	Comprender las necesidades de los involucrados
	Crear un modelo del comportamiento del sistema	Síntesis	Validación de requerimientos	Análisis de los Requerimientos	Definir el sistema
	Crear un modelo de los objetos	Análisis y control del sistema	Análisis funcional	Representación de los requerimientos	Analizar el alcance del proyecto
	Ejecutar el análisis		Evaluación y estudio de funciones	Comunicación de los requerimientos	Modificar la definición del sistema
	Crear un plan secuencial de construcción y pruebas		Verificación de funciones	Validación de requerimientos	Administrar los cambios de requerimientos

*Nota.* Adaptado de *Ingeniería de requerimientos*, (pág. 22), por Cruz Valle, (2005).

### **2.3.2. Requerimientos funcionales**

Esta se expresa en los servicios que debe proporcionar el software, de la manera en que éste debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares. En algunos casos, los requerimientos funcionales de los sistemas también pueden declarar explícitamente lo que el sistema no debe hacer

Los requerimientos funcionales deben ser:

- **Precisos** la ambigüedad a la hora de definir los requerimientos.

- **Completos** deben incluir la descripción de todos los servicios y características
- **Consistentes** no puede a haber contradicciones en la descripción

### 2.3.3. *Requerimientos no funcionales*

Estas se aplican al software en su totalidad se describen una restricción sobre el sistema que limita nuestras elecciones en la construcción de una solución al problema. Restringen los servicios o funciones ofrecidas por el sistema. Incluyen restricciones de tiempo, el tipo de proceso de desarrollo a utilizar, fiabilidad, tiempo de respuesta y capacidad de almacenamiento.

La carencia de consenso se ve reflejada en la diversidad de términos utilizados en las definiciones, con significados poco claros, que dan lugar a la ambigüedad sobre su alcance o lo que representan los mismos; las discrepancias conceptuales también pueden encontrarse en las clasificaciones propuestas para Requerimientos No Funcionales, que incluyen árboles o categorías de conceptos que se muestran en forma separada en otras definiciones. Silvana del Valle, (2012)

**Figura 2.**

*Clasificación de requerimientos no funcionales*



*Nota.* Adaptado de *Requerimientos No funcionales para aplicaciones Web (Doctoral dissertation, Universidad Nacional de La Plata).*, ( págs. 16, 92).

## **2.4. MODELO ITERATIVO E INCREMENTAL**

### ***2.4.1. Modelo Iterativo e Incremental***

El modelo iterativo-incremental (MII) se ha constituido en una base para la generación de nuevas metodologías y prácticas en el desarrollo de software, y ha sido base para la evolución de nuevas propuestas de desarrollo en una era en la que el factor de cambio en las necesidades de información es cada vez más frecuente y juega un papel preponderante y constante, por lo que se requiere un alto grado de flexibilidad y capacidad de reacción ante las expectativas y necesidades que surgen de forma natural. Solano Fernández y Porras Alfaro, (p. 165 y 167)

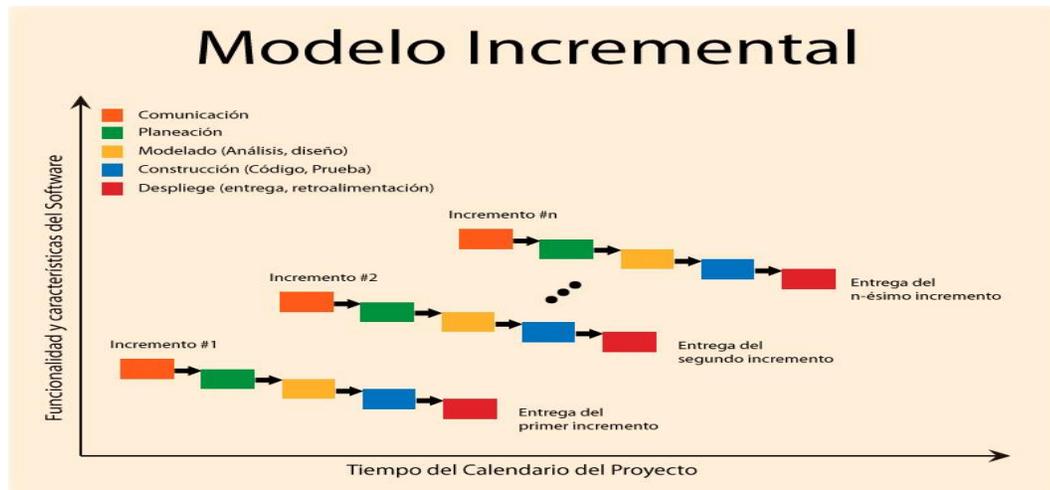
El desarrollo iterativo e incremental es un proceso de desarrollo de software cíclico en respuesta a la debilidad del modelo en cascada. Empieza con una planificación inicial y termina con el despliegue con la iteración cíclica en el medio

### ***2.4.2. Modelo Incremental***

El modelo incremental combina elementos del modelo en cascada con la filosofía interactiva de construcción de prototipos. Se basa en la filosofía de construir incrementando las funcionalidades del programa. Este modelo aplica secuencias lineales de forma escalonada mientras progresa el tiempo en el calendario. Cada secuencia lineal produce un incremento del software. Información: Ingeniería del Software. Pressman R. S., (2005).

**Figura 3.**

*Modelo incremental*



*Nota.* Adaptado de *Ingeniería de Software*, (pág. 23), por *Pressman R. S.*, (2005)

## **Ventajas**

Entre las ventajas que puede proporcionar un modelo de este tipo encontramos las siguientes:

Mediante este modelo se genera software operativo de forma rápida y en etapas tempranas del ciclo de vida del software.

- Es un modelo más flexible, por lo que se reduce el coste en el cambio de alcance y requisitos.
- Es más fácil probar y depurar en una iteración más pequeña.
- Es más fácil gestionar riesgos.
- Cada iteración es un hito gestionado fácilmente.

### **2.4.3. Modelo Iterativo**

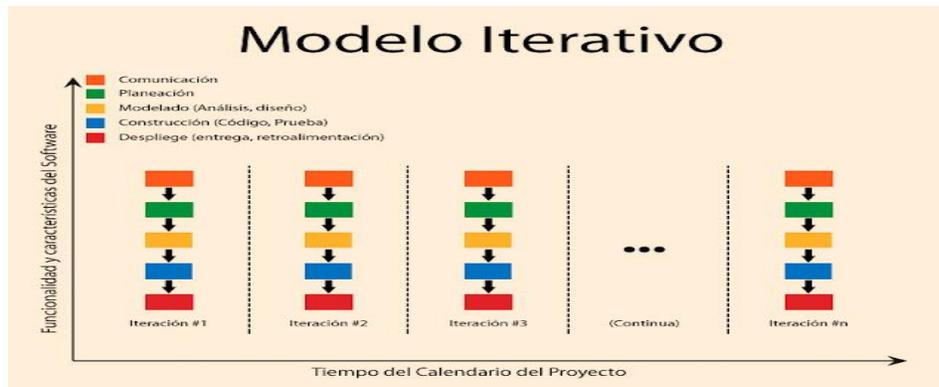
Es un modelo derivado del ciclo de vida en cascada. Este modelo busca reducir el riesgo que surge entre las necesidades del usuario y el producto final por malos entendidos durante la etapa de recogida de requisitos.

Consiste en la iteración de varios ciclos de vida en cascada. Al final de cada iteración se le entrega al cliente una versión mejorada o con mayores funcionalidades del producto. El cliente es quien después de cada iteración evalúa el producto y lo corrige

o propone mejoras. Estas iteraciones se repetirán hasta obtener un producto que satisfaga las necesidades del cliente. Pressman, (2005)

**Figura 4.**

*Desarrollo iterativo e incremental*



*Nota.* Adaptado de *Metodologías de programación MODELO ITERATIVO INCREMENTAL*, por Gomez Castro, (2018)

Las ventajas del desarrollo iterativo son, entre otras, las siguientes:

- Software que funciona desde el principio.
- Entrega valor de negocio con cada iteración.
- Se adapta a los cambios en los requisitos a lo largo del desarrollo.
- El esfuerzo de diseño se gasta sólo para lo que se construirá a continuación.
- Capacidad para corregir el rumbo en cada iteración.
- Funciona bien con los clientes que necesitan algo para patear para averiguar lo que necesitan.
- Gestión de riesgos más fácil y sencilla. Los riesgos pueden ser identificados y abordados por iteración.

Los inconvenientes del desarrollo iterativo son:

- La arquitectura del sistema necesita una atención proactiva y puede requerir cambios fundamentales en el camino.

## 2.5. INGENIERÍA WEB

La Ingeniería Web surge como una nueva disciplina orientada a solucionar los problemas derivados de una proliferación de sistemas web de baja calidad, realizados

con una carencia completa de proceso. Esta nueva disciplina identifica nuevos elementos propios de las aplicaciones web que no se cubren en las Ciencias de la Computación, en la Ingeniería del Software o en los Sistemas de Información.

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

Según Guevara, (2007), la Ingeniería Web (Web) aplica “sólidos principios científicos, de ingeniería y de administración, y enfoques disciplinados y sistemáticos para el desarrollo, despliegue y mantenimiento exitoso de sistemas y aplicaciones basados en Web de alta calidad” (p. 13) .

La Ingeniería Web intenta evitar el caos existente en el desarrollo de sistemas basados en la Web:

- Controlar el proceso
- Minimizar riesgos
- Potenciar la calidad y la capacidad de mantenimiento

La ingeniería web se debe al crecimiento desenfrenado que está teniendo la Web está ocasionando un impacto en la sociedad y el nuevo manejo que se le está dando a la información en las diferentes áreas en que se presenta ha hecho que las personas tiendan a realizar todas sus actividades por esta vía.

### **UWE (UML - Based Web Engineering)**

UWE es una metodología basada en el Proceso Unificado y UML para el desarrollo de aplicaciones Web, cubre todo el ciclo de vida de las aplicaciones Web. Su proceso de desarrollo se basa en tres frases principales: la fase de captura de requisitos, la fase de análisis y diseño y la fase de la implementación.

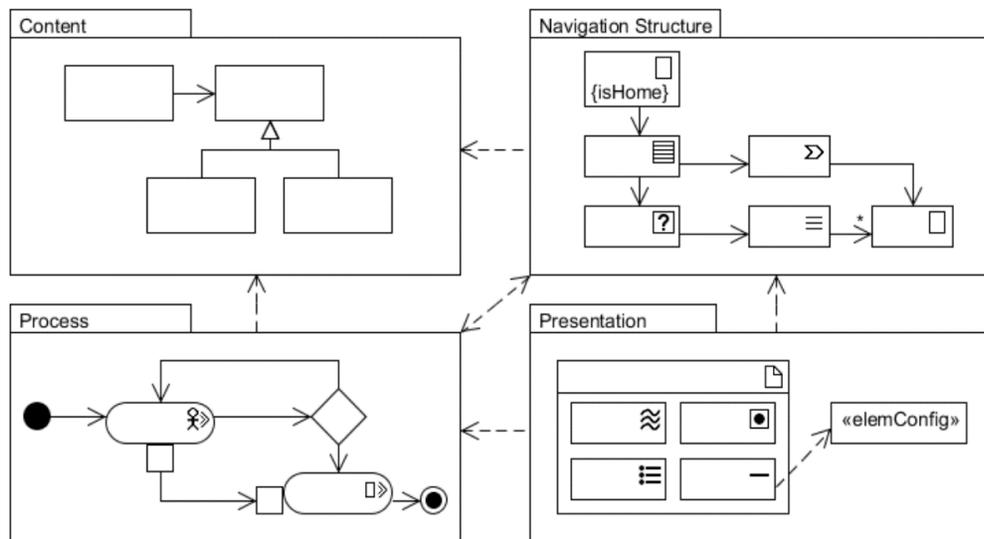
La metodología para el desarrollo de un servicio web es un conjunto de actividades llevadas a cabo para hacer más eficiente el desarrollo y aplicación de un servicio web ya que indica cómo hay que obtener los distintos productos parciales y finales. Es esencial planificar las actividades que se llevarán a cabo, unificar los criterios para el desarrollo y tener puntos de control y revisión.

Debemos también destacar las características relevantes del proceso UWE como la utilización del paradigma orientado a objetos, su orientación al usuario.

En la siguiente figura podemos ver una vista general de UWE, con las fases que tiene como.

**Figura 5.**

*Vista General de las Transformaciones entre modelos de UWE.*



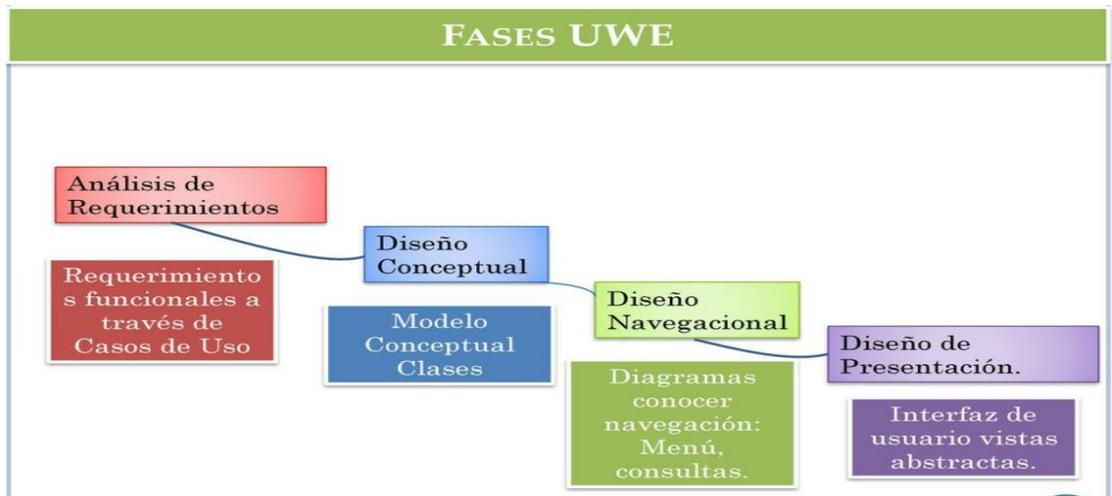
*Nota. Adaptado de (UWE - UML-based WEB Engineering), por Anonimo, (2012)*

### 2.5.1.1. Fases de la UWE

UWE cubre todo el ciclo de vida de este tipo de aplicaciones centrando además su atención en aplicaciones personalizadas o adaptativas.

**Figura 6.**

*Fases de la Metodología UWE*



*Nota. Adaptado de (UWE - UML-based WEB Engineering).*

### **2.5.2. Captura, análisis y especificación de requisitos**

En este proceso, iterativo e incremental, incluye flujos de trabajo y puntos de control, y sus fases coinciden con las propuestas en el proceso unificado de modelado Guerrero, (2014). La UWE se centra en capturar las necesidades y expectativas de los usuarios para desarrollar sistemas que satisfagan sus requerimientos.

- **Identificación de autores:** Un Actor es usuario que puede tener diversos roles y que participa en la interacción con el sistema o aplicación. Un actor no necesariamente es una persona.
- **Requerimientos funcionales:** Describen las funciones y comportamientos específicos que se esperan del sistema. Su función principal es definir qué debe hacer el sistema, los servicios que debe proporcionar y cómo debe reaccionar a entradas específicas.
- **Requerimientos no funcionales:** Se utiliza para capturar las propiedades o las características del sistema, el rendimiento, disponibilidad, seguridad entre otros. Se define a través de tablas que

contengan los “requerimientos no funcionales” y función “restricciones del sistema”.

### **2.5.3. Diseño del sistema**

El diseño del sistema se centra en traducir los requisitos funcionales y no funcionales que implica la creación de modelos detallados que representan como las funciones estas son definidas en los requisitos que se cumplirán, bajo a la estructura de la aplicación web.

#### **2.5.3.1. Diagrama de Casos de Usos.**

Un caso de uso es una descripción de las acciones de un sistema desde el punto de vista del usuario.

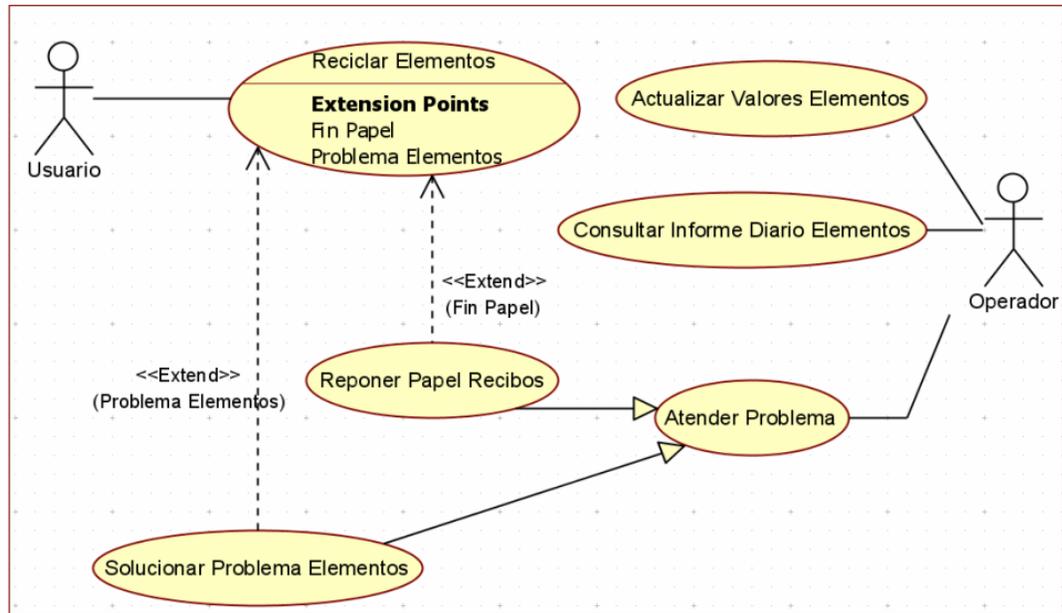
Los Casos de Uso:

- Describen el comportamiento de un sistema desde el punto de vista del usuario.
- Permiten definir los límites del sistema y las relaciones entre el sistema y el entorno.
- Particionan el conjunto de necesidades atendiendo a la naturaleza de los usuarios que participan en el mismo.
- El usuario debería poder entenderlos para realizar su validación.
- Guían todo el proceso de desarrollo del sistema.

El modelo de casos de uso emplea el concepto de actores, para representar los roles que los usuarios pueden jugar en el sistema y los casos de uso para representar lo que los usuarios pueden hacer con el sistema.

**Figura 7.**

*Diagrama de Casos de Uso.*



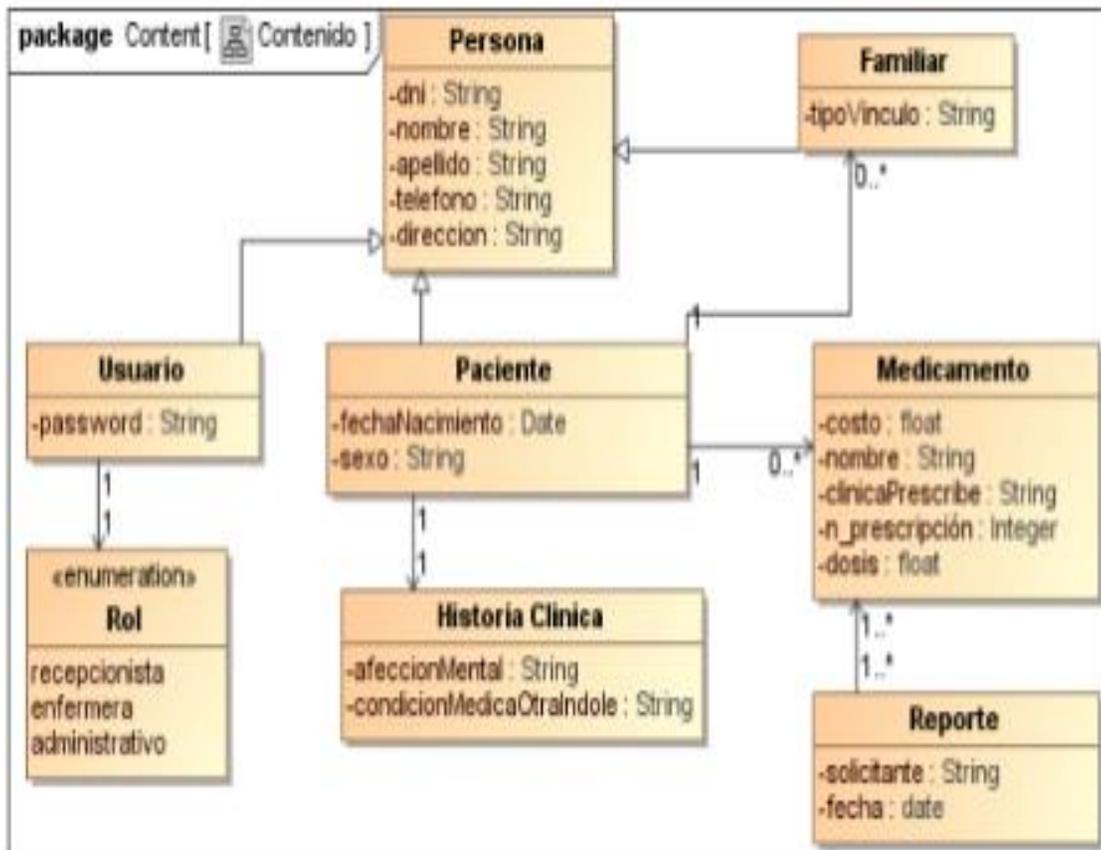
*Nota.* Adaptado de INGENIERÍA DEL SOFTWARE I, (p. 23), por López, (s.f).

### **2.5.3.2. Modelo Conceptual**

El modelo de contenido es el modelo conceptual tomando en cuenta los requerimientos especificados en los casos de uso se representa en un diagrama de clases. En él se representa la información del dominio, sus datos persistentes, mediante un diagrama de clases UML. En la figura 4 se presenta el modelo de contenido del MHC-PMS mediante un diagrama de clases de UML, donde se puede observar las distintas clases que forman parte del sistema con sus respectivos atributos y estereotipos definido por UWE.

Figura 8.

Modelo de Contenido (UWE): MHC-PMS



Nota. Adaptado de (Un caso de estudio sobre la aplicación de UWE para la generación de Sistemas Web, p. 7) , por Rotta et al., ( 2016)

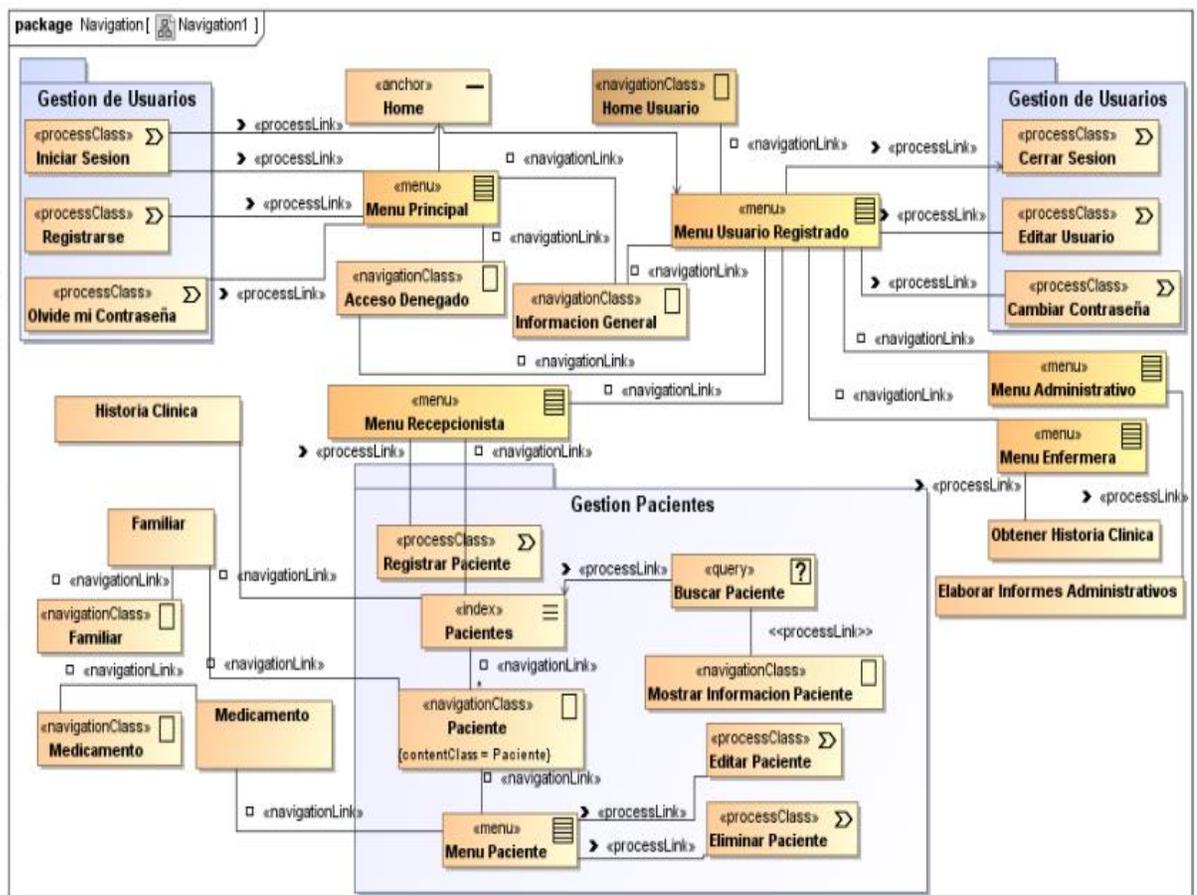
### 2.5.3.3. Modelo de navegación

El modelo de navegación de una aplicación web comprende la especificación de qué objetos pueden ser visitados mediante la navegación, Mediante estos diagramas se representa el diseño y la estructura de las rutas de navegación al usuario.

Tiene como objetivo la representación de nodos y enlaces de la estructura de hipertexto, y el diseño de las rutas de navegación mediante diagramas de clases UML.

**Figura 9.**

*Modelo de Navegación (UWE): MHC-PMS.*



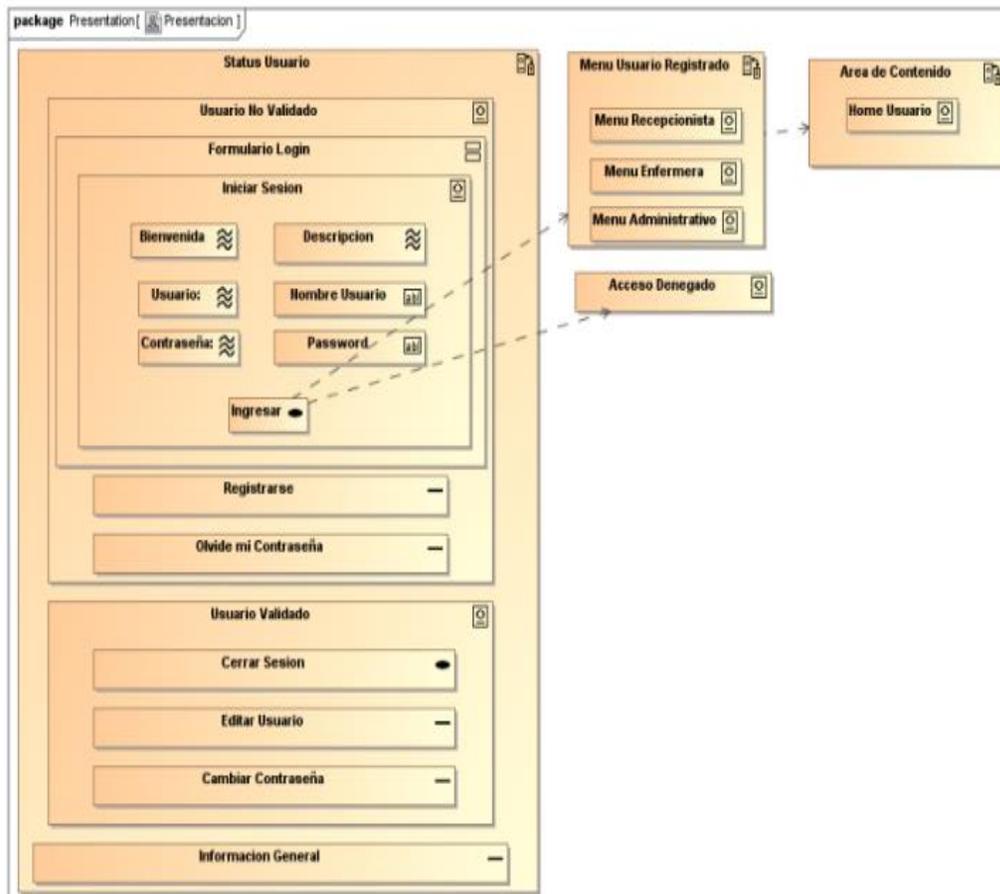
*Nota.* Adaptado de *Un caso de estudio sobre la aplicación de UWE para la generación de Sistemas Web*, (p. 8), por Rotta et al., (2016).

#### 2.5.3.4. Modelo de presentación

En el modelo de presentación representa el layout subyacente a los modelos de navegación y procesos mediante una presentación abstracta, ya que una presentación concreta requeriría la especificación de propiedades físicas adicionales. Utiliza un diagrama de clases de UML para modelar la estructura.

**Figura 10.**

*Modelo de Presentación (UWE): MHC-PMS.*



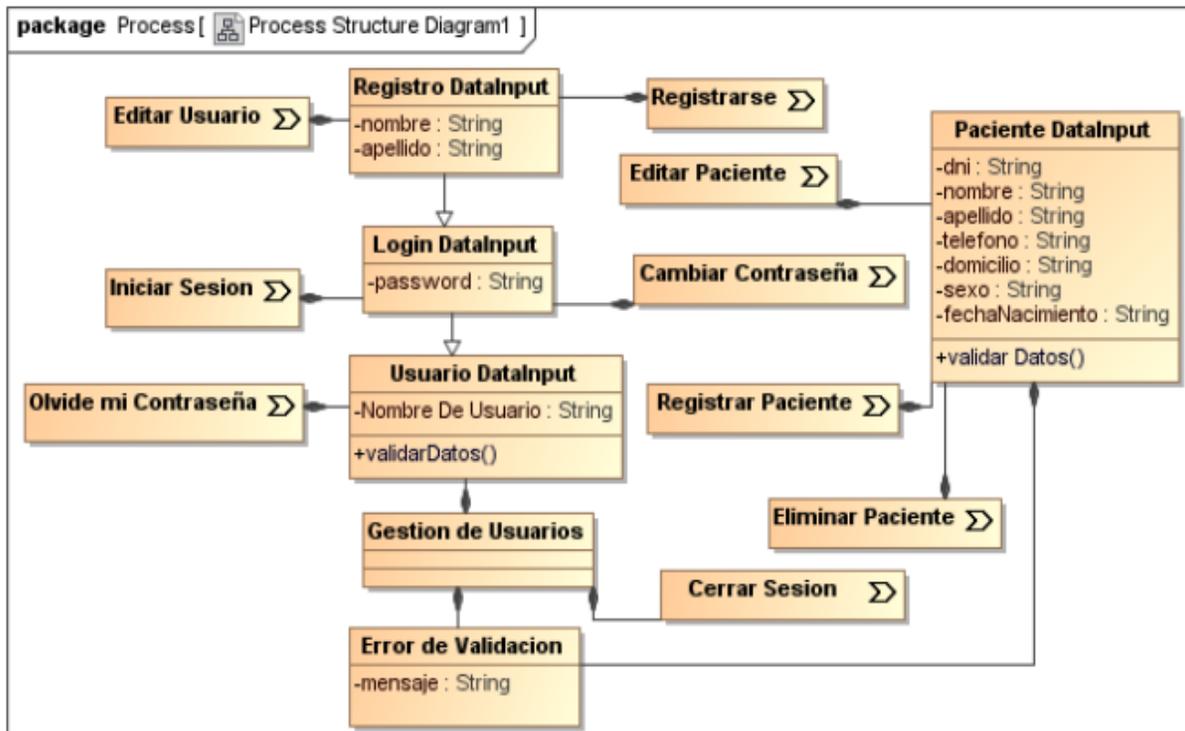
*Nota.* Adaptado de *Un caso de estudio sobre la aplicación de UWE para la generación de Sistemas Web*, (p. 9), por *Rotta et al.*, (2016).

### **2.5.3.5. Modelo de procesos**

En el modelo de procesos se representan los aspectos dinámicos de la aplicación Web y especifica funcionalidad como transacciones y workflows de actividades. Se modela mediante un diagrama de actividades de UML, y es resultado de refinar el diagrama de actividades modelado durante la especificación de requerimientos. Muestra el flujo de la ejecución representado por nodos de actividad conectados, los nodos de control que proveen constructores de flujo de control como decisiones y sincronización y nodos de objetos que representan el flujo de datos.

Figura 11.

Modelo de Estructura del Proceso (UWE): MHC-PMS.



Nota. Adaptado de *Un caso de estudio sobre la aplicación de UWE para la generación de Sistemas Web*, (p. 10), por Rotta et al., (2016)

#### 2.5.4. Codificación del sistema

La codificación del sistema es una fase fundamental en el desarrollo del software, que consiste, en llevar a código fuente, así como las habilidades solidas de programación en el lenguaje elegido diseñando todas las fases anteriores.

#### 2.5.5. Pruebas

Estas pruebas son vitales para el proceso del desarrollo del software, en las pruebas de caja blanca y caja negra que son utilizadas en la actualidad, dando así que en las pruebas de la caja blanca están destinadas para realizar pruebas estructurales del código. Como ser las pruebas de caja negra son dedicadas en evaluar las entradas y salidas de datos, sin importar el código.

### **2.5.6. Instalación o fase de Implementación**

En esta fase de la instalación o implementación, busca asegurar que el software desarrollado se integre exitosamente en el entorno operativo, garantizando así la función de cumplir de manera efectiva, con las necesidades y expectativas de los usuarios, lo cual sigue la codificación del sistema, donde el software ha sido desarrollado y se ha completado la fase de pruebas.

### **2.5.7. Mantenimiento**

Es el proceso de control, mejora y optimización del software ya desarrollado e instalado, que también incluye depuración de 25 errores y defectos que puedan haberse filtrado de la fase de pruebas de control. Apaza Bilbao, (2021)

## **2.6. LENGUAJE UNIFICADO DE MODELADO UML**

El lenguaje unificado de modelado es abreviado y reconocido como UML. Es la notación gráfica utilizada para visualizar, diseñar e implementar un sistema de software, incluyendo los procesos e interacciones que forman el sistema. Cubero, (2020)

El lenguaje UML sirve como una herramienta versátil para la representación visual y la comunicación de sistemas, lo que ayuda en todas las etapas del ciclo de vida del software, desde la concepción y diseño hasta la implementación y mantenimiento.

La finalidad de los diagramas es presentar diferentes perspectivas del sistema, se les conoce como modelo que es una presentación simplificada de la realidad.

Diagramas comunes que se utilizaran: 30

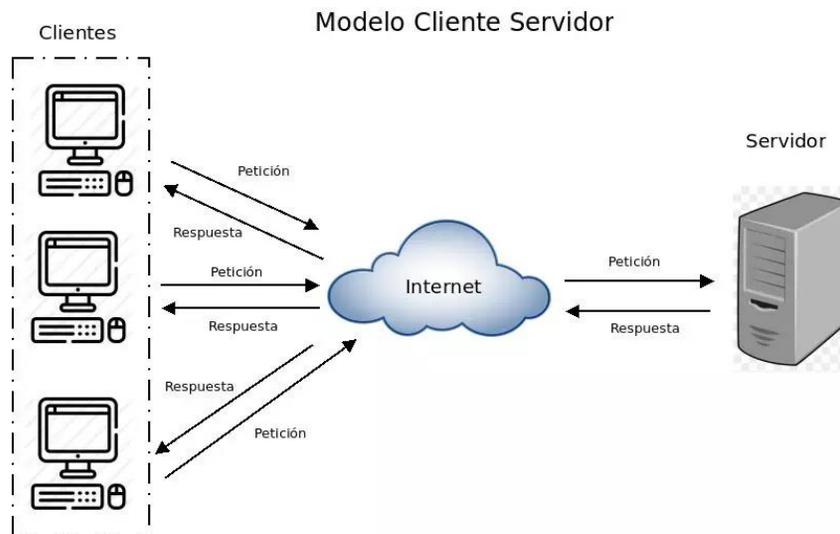
- ✓ Diagrama de casos de uso
- ✓ Diagrama de clases

## 2.7. ARQUITECTURA CLIENTE/SERVIDOR

El modelo Cliente/Servidor como se evidencia en la Figura 4 es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Las aplicaciones Clientes realizan peticiones a una o varias aplicaciones Servidores, que deben encontrarse en ejecución para atender dichas demandas. García Hoyos, (2019, p. 21).

**Figura 12.**

*Modelo cliente-servidor*



*Nota.* Adaptado de *Modelo cliente servidor*, por Andrés Schiaffarino, (2019)

De esta forma, y gracias a esta arquitectura, la totalidad de los procesadores, clientes y servidores, trabajan de forma cooperativa para realizar un determinado tratamiento de la información.

### 2.7.1. Características del modelo cliente servidor

Según Domínguez Betancourt, (2011), nombra las siguientes características del modelo CLIENTE/SERVIDOR:

- Las características básicas de una arquitectura Cliente/Servidor son: Combinación de un cliente que interactúa con el usuario, y un servidor que interactúa con los recursos compartidos. El proceso del cliente proporciona la interfaz entre el usuario y el resto del sistema. El proceso del servidor actúa como un motor de software que maneja recursos compartidos tales como bases de datos, impresoras, módems, etc.
- Las tareas del cliente y del servidor tienen diferentes requerimientos en cuanto a recursos de cómputo como velocidad del procesador, memoria, velocidad y capacidades del disco e input-output devices.
- Se establece una relación entre procesos distintos, los cuales pueden ser ejecutados en la misma máquina o en máquinas diferentes distribuidas a lo largo de la red.
- Existe una clara distinción de funciones basada en el concepto de "servicio", que se establece entre clientes y servidores.
- La relación establecida puede ser de muchos a uno, en la que un servidor puede dar servicio a muchos clientes, regulando su acceso a recursos compartidos.
- Los clientes corresponden a procesos activos en cuanto a que son éstos los que hacen peticiones de servicios a los servidores. Estos últimos tienen un carácter pasivo ya que esperan las peticiones de los clientes.
- No existe otra relación entre clientes y servidores que no sea la que se establece a través del intercambio de mensajes entre ambos. El mensaje es el mecanismo para la petición y entrega de solicitudes de servicio.
- El ambiente es heterogéneo. La plataforma de hardware y el sistema operativo del cliente y del servidor no son siempre la misma. Precisamente una de las principales ventajas de esta arquitectura es la posibilidad de conectar clientes y servidores independientemente de sus plataformas.
- El concepto de escalabilidad tanto horizontal como vertical es aplicable a cualquier sistema Cliente/Servidor. La escalabilidad horizontal permite agregar

más estaciones de trabajo activas sin afectar significativamente el rendimiento. La escalabilidad vertical permite mejorar las características del servidor o agregar múltiples servidores.

### **2.7.2. Componentes del modelo cliente servidor**

Según Schiaffarino, (2019), nombra algunos conceptos básicos que lo conforman.

- **Red:** Una red es un conjunto de clientes, servidores y base de datos unidos de una manera física o no física en el que existen protocolos de transmisión de información establecidos.
- **Cliente:** El concepto de cliente hace referencia a un demandante de servicios, este cliente puede ser un ordenador como también una aplicación de informática, la cual requiere información proveniente de la red para funcionar.
- **Servidor:** Un servidor hace referencia a un proveedor de servicios, este servidor a su vez puede ser un ordenador o una aplicación informática la cual envía información a los demás agentes de la red.
- **Protocolo:** Un protocolo es un conjunto de normas o reglas y pasos establecidos de manera clara y concreta sobre el flujo de información en una red estructurada.
- **Servicios:** Un servicio es un conjunto de información que busca responder las necesidades de un cliente, donde esta información pueden ser mail, música, mensajes simples entre software, videos, etc.
- **Base de datos:** Son bancos de información ordenada, categorizada y clasificada que forman parte de la red, que son sitios de almacenaje para la utilización de los servidores y también directamente de los clientes.

### **2.7.3. Diferencia entre cliente y servidor**

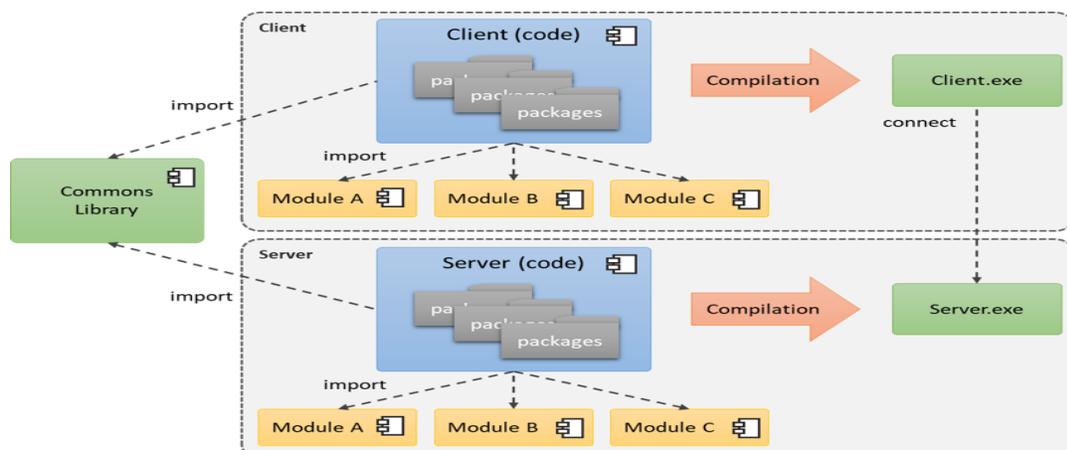
La diferencia principal entre una máquina cliente y una máquina servidor está en su desempeño. Las máquinas cliente son consideradas óptimas para aplicaciones

que requieren tiempos de inicio rápidos. Una máquina servidor es considerada óptima para aplicaciones donde se enfatiza más el desempeño. Betina, (2013)

- La primera diferencia principal entre un cliente y un servidor son sus configuraciones de operación.
- El sistema del servidor se basa en un protocolo operativo relativamente simple, ya que su funcionalidad se limita principalmente a generar solicitudes.
- Solo un único usuario puede realizar solicitudes utilizando un sistema operativo cliente en un momento dado.
- Los niveles de eficiencia de un servidor son mucho más mejorados que los de un sistema cliente.
- El sistema del servidor puede almacenar y analizar grandes conjuntos de datos.
- Los servidores nunca se apagan. Apagar los servidores puede ser catastrófico para los sistemas cliente que solicitan servicios constantemente.

**Figura 13.**

*Estructura de una aplicación Cliente Servidor*



*Nota.* Como podemos ver en la imagen, el cliente y el servidor son construidos en lo general como Monolíticos. (Arquitectura Cliente-Servidor)

#### **2.7.4. Tipos de cliente servidor**

Según Schiaffarino A (2019), dentro de la arquitectura cliente servidor existen tres tipos en donde hablaremos brevemente de cómo funciona cada uno de ellos.

- Arquitectura de dos capas: Esta se utiliza para describir los sistemas cliente servidor en donde el cliente solicita recursos y el servidor responde directamente a la solicitud con sus propios recursos.
- Arquitectura de tres capas: En la arquitectura de tres capas existe un nivel intermediario, eso significa que la arquitectura generalmente está compartida por un cliente que como hablamos más arriba es el que solicita los recursos equipados con una interfaz de usuario o mediante un navegador web.
- La capa del medio es denominada software intermedio cuya tarea es proporcionar los recursos solicitados pero que requiere de otro servidor para hacerlo.
- Arquitectura N capas: En la arquitectura de tres capas, los servidores dos y tres realizaron una tarea específica por lo tanto un servidor web puede usar los servicios de otros servidores para poder proporcionar su propio servicio.
- La arquitectura en tres niveles es potencialmente una arquitectura en N capas ya que, así como está contemplado en tres niveles como el caso anterior puede estar compuesto por N servidores donde cada uno de ellos brindan su servicio específico.

#### **2.8. MÉTRICAS DE CALIDAD**

En la mayoría de los desafíos técnicos, las métricas nos ayudan a entender tanto el proceso técnico que se utiliza para desarrollar un producto, como el propio producto.

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

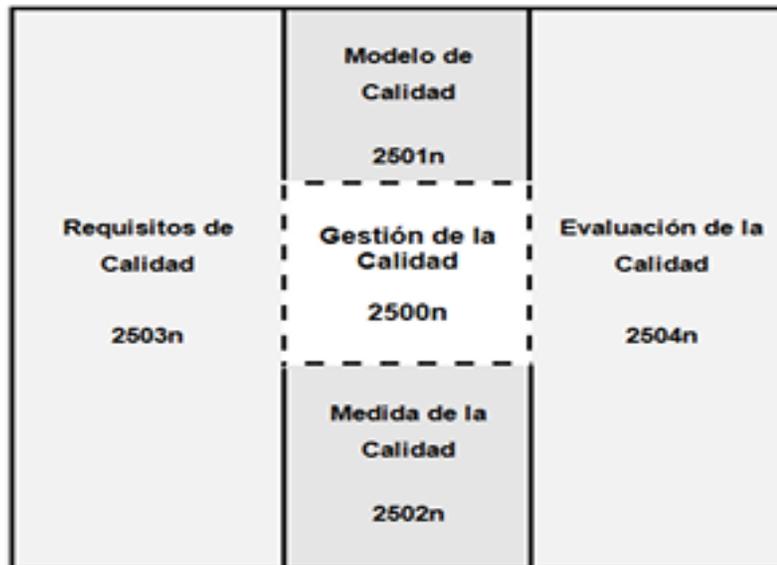
### 2.8.1. Norma ISO/IEC 25000

La norma ISO/IEC 25000 provee una guía para el uso de las nuevas series y estándares internacionales, llamados requisitos y evaluación de calidad de productos de software – SquaRE.

Su principal objetivo es guiar la evaluación de calidad de productos software estableciendo criterios para la especificación de requisitos de calidad de software, sus métricas y su evaluación. Bachelor's thesis y Quito, (2015, 2014).

**Figura 14.**

*División de la norma ISO/IEC 25000*



*Nota. Adaptado de ISO/IEC 25000. Arroyo, (2008)*

### 2.8.2. Funcionalidad

Conjunto de atributos que se relacionan con la existencia de un conjunto de funciones y sus propiedades específicas. Las funciones son aquellas que satisfacen lo indicado o implica necesidades.

Funcionalidad es la capacidad del software de cumplir y proveer las funciones para satisfacer las necesidades explícitas e implícitas cuando es utilizado en condiciones específicas. Largo Garcia y Marin Mazo, (2003)

- **Adecuación:** La capacidad del software para proveer un adecuado conjunto de funciones que cumplan las tareas y objetivos especificados por el usuario.
- **Exactitud:** La capacidad del software para hacer procesos y entregar los resultados solicitados con precisión o de forma esperada.
- **Interoperabilidad:** La capacidad del software de interactuar con uno o más sistemas específicos.
- **Seguridad:** La capacidad del software para proteger la información y los datos de manera que los usuarios o los sistemas no autorizados no puedan acceder a ellos para realizar operaciones, y la capacidad de aceptar el acceso a los datos de los usuarios o sistemas autorizados
- **Conformidad de la funcionalidad:** La capacidad del software de cumplir los estándares referentes a la funcionalidad.

### **2.8.3. Fiabilidad**

Conjunto de atributos relacionados con la capacidad de mantener un nivel de presentación bajo condiciones establecidas durante un periodo de tiempo establecido.

Capacidad del software para asegurar un nivel de funcionamiento adecuado cuando es utilizando en condiciones específicas. En la **ISO/IEC 2382-14:1997** es "la habilidad de la unidad funcional de realizar una función requerida".

- **Madurez:** La capacidad que tiene el software para evitar fallas cuando encuentra errores. Ejemplo, la forma como el software advierte al usuario cuando realiza operaciones en la unidad de diskette vacía, o cuando no encuentra espacio suficiente el disco duro donde esta almacenando los datos.
- **Tolerancia a errores:** La capacidad que tiene el software para mantener un nivel de funcionamiento en caso de errores.

- **Recuperabilidad:** La capacidad que tiene el software para restablecer su funcionamiento adecuado y recuperar los datos afectados en el caso de una falla.
- **Conformidad de la fiabilidad:** La capacidad del software de cumplir a los estándares o normas relacionadas a la fiabilidad.

#### **2.8.4. Usabilidad**

Conjunto de atributos relacionados con el esfuerzo necesitado para el uso, y en la valoración individual de tal uso, por un establecido o implicado conjunto de usuarios.

Es la capacidad del software de ser entendido, aprendido, y usado en forma fácil y atractiva. Está determinada por los usuarios finales y los usuarios indirectos del software, dirigidos a todos los ambientes, a la preparación del uso y el resultado obtenido.

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

### **2.8.5. Eficiencia**

Conjunto de atributos relacionados con la relación entre el nivel de desempeño del SW y la cantidad de recursos necesitados bajo condiciones establecidas.

Es la forma del desempeño adecuado, de acuerdo a al número recursos utilizados según las condiciones planteadas.

- **Comportamiento de tiempos:** Los tiempos adecuados de respuesta y procesamiento, el rendimiento cuando realiza su función en condiciones específicas.
- **Utilización de recursos:** La capacidad del software para utilizar cantidades y tipos adecuados de recursos cuando este funciona bajo requerimientos o condiciones establecidas. Ejemplo, los recursos humanos, el hardware, dispositivos externos.
- **Conformidad de eficiencia:** La capacidad que tiene el software para cumplir con los estándares o convenciones relacionados a la eficiencia.

### **2.8.6. Mantenibilidad**

Conjunto de atributos relacionados con la facilidad de extender, modificar o corregir errores en un sistema SW.

Es la cualidad que tiene el software para ser modificado. Incluyendo correcciones o mejoras del software, a cambios en el entorno, y especificaciones de requerimientos funcionales.

- **Capacidad de ser analizado:** La forma como el software permite diagnósticos de deficiencias o causas de fallas, o la identificación de partes modificadas.
- **Confiabilidad:** La capacidad del software para que la implementación de una modificación se pueda realizar, incluye también codificación, diseño y documentación de cambios.
- **Estabilidad:** La forma como el software evita efectos inesperados para modificaciones del mismo.

- **Facilidad de prueba:** La forma como el software permite realizar pruebas a las modificaciones sin poner el riesgo los datos.
- **Conformidad de facilidad de mantenimiento:** La capacidad que tiene el software para cumplir con los estándares de facilidad de mantenimiento.

### **2.8.7. Portabilidad**

Conjunto de atributos relacionados con la capacidad de un sistema SW para ser transferido desde una plataforma a otra.

La capacidad que tiene el software para ser trasladado de un entorno a otro.

- **Adaptabilidad:** Es como el software se adapta a diferentes entornos especificados (hardware o sistemas operativos) sin que implique reacciones negativas ante el cambio. Incluye la escalabilidad de capacidad interna (Ejemplo: Campos en pantalla, tablas, volúmenes de transacciones, formatos de reporte, etc.).
- **Facilidad de instalación:** La facilidad del software para ser instalado en un entorno específico o por el usuario final.
- **Coexistencia:** La capacidad que tiene el software para coexistir con otro o varios softwares, la forma de compartir recursos comunes con otro software o dispositivo.
- **Reemplazabilidad:** La capacidad que tiene el software para ser reemplazado por otro software del mismo tipo, y para el mismo objetivo. es importante para el usuario, la propiedad de poder migrar los datos a otro software de diferente proveedor.
- **Conformidad de portabilidad:** La capacidad que tiene el software para cumplir con los estándares relacionados a la portabilidad.

## **2.9. SEGURIDAD DE SOFTWARE ISO – 27001**

La ISO/IEC 27001 es una norma hecha con el propósito de proporcionar los requisitos para el establecimiento, implementación mantenimiento y mejora continua

de un sistema de gestión de seguridad de la información. La adopción de un SGSI es una decisión estratégica para toda organización. información, ISO/IEC 27001 Seguridad de la, (s.f.)

La norma ISO 27001:2013 trabaja sobre un enfoque a procesos, una aplicación detallada y puntual de la misma, requiere de un manual de procesos, completa y debidamente establecidos, definidos, documentados y validados. El correcto diseño, establecimiento y operación de un SGSI permite a la organización realizar más eficientemente sus actividades, sin embargo, si no se ha aplicado una reingeniería de procesos a la organización previa al establecimiento del SGSI, esta ralentiza sus operaciones. Esta norma tiene un total de 14 Dominios, 35 Objetivos de Control y 114 Controles. Mantilla Guerra, (2017)

Esta norma internacional especifica los requisitos para establecer, implementar, mantener y mejorar continuamente un sistema de gestión de seguridad de la información en el contexto de la organización. Esta norma también incluye requisitos para la evaluación y el tratamiento de los riesgos de seguridad de la información a la medida de las necesidades de la organización. Los requisitos establecidos en esta Norma Internacional son genéricos y se pretende que sean aplicables a todas las organizaciones, sin importar su tipo, tamaño o naturaleza. Valero Duran, (2013)

Los objetivos del SGSI son preservar la:

- Confidencialidad.
- Integridad.
- Disponibilidad.

### **2.9.1. Estructura de la ISO 27001.**

Según ISOTools EXCELENCE, (2019), cuenta con la siguiente estructura.

- **Objeto y campo de aplicación:** La norma comienza aportando unas orientaciones sobre el uso, finalidad y modo de aplicación de este estándar.
- **Referencias Normativas:** Recomienda la consulta de ciertos documentos indispensables para la aplicación de ISO27001.

- **Términos y Definiciones:** Describe la terminología aplicable a este estándar.
- **Contexto de la Organización:** Este es el primer requisito de la norma, el cual recoge indicaciones sobre el conocimiento de la organización y su contexto, la comprensión de las necesidades y expectativas de las partes interesadas y la determinación del alcance del SGSI.
- **Liderazgo:** Este apartado destaca la necesidad de que todos los empleados de la organización han de contribuir al establecimiento de la norma. Para ello la alta dirección ha de demostrar su liderazgo y compromiso, ha de elaborar una política de seguridad que conozca toda la organización y ha de asignar roles, responsabilidades y autoridades dentro de la misma.
- **Planificación:** Esta es una sección que pone de manifiesto la importancia de la determinación de riesgos y oportunidades a la hora de planificar un Sistema de Gestión de Seguridad de la Información, así como de establecer objetivos de Seguridad de la Información y el modo de lograrlos.
- **Soporte:** En esta cláusula la norma señala que para el buen funcionamiento del SGSI la organización debe contar con los recursos, competencias, conciencia, comunicación e información documentada pertinente en cada caso.
- **Operación:** Para cumplir con los requisitos de Seguridad de la Información, esta parte de la norma indica que se debe planificar, implementar y controlar los procesos de la organización, hacer una valoración de los riesgos de la Seguridad de la Información y un tratamiento de ellos.
- **Evaluación del Desempeño:** En este punto se establece la necesidad y forma de llevar a cabo el seguimiento, la medición, el análisis, la evaluación, la auditoría interna y la revisión por la dirección del Sistema de Gestión de Seguridad de la Información, para asegurar que funciona según lo planificado.
- **Mejora:** Por último, en la sección décima vamos a encontrar las obligaciones que tendrá una organización cuando encuentre una no conformidad y la importancia de mejorar continuamente la conveniencia, adecuación y eficacia del SGSI.

## **2.10. PRUEBAS DE ACEPTACIÓN**

Para llevar a cabo las pruebas verificaremos el comportamiento del programa sobre un conjunto de casos de prueba. Estos casos de prueba se generarán mediante técnicas y estrategias específicas de pruebas que nos ayudarán a conseguir la búsqueda de los errores de un programa. Para encontrar los errores, dos de las técnicas más utilizadas en las pruebas son las técnicas de caja blanca y caja negra. Sin embargo, a la hora de probar tenemos que establecer prioridades. Una de las prioridades más importantes que hay que tener en cuenta son los recursos de los que se va a disponer en el proyecto. Al realizar un análisis de los riesgos para asegurar que se dispone de recursos suficientes para poder llevar a cabo las pruebas. Estos recursos irán desde el personal, hasta las herramientas que se vayan a utilizar (Sánchez Peño, 2015, pág. 23).

Desde el inicio hasta la implementación, el software deberá someterse a varios tipos de pruebas. El objetivo siempre será asegurar la calidad para evitar reprocesos y garantizar las funcionalidades de la aplicación, tanto para el usuario final, como para el cliente.

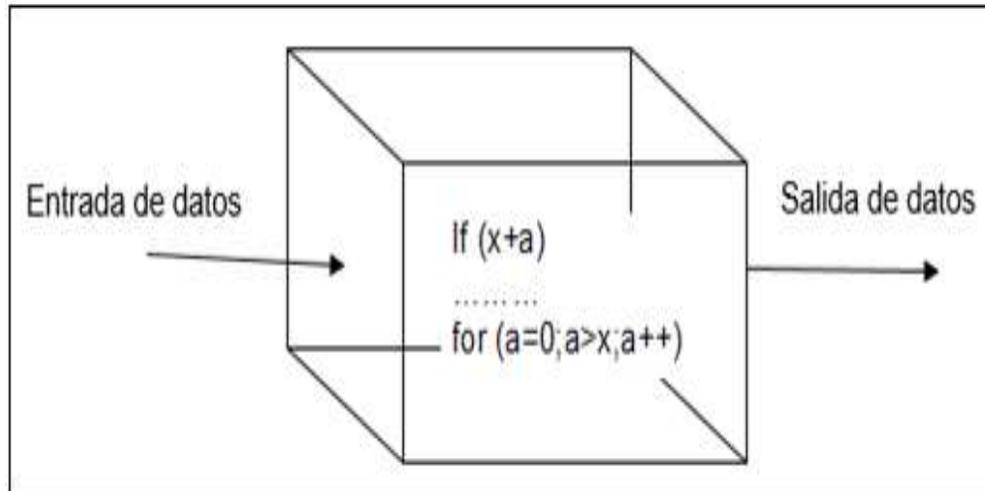
### **2.10.1. Prueba de caja blanca**

En las pruebas de Caja Blanca se desarrollan casos de prueba que produzcan la ejecución de cada posible ruta del programa o módulo, considerándose una ruta como una combinación específica de condiciones manejadas por un programa.

Hay que señalar que no todos los errores de software se pueden descubrir verificando todas las rutas de un programa, hay errores que se descubren al integrar unidades del sistema y pueden existir errores que no tengan relación con el código específicamente. Caisa Guayta y Semblantes, (2011).

**Figura 15.**

*Caja blanca*



*Nota. Pruebas de Software, Fundamentos y Técnicas (2015).*

Al usar estos 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.

### **2.10.2. Prueba de Ruta básica**

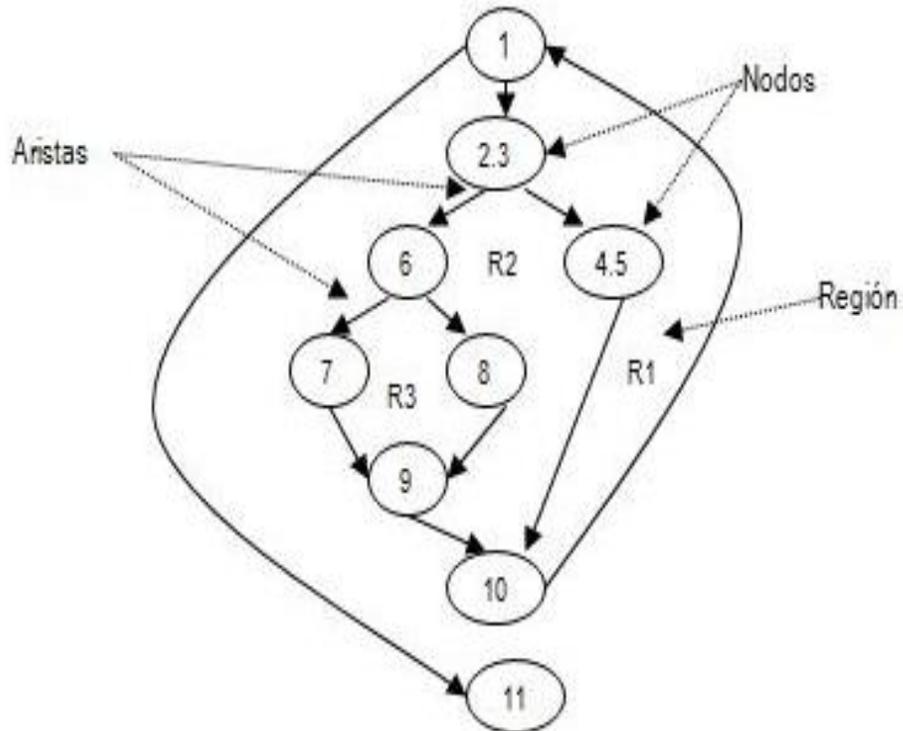
La prueba de ruta o trayectoria básica es una técnica que fue propuesta por (Tom McCabe) y consiste en definir un conjunto de caminos usando la medida de complejidad ciclomática (VG), es decir permite que el diseñador de casos de prueba, al obtener una medida de complejidad lógica de un diseño procedimental y que se use dicha medida como guía para definir un conjunto básico de rutas de ejecución.

Antes de tratar el método de la RB, debe presentarse una notación simple para la representación de flujo de control, llamada gráfica de flujo.

La gráfica de flujo describe un flujo de control lógico empleado, cada construcción estructurada tiene su símbolo correspondiente en la notación.

**Figura 16.**

*Prueba del camino básico*



*Nota. Adaptado de Complejidad Ciclos maticas, Rojas, (2011)*

La complejidad ciclo matica determina el número de caminos a probar, mediante la siguiente formula.

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

Los pasos en las pruebas de caminos básicos son:

- 1) Dibujar grafo de flujo.
- 2) Determinar la complejidad ciclo matica del grafo.
- 3) Determinar los caminos linealmente independientes.
- 4) Diseñar los casos de prueba.

### 2.10.3. Prueba de caja negra

Las pruebas de caja negra se centran en lo que se espera de un módulo, es decir, intentan encontrar casos en que el módulo no se atiene a su especificación. Por ello se denominan pruebas funcionales, y el probador se limita a suministrarle datos como entrada y estudiar la salida, sin preocuparse de lo que pueda estar haciendo el módulo por dentro. Caisa Guayta y Semblantes Chicaiza, (2008 y 2009)

Consiste en estudiar la especificación de las funciones, la entrada y la salida para derivar los casos. Aquí, la prueba ideal del software consiste en probar todas las posibles entradas y salidas del programa.

**Figura 17.**

*Prueba de Caja Negra*



*Nota. Adaptado de Complejidad Ciclo matica prueba caja negra, Rojas, (2011)*

En esta trayectoria de la prueba de caja negra, también encuentra errores.

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

#### **2.10.4. Prueba de estrés**

Las pruebas de estrés (stress testing): Evalúa el comportamiento de los sistemas cuando se llevan más allá de sus límites operacionales (que pueden ser muy superiores requisitos no funcionales) estas pruebas de estrés se deben diseñar para llevar los límites de los recursos del sistema hasta que exponer los puntos débiles de la aplicación. Zapata y Velásquez, (2011, págs. 143-154)

Mientras las pruebas de estrés de hardware tienen como objetivo determinar la estabilidad y vida útil operativa bajo una carga extrema, las pruebas de estrés de software tienen como objetivo determinar la robustez, disponibilidad y capacidad de gestionar errores bajo condiciones de carga intensas.

Para ello las pruebas de resistencia se centran en diversos aspectos como;

- Asegurar que el sistema funciona en condiciones fuera de lo habitual.
- Analizar el comportamiento del sistema tras un fallo y garantizar su recuperación.
- Verificar que los fallos no provoquen problemas de seguridad.
- Garantizar la visualización de los mensajes de error en caso de emergencia.

Sabiendo estos conceptos con las pruebas de estrés estas nos ayudan a determinar el comportamiento del sistema en caso de carga extrema, como puede ser un ataque DDos, por lo tanto, es recomendable realizar estas pruebas de modo que permite detectar problemas que pueden pasar desapercibidos durante cortos periodos de tiempo.

#### **2.11. ESTIMACIÓN DE COSTO**

El producto de software ha tenido una muy alta frecuencia de exceso de calendarización, costos, problemas de calidad e indiscutibles cancelaciones.

Las estimaciones de tiempo y costo deberían ser exactas, naturalmente. Pero si ellas difieren de los resultados reales, es más seguro ser ligeramente conservador

que ser optimista. Una de las principales quejas sobre los proyectos de software se refiere a su tendencia alarmante de exceder gastos y calendarios planificados. Desafortunadamente, tanto clientes como ejecutivos superiores tienden a ejercer presiones considerables en los administradores de proyectos y en el personal encargado de realizar las estimaciones. La mejor defensa es una buena colección de datos históricos de proyectos similares. Araujo, (2014, pág. 19)

Los rasgos principales de instrumentos de estimación de software comerciales incluyen estos atributos:

- Estimación de fiabilidad y calidad.
- Análisis de valor y riesgo.
- Retorno de Inversión.
- Posibilidad de compartir datos con herramientas de administración de proyectos.
- Medios de medición para reunir datos históricos.
- Costo y tiempo para completar estimaciones que combinan datos históricos con datos proyectados.
- Apoyo para evaluaciones de procesos de software.
- Análisis estadístico de múltiples proyectos y análisis de cartera.
- Conversión monetaria para acordar proyectos en el exterior.

**Proceso de estimación:** El proceso de estimación del coste de un producto software está formado por un conjunto de técnicas y procedimientos que se usan en la organización para poder llegar a una predicción fiable. Este es un proceso continuo, que debe ser usado y consultado a lo largo de todo el ciclo de vida del proyecto. Se divide en los siguientes pasos. Agarwal, (2001, pág. 20)

- Estimación del tamaño.
- Estimación del costo y del esfuerzo.
- Estimación de la programación temporal.

- Estimación de la cantidad de recursos computacionales.
- Asunción de riesgos.
- Inspección y aprobación.
- Redacción de informes de estimación.
- Medición y perfeccionamiento del proceso.

### **2.11.1. Características de COCOMO II**

La estructura de COCOMO II se basa en modelos que asumen que se progresa a lo largo de un desarrollo de tipo espiral para consolidar los requisitos y la arquitectura, reduciendo el riesgo.

En la siguiente visualizamos la estructura de COCOMO II, se basa en los modelos que asumen que se progresa a lo largo de un desarrollo para consolidar los requisitos y la arquitectura, reduciendo el riesgo, tales como:

- Modelo de composición de aplicaciones.
- Modelo de Diseño temprano.
- Modelo de arquitectura tardía.

**Figura 18.**

*Modelo arquitectura COCOMO II*



*Nota. Adaptado de la Arquitectura COCOMO II. Broehm , (2000)*

### **2.11.2. Modelos de COCOMO II**

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

- Composición de Aplicación: Se emplea en desarrollos de software durante la etapa de prototipado de las interfaces de cada usuario, la interacción del sistema, evaluación del rendimiento y la evaluación de la madurez son etapas o factores más importantes del ciclo de vida.
- Diseño Temprano: Se utiliza en las primeras etapas del desarrollo en las cuales se evalúan las alternativas de hardware y software de un proyecto.
- Post-Arquitectura: Se aplica en la etapa de desarrollo, después de definir la arquitectura del sistema, y en la etapa de mantenimiento.

### 2.11.3. Estimación de esfuerzo

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

La fórmula en este modelo es la siguiente:

$$PM = \frac{NOP}{PROD}$$

Donde:

**NOP:** (Nuevos Puntos Objeto): Tamaño del nuevo software a desarrollarse expresado en:

Puntos Objeto y se calcula de la siguiente manera:

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

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

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

**PROD:** Es la productividad promedio determinada a partir del análisis de proyectos en:

**Tabla 2.**

*Estimación de Esfuerzo*

Experiencia y capacidad de los desarrolladores	Muy Bajo	Bajo	Normal	Alto	Muy Alto
Madurez y Capacidad del ICASE	Muy Bajo	Bajo	Normal	Alto	Muy Alto
PROD	4	7	13	25	50

*Nota.* Adaptado de (*Un Modelo de Estimación de proyectos de Software, 2010*)

#### **2.11.4. Estimación del cronograma.**

La versión inicial de COCOMO II provee un modelo de estimación del cronograma similar al presentado en COCOMO' 81 y ADA COCOMO. La ecuación inicial para los tres modelos de COCOMO II es:

$$TDEV = [3.0 \times PM_*^{(0.33+0.2 \times (B-1.01))}] \times \frac{SCED\%}{100}$$

Donde:

**TDEV:** es el tiempo calendario en meses que transcurre desde la determinación de los requerimientos a la culminación de una actividad que certifique que el producto cumple con las especificaciones.

**PM:** es el esfuerzo expresado en meses personas, calculado sin tener en cuenta el multiplicador de esfuerzo SCED.

**SCED %:** es el porcentaje de compresión/expansión del cronograma.

#### ***Las líneas de código fuente.***

Considera de código fuente sentencia fuente lógica como línea standard de código, se usa el checklist de definición desarrollado por el SEI, que permite unificar criterios en la definición de una línea de código fuente. Existen herramientas automatizadas para medir la cantidad de líneas de código fuente, como por ejemplo Amadeus.

### **2.12. HERRAMIENTAS**

#### **2.12.1. Lenguaje de programación**

##### **2.12.1.1. PHP**

PHP es un acrónimo recursivo para “PHP: Hypertext Preprocessor”, originalmente Personal Home Page, es un lenguaje interpretado libre, usado originalmente solamente para el desarrollo de aplicaciones presentes y que actuaran en el lado del servidor. Arias, (2013)

PHP es un lenguaje interpretado del lado del servidor que surge dentro de la corriente denominada código abierto (open source).

Se caracteriza por su potencia, versatilidad, robustez y modularidad. Al igual que ocurre con tecnologías similares, los programas son integrados directamente dentro del código HTML. El lenguaje PHP es uno de los lenguajes más utilizados actualmente en el desarrollo de aplicaciones web y viene experimentando un constante crecimiento en su nivel de utilización en Interno Zambrano, Vélez Vélez , y Almeida Zambrano, ( 2020)

#### **2.12.1.2. JavaScript**

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas.

JavaScript, comúnmente abreviado como JS, es un lenguaje de programación del tipo interpretado. Está basado en el estándar ECMAScript y, a pesar de ser débilmente tipado y dinámico, también se define como un lenguaje orientado a objetos. Cabello, (2010)

#### **2.12.2. Base de Datos**

Una base de datos es una recopilación organizada de información o datos estructurados que normalmente se almacena de forma electrónica en un sistema.

##### **2.12.2.1. PostgreSQL**

PostgreSQL es un gestor de bases de datos relacional es organizado en diferentes niveles. El nivel más externo es el clúster. En un servidor se pueden instalar varios clústeres del gestor, siempre y cuando cada uno de ellos utilice un puerto diferente. Clavadetscher, (2015)

El Sistema Gestor de Bases de Datos Relacionales Orientadas a Objetos conocido como PostgreSQL (y brevemente llamado Postgres95) está derivado del paquete PostgreSQL escrito en Berkeley. Con cerca de una década de desarrollo tras él, PostgreSQL es el gestor de bases de datos de código abierto más avanzado hoy

en día, ofreciendo control de concurrencia multi-versión, soportando casi toda la sintaxis SQL (incluyendo subconsultas, transacciones, y tipos y funciones definidas por el usuario), contando también con un amplio conjunto de enlaces con lenguajes de programación (incluyendo C, C++, Java, Perl, tcl y Python). Lockhart, (1988)

### **2.12.3. Framework**

El framework es un esquema o marco de trabajo que ofrece una estructura base para elaborar un proyecto con objetivos específicos, una especie de plantilla que sirve como punto de partida para la organización y desarrollo de software.

Son similares a la vista estructural, pero su énfasis primario radica en la (usualmente una sola) estructura coherente del sistema completo, en vez de concentrarse en su composición. Los modelos de framework a menudo se refieren a dominios o clases de problemas específicos. Rojas y García, (2004)

#### **2.12.3.1. Backend**

Se encarga de escribir el código que crea o utiliza los servicios que dan respuesta a los pedidos y de la comunicación con el frontend.

Su principal desafío es garantizar la seguridad del sitio a atender y resolver a varios pedidos a la vez, optimizando así los tiempos de respuesta (Performance).

Teniendo en cuenta que la información recopilada en el Back-end debe ser suministrada como servicio al Front-end para su debida manipulación y presentación visual al usuario, se configuran unas rutas dentro del proyecto para poder controlar cada uno de los elementos que serán presentados al Front-end; es preciso mencionar que el proyecto que abarca la sección del desarrollo Front-end para la solución software presentada, ha sido realizado. Arias Estupiñán, (2022)

#### **2.12.3.2. Laravel**

Laravel es un framework para el lenguaje de programación PHP. Aunque PHP es conocido por tener una sintaxis poco deseable, es fácil de usar, fácil de desplegar y se le puede encontrar en muchos de los sitios web modernos que usas día a día.

Laravel no solo ofrece atajos útiles, herramientas y componentes para ayudarte a conseguir el éxito en tus proyectos basados en web, sino que también intenta arreglar alguna de las flaquezas de PHP. (Richos Organization, 2018)

### **2.12.3.3. Frontend**

Esta se encarga de escribir el código para hacer visible el sitio, que permite la interacción con los usuarios y de la comunicación con el Backend.

Sus principales desafíos tienden a adaptarse a los distintos navegadores, lo cual se despliegan rápidamente al sitio (Performance) Lograr que los clientes realicen pedidos (usabilidad). (Roberto Caroca, 2020)

### **2.12.4. HTML**

Una página web está compuesta de un lenguaje de marcas. Este lenguaje de marcas se llama HTML. HTML (Lenguaje de Modelado de Hiper Texto), son etiquetas para construir páginas web y está a cargo de la estructura o el esqueleto de la página. Las etiquetas HTML puedes escribir lo que desees en la página textos, enlaces imágenes. (Boté, J y Juanjo Boté, 2013)

### **2.12.5. Css**

CSS (Cascading Style Sheets), es un lenguaje de hojas de estilo, es decir, te permite aplicar estilos de manera selectiva a elementos en HTML.

Las hojas de estilo en cascada (CSS, Cascading Style Sheets) es un lenguaje de estilos que define el aspecto, la presentación y la posición que tendrán los diferentes elementos que componen una página web. El protagonismo que ha ido adquiriendo el uso de las hojas de estilo en el mundo del diseño web ha ido creciendo año tras año hasta convertirse en un mecanismo totalmente imprescindible. (Gómez, Mario Rubiales, 2022)

# **CAPÍTULO III**

## **MARCO APLICATIVO**

### 3. MARCO APLICATIVO.

#### 3.1. INTRODUCCIÓN.

El presente capítulo constituye el proceso central para la solución al problema planteado en los capítulos anteriores, identificando los diferentes procesos de control, administración cobro del suministro, con el fin de identificar los requerimientos que se precisan en el sistema, también se identificarán los principales actores, de esta manera se logrará preparar un adecuado desarrollo del sistema.

#### 3.2. DESARROLLO MODELADO MEDIANTE UWE

##### 3.2.1. *Captura, análisis y especificación de requisitos*

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

**Tabla 3.**

*Captura, análisis y especificación de requisitos*

HERRAMIENTA	CARACTERISTICAS
Observación	En la unidad de Recaudaciones e Impuestos, perteneciente a la Dirección Administrativa Financiera del GAMCH se pudo observar las dificultades que, a la hora de procesar los registros de afiliación, servicios del mismo suministro y del control de cobros del consumo, generan inconformidad e inconsistencia y descoordinación de los usuarios en el control de la información con las organizaciones Territoriales de Base OTBS.

Documentación	Se revisaron los documentos que se manejan dentro del proceso en la unidad, los requerimientos, normas que se debe seguir y los formatos para la realización de los reportes.
Entrevista	Se realizó entrevistas con el personal involucrado dentro del proceso, quienes son: <ul style="list-style-type: none"> <li>• Responsable de la Unidad Recaudaciones e impuestos.</li> <li>• Encargada de la caja – recaudación.</li> <li>• Dirección Administrativa Financiera.</li> <li>• Responsable de la unidad de intendencia.</li> </ul>

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

### **3.2.2. Requerimientos Funcionales**

Los requerimientos funcionales (RF), se enfocan principalmente en lo que el sistema necesita para los diferentes procesos.

**Tabla 4.**

*Requerimientos funcionales*

ROL	DESCRIPCIÓN	FUNCIÓN
RF_1	Base de datos	Almacenamiento de datos, gestor de base de datos
RF_2	Acceder al sistema	Accede al sistema de los tres tipos de usuarios (Administrador, Técnico y usuarios).
RF_3	Gestionar Usuarios	Registrar a los usuarios con acceso al sistema, con datos necesarios de la persona asignando el cargo que tiene.
RF_4	Gestionar Roles	Registrar los permisos del sistema, y asignar un rol a los distintos cargos del sistema

---

RF_5	Gestionar Permisos	Permite la asignación de permisos a los usuarios dependiendo del rol.
RF_6	Visualización de la página Web	Visualización y funcionamiento correcto en los diferentes navegadores.
RF_7	Interfaz amigable	La interfaz gráfica debe ser fácil y entendible para el usuario.
RF_8	Administración del software	El usuario con perfil de administrador, tiene el acceso total a todos los módulos y operaciones de la administración de la información y control de caja recaudaciones.
RF_9	Perfil del usuario	Muestra la información del usuario y permite a la actualización de foto de perfil y la modificación de la contraseña.
RF_10	Registro y control de los procesos del sistema	Registro de todos los procesos que realicen los usuarios con acceso al sistema.
RF_11	Registro de nuevas afiliaciones y administración de la información.	El usuario con perfil de técnico, tiene el acceso de crear editar, asignar y actualizar el tipo de categorización del servicio, incorporar nuevas direcciones (calles, avenidas) en la población para la instalación del suministro de agua potable en el abastecimiento de la población.
RF_12	Registro de control de cobros, instalación, servicio y emisión de facturas del suministro.	El usuario con perfil de Cajero (a), tiene el acceso de apertura cobros, instalación, deudas pendientes y la emisión de factura/recibo de los respectivos procesos de control en el consumo del suministro.

---

RF_13	Generar Reporte Afiliados	Permite mostrar información de las ultimas afiliaciones en el transcurso del año.
RF_14	Generar Reporte Cobros del servicio del agua	Permite listar las entradas el monto total de cuanto se está recaudando en los últimos años, meses y semanas.
RF_15	Generar Reporte Deudas pendientes.	Permite visualizar la información detallada por cada gestión (anual).

*Nota.* Se realizo la identificación de los requerimientos funcionales.

### **3.2.3. Requerimientos No Funcionales**

Los requerimientos no funcionales (RFN), como su nombre lo indica son cualidades y características del sistema que el usuario puede visualizar, es decir que el sistema proporciona que no se relacionan de forma directa con el comportamiento funcional.

**Tabla 5.**

*Requerimientos no funcionales*

ROL	FUNCIÓN
RNF_1	El sistema debe funcionar correctamente en cualquier ordenador que tenga disponible la conexión a internet desde los navegadores modernos como ser: Google Chrome, Brave, Mozilla Firefox, Microsoft Edge y otros.
RNF_2	El sistema debe tener un entorno amigable al usuario, para tener un entendimiento de las funcionalidades y operaciones que se pueden realizar en el sistema.
RNF_3	El acceso debe ser restringido mediante autenticación de usuario y contraseña.

---

RNF_4	El sistema debe tener un menú lateral para que el usuario pueda navegar a través de ella por diferentes páginas.
RNF_5	El sistema debe tener pestañas de navegación para tener un mejor seguimiento de las páginas web.
RNF_6	Los datos modificados se actualizan para todos los usuarios que acceden al sistema.
RNF_7	El soporte y mantenimiento periódico del sistema asegura un buen rendimiento al sistema.
RNF_8	El sistema debe responder a todas las solicitudes del usuario.
RNF_9	El sistema tendrá que encriptar las contraseñas de los usuarios.
RNF_10	La sesión tendrá un determinado periodo de tiempo activo para que después el usuario vuelva a iniciar sesión.
RNF_11	El URL del sistema debe ser encriptado para que No pueda ser visible para posibles accesos no autorizados.
RNF_12	El sistema debe permitir agregar nuevos módulos y funciones a predicción futuro actualización a grandes escalas de seguridad confiabilidad.

---

*Nota.* Se realizo la identificación de los requerimientos No funcionales.

### **3.2.4. Diseño del sistema.**

Para el diseño de actores del sistema se utilizará las siguientes fases: Descripción de actores, diagramas de caso de uso, diagramas de clase, modelo conceptual y modelo navegacional.

#### **3.2.4.1. Descripción de actores.**

En la identificación de actores nos permitió conocer a las personas involucradas en el proceso de entrega, revisión y reporte de la información en cuanto a las actividades programadas dentro de la institución.

**Tabla 6.**

*Descripción de actores*

ROLES	DESCRIPCIÓN
<p><b>Administrador</b></p>	<p>Es el encargado de administrar el sistema, realiza la gestión de usuarios: El rol como administrador cumple la siguiente función de asignar los roles y permisos.</p> <ul style="list-style-type: none"> <li>• Nombre de usuario</li> <li>• Contraseña</li> <li>• Nivel de acceso</li> <li>• Para tener un control de acceso seguro y diferenciado.</li> </ul> <p>Se encarga de las configuraciones del sistema web.</p>
<p><b>Responsable</b></p>	<p>Su funcionalidad es de asignar a cada miembro del equipo tareas específicas y supervisa su funcionamiento.</p>

---

Encargado de generar reportes de la unidad de Recaudaciones.

Entre las funciones que cumplirá es registrar nuevas afiliaciones.

**Técnico**

Se encarga de asignar nuevas tomas de los procesos en las instalaciones del suministro.

Se encarga en la actualización de datos de las transferencias del servicio.

Se encarga de habilitar nuevas aperturas de cobros.

De acuerdo a las normas, se encarga de llenar los registros de control de cobro de los contribuyentes.

**Cajera**

Valida los datos de pago por concepto del trámite.

Se encargará de emitir facturas y reportes de las funciones que le corresponde como ser las transacciones de cobro por servicio.

---

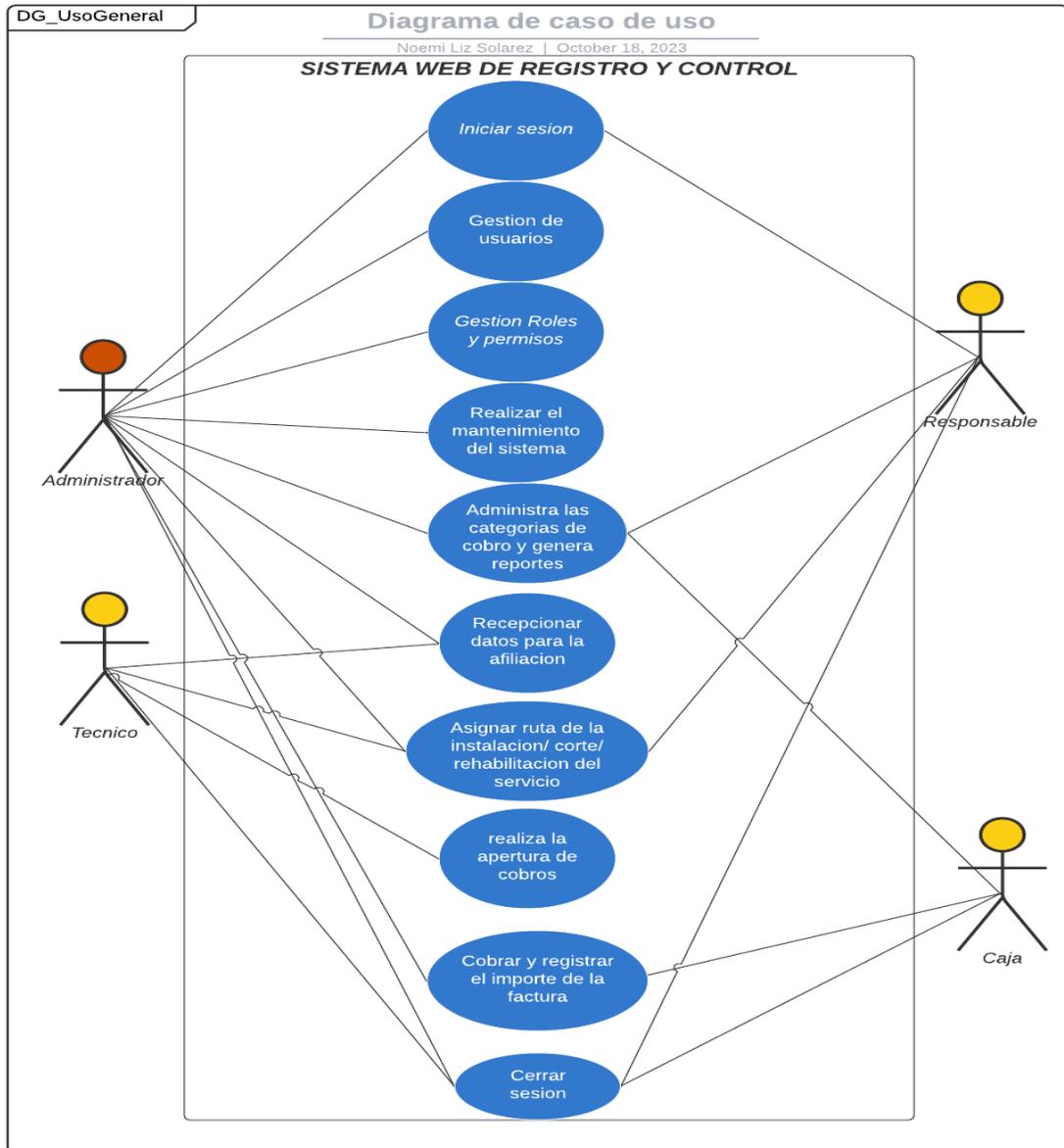
*Nota.* Se realizó la descripción de los actores y cada uno de los roles.

**3.2.4.2. Diagrama de caso de uso general**

A continuación, se hace el modelado donde se puede apreciar como interactúan los actores sobre los casos de uso del sistema.

**Figura 19.**

*Diagrama de caso de uso general*



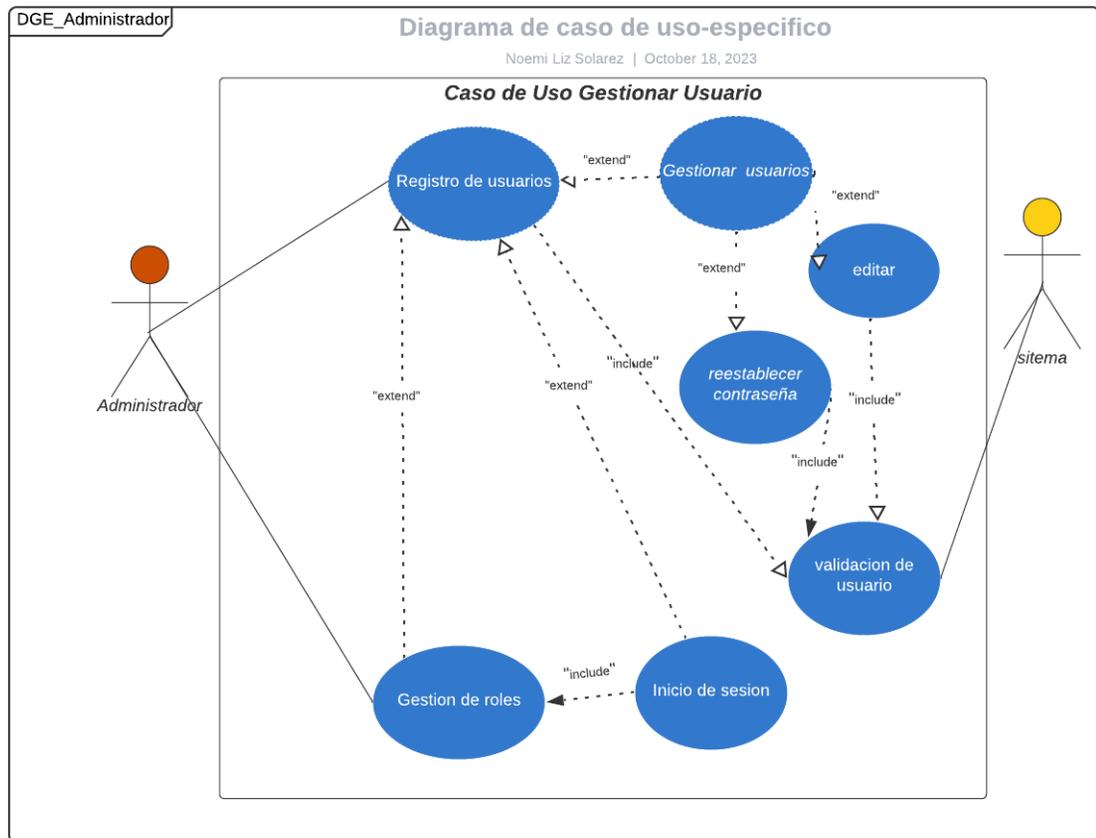
*Nota.* Se realizo la descripción del diagrama de caso de uso general.

### **3.2.4.3. Especificaciones de casos de usos**

A continuación, se describirá en la tabla las especificaciones parte de la administración del sistema:

**Figura 20.**

*Diagrama de caso de uso - Ingreso al sistema*



*Nota.* El caso de uso "Ingreso al Sistema de una Imagen" representa el flujo básico de cómo un usuario ingresa al sistema

**Tabla 7.**

*Descripción de caso de uso - Ingreso al sistema*

Caso de Uso	Ingreso al sistema
Identificador:	<b>DCU01</b>
Descripción:	El sistema permite gestionar los usuarios registrados en el mismo, Validar a un usuario ya registrado para el uso del sistema.

---

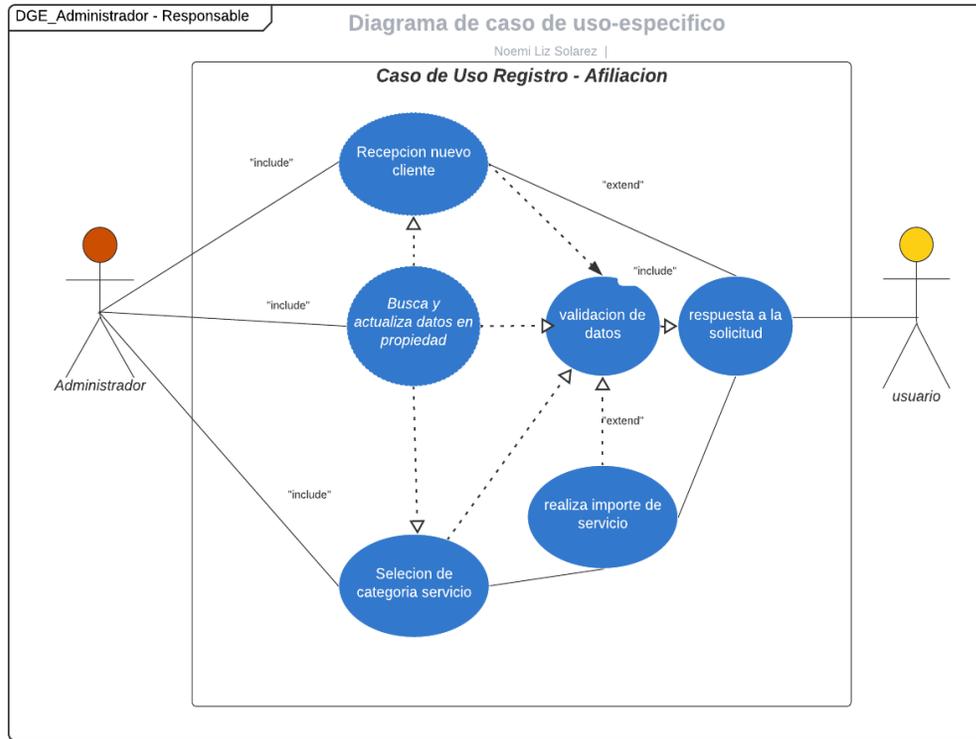
Actores:	Administrador
Precondiciones:	<p>El usuario tiene que estar registrado en el sistema.</p> <p>El caso de uso comienza cuando el usuario selecciona la opción de Administración de usuarios</p>
Flujo normal de Eventos:	<ol style="list-style-type: none"> <li>1 El sistema despliega una lista de usuarios registrados</li> <li>2 El usuario solicita crear, modificar, reset password, activar o desactivar un usuario.</li> <li>3 El sistema solicita la información para crear, modificar, reset de password, activar o desactivar un usuario</li> <li>4 El sistema valida los datos y realiza la acción solicitada</li> </ol>
Post Condiciones:	<p>El usuario logra crear modificar, reset password, activar o desactivar un usuario.</p> <p>Validación realizada con éxito.</p>
Flujos alternativos:	<p style="text-align: center;"><b>Usuario no registrado</b></p> <p>Si el usuario trata de crear o modificar un usuario con datos ya existentes muestra un mensaje de error “El usuario ya existe”</p>

---

*Nota.* Se describe el proceso que sigue un usuario para ingresar al sistema.

**Figura 21.**

Caso de uso – Gestión de administración información afiliación de usuarios



Nota. Este caso de uso describe las acciones llevadas a cabo para afiliación de los usuarios.

**Tabla 8.**

Descripción de caso de uso de afiliados

Caso de Uso	Afiliar Usuarios
Número del caso de uso:	<b>DCU02</b>
Descripción:	Tienen acceso a los registros de los clientes, puede realizar las operaciones de: agregar, modificar y actualizar datos de la información de los afiliados.
Actores:	Personal Administrador y Responsable

---

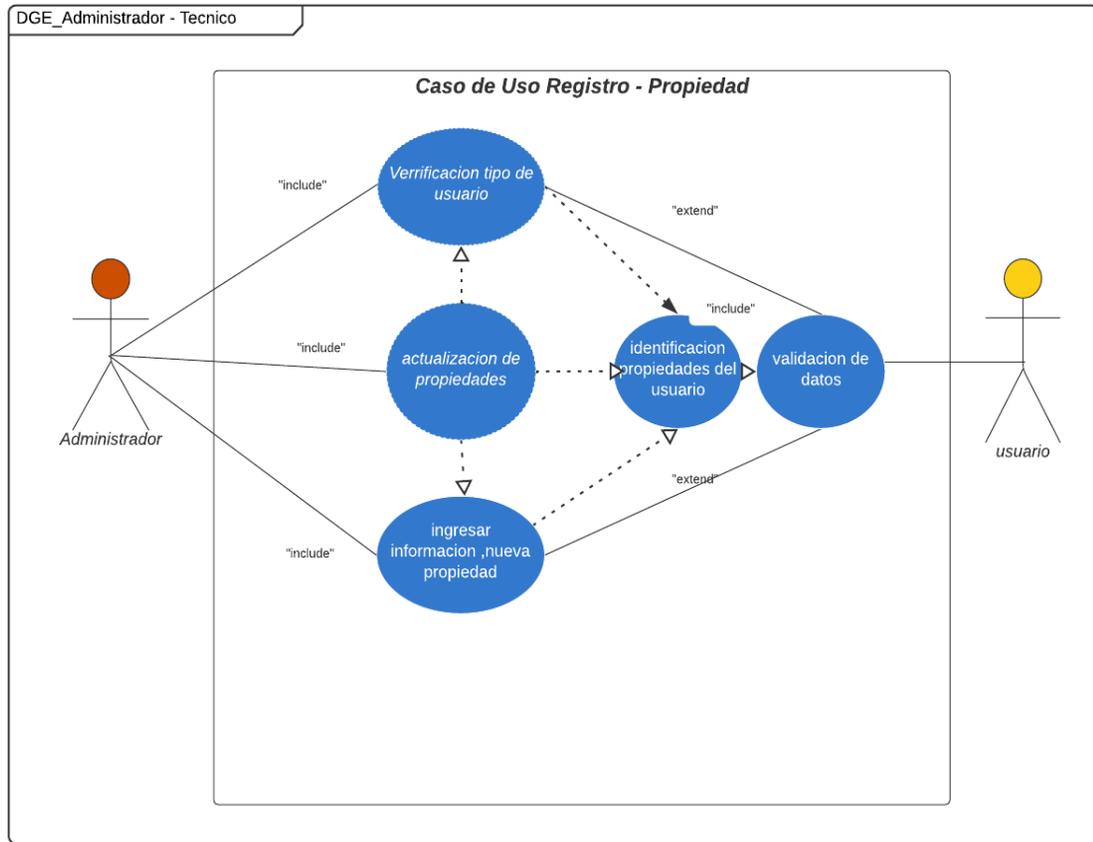
Precondiciones:	El usuario debe estar registrado como administrador o técnico, con los roles y permisos ingresados con anterioridad y haber iniciado la sesión con éxito.
Flujo normal de Eventos:	<p style="text-align: center;"><b>Agregar</b></p> <ol style="list-style-type: none"> <li>1 El usuario escoge la opción afiliar al tipo de persona, seleccionando persona natural o jurídica.</li> <li>2 El sistema muestra las opciones: agregar, modificar, eliminar.</li> <li>3 El sistema muestra los campos para ingresar la información de la información.</li> <li>4 El usuario ingresa la información.</li> <li>5 El sistema verifica el ingreso correcto de la información en todos los campos requeridos.</li> </ol>
Post Condiciones:	El sistema registra toda solicitud para el almacenamiento de datos de los nuevos afiliados.
Flujos alternativos:	<p style="text-align: center;"><b>Modificar</b></p> <p>En caso de cometer algún error en la adición, eliminación o modificación el sistema informara el sistema informara si los campos están vacíos el sistema mostrará mensaje de alerta que “Los campos son requeridos”</p>

---

*Nota.* Se describe las acciones llevadas a cabo para gestionar la información de afiliación de los usuarios.

**Figura 22.**

*Diagrama de caso de uso - Registro de propiedad al afiliado*



*Nota.* Este caso de uso denota el proceso de registro de propiedad.

**Tabla 9.**

*Descripción de caso de uso - Registro de propiedad de afiliado*

Caso de Uso	Registro de propiedad de afiliado
Número del caso de uso:	DCU03
Descripción:	Permite registrar nuevas afiliaciones y actualizar las propiedades de un afiliado.
Actores:	Administrador, Técnico

---

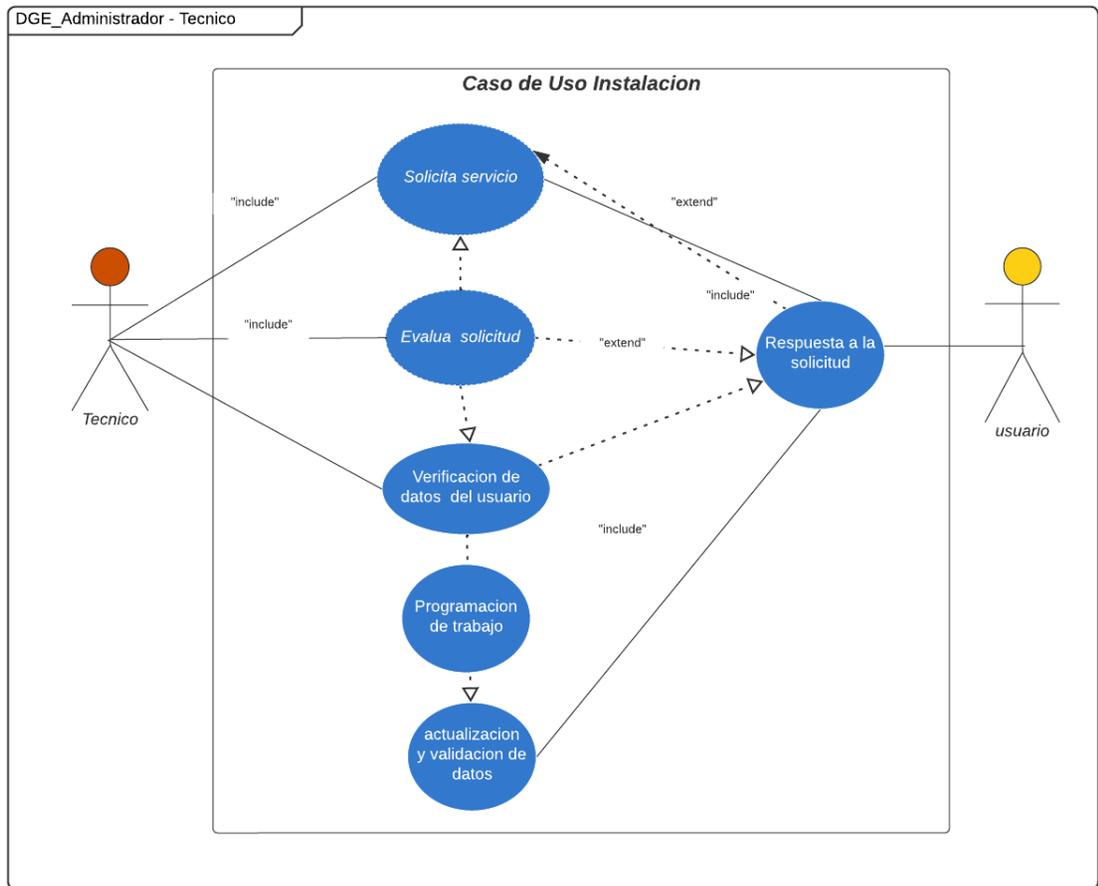
Precondiciones:	El actor debe ser usuario del sistema y tener el rol correspondiente dentro del sistema.
Flujo normal de Eventos:	<p>Autenticación del actor en el sistema. Elegir la opción de registrar propiedad, mostrando el listado de los afiliados, persona natural y jurídica.</p> <p>El sistema valida que la información ingresada sea correcta para una propiedad.</p> <p>El sistema registra la nueva propiedad con la información introducida.</p> <p>Registrar el estado según el criterio del responsable.</p> <p style="text-align: center;">Salir del Sistema.</p>
Post Condiciones:	El sistema registra los cambios y actualizaciones de las propiedades pertenecientes a cada afiliado.
Flujos alternativos:	Si los campos están vacíos el sistema mostrará mensaje de error “Los campos son requeridos”

---

*Nota.* Descripción del proceso de registro de propiedad de afiliado.

**Figura 23.**

*Caso de uso instalación del servicio*



*Nota.* En este caso de uso denota la instalación del servicio.

**Tabla 10.**

*Descripción de caso de uso - Instalación del servicio*

Caso de Uso	Instalación del servicio
Número del caso de uso:	DCU04
Descripción:	Permite registrar nuevas instalaciones y actualizar los nuevos servicios de un afiliado.

---

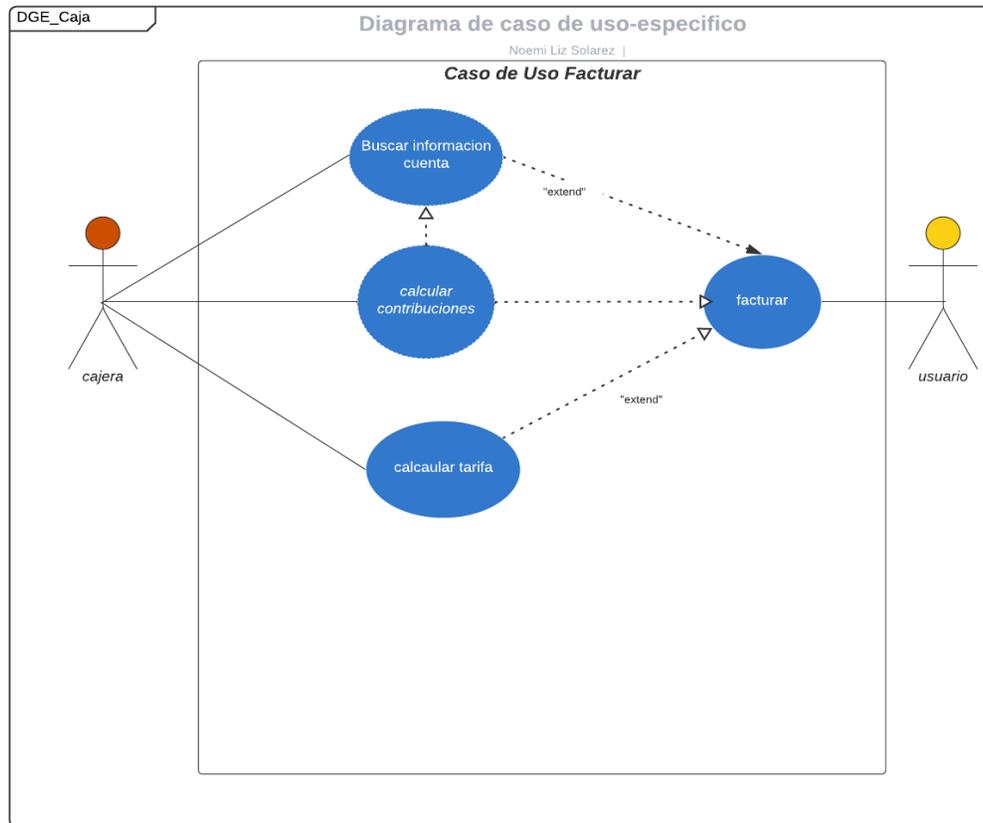
Actores:	Administrador, Técnico.
Precondiciones:	<p>El usuario debe estar registrado como Técnico o administrador. Datos de permisos ingresados con anterioridad, haber iniciado la sesión con éxito.</p>
Flujo normal de Eventos:	<ul style="list-style-type: none"> <li>• Autenticación del actor en el sistema. Elegir la opción de nueva instalación.</li> <li>• Revisar la información de solicitud de instalación de los nuevos afiliados.</li> <li>• Registrar el estado según el criterio del responsable.</li> <li>• Salir del Sistema</li> </ul>
Post Condiciones:	<p>El sistema registra los cambios y actualizaciones de las propiedades pertenecientes a cada afiliado.</p>
Flujos alternativos:	<p>Si los campos están vacíos el sistema mostrará mensaje de error “Los campos son requeridos”</p>

---

*Nota.* Descripción del proceso de instalación del servicio.

**Figura 24**

Caso de uso – Categorización de cobro del servicio



Nota. Caso de uso donde se realiza el cobro del servicio.

**Figura 25.**

Descripción del caso de uso - cobro del servicio

Caso de Uso	Cobro del servicio
Número del caso de uso:	DCU05
Descripción:	Tienen acceso a la apertura y regularización de cobro y la administración de la estructura de normas.

---

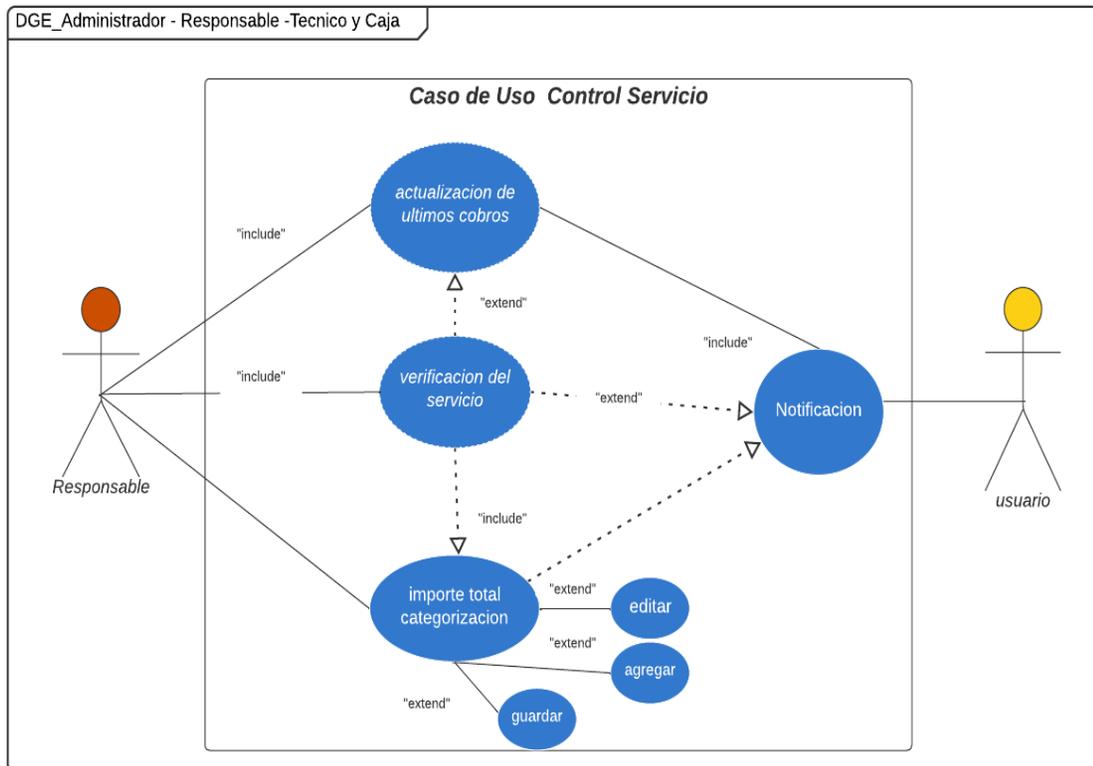
Actores:	Administrador, técnico, caja.
Precondiciones:	<p>El usuario debe estar registrado como administrador. Datos de permisos ingresados con anterioridad, haber iniciado la sesión con éxito.</p> <p>El caso de uso comienza cuando ingresa al menú de administración, técnico o caja.</p>
Flujo normal de Eventos:	<ul style="list-style-type: none"> <li>• El sistema desplegara una lista de los usuarios natural y jurídica para el registro de pago.</li> <li>• Se despliega las opciones de cobro de todas sus propiedades donde podrá seleccionar la cantidad que se cancelará.</li> </ul>
Post Condiciones:	Registro de cobro, la recaudación se realiza correctamente.
Flujos alternativos:	Si los campos están vacíos el sistema mostrará mensaje de error “Los campos son requeridos”

---

*Nota.* Descripción de todos los pasos que debe seguir para el cobro del servicio.

**Figura 26.**

*Caso de uso - control del servicio*



*Nota.* Caso de uso donde se realiza el control del servicio.

**Tabla 11.**

*Descripción del caso de uso - Control del servicio*

Caso de Uso	Gestionar Usuarios
Número del caso de uso:	DCU08
Descripción:	Tienen acceso a la administración de la información de los usuarios, para verificar el control de pagos de cada usuario.

---

Actores:	Administrador, Responsable, técnico y caja.
Precondiciones:	El usuario debe estar registrado como administrador. Datos de permisos ingresados con anterioridad, haber iniciado la sesión con éxito.
Flujo normal de Eventos:	
Post Condiciones:	
Flujos alternativos:	Si los campos están vacíos el sistema mostrará mensaje de error “Los campos son requeridos”

---

*Nota.* Descripción de todos los pasos que se debe seguir para realizar el control.

**Tabla 12.**

*Descripción de caso de uso - reportes administrativas del suministro del servicio*

Caso de Uso	Gestionar Usuarios
Número del caso de uso:	DCU09
Descripción:	
Actores:	Administrador, Responsable, técnico y caja.
Precondiciones:	El usuario debe estar registrado como administrador. Datos de permisos

---

---

ingresados con anterioridad, haber iniciado la sesión con éxito.

Flujo normal de Eventos:

Post Condiciones:

Flujos alternativos:

Si los campos están vacíos el sistema mostrará mensaje de error “Los campos son requeridos”

---

*Nota.* Descripción de los pasos que se debe seguir para realizar los reportes del servicio.

#### **3.2.4.4. Diseño conceptual**

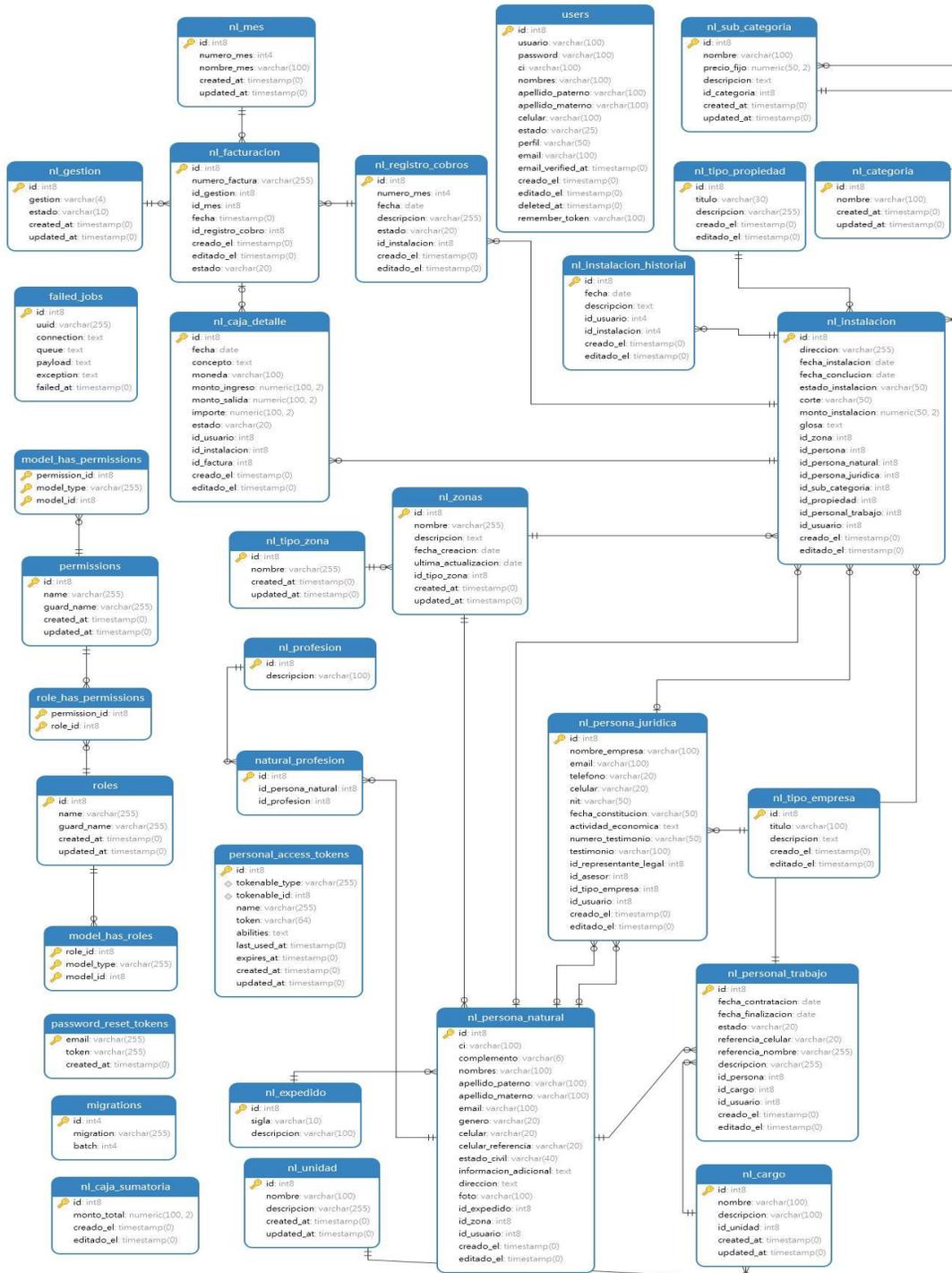
En esta etapa, se construirá un Modelo Conceptual a partir de los Indicadores y Perspectivas obtenidas en el paso anterior.

La representación gráfica del Modelo Conceptual es la siguiente:

A través de este Modelo, se podrá observar con claridad cuáles son los alcances del proyecto, para luego poder trabajar sobre ellos.

Figura 27.

Diseño conceptual



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

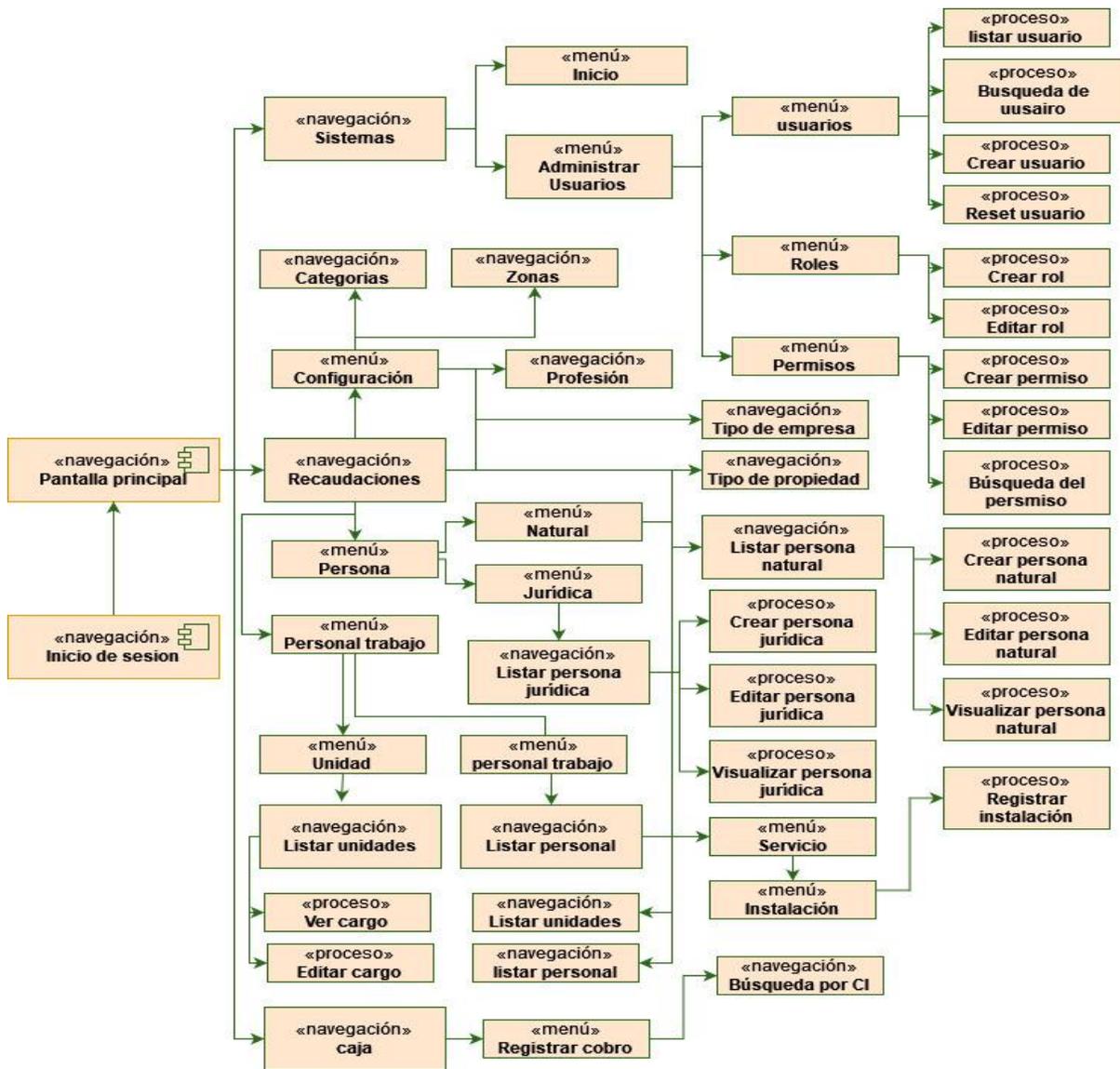
### 3.2.4.5. Modelo de navegación

A continuación, se muestra el diseño del sistema de la interfaz de toma en cuenta una interfaz amigable, visible que reconozca con el nombre de la institución que permite seleccionar acciones fácilmente.

- **Modelo de navegación - Administrador**

Figura 28.

Modelo de navegación – Administrador

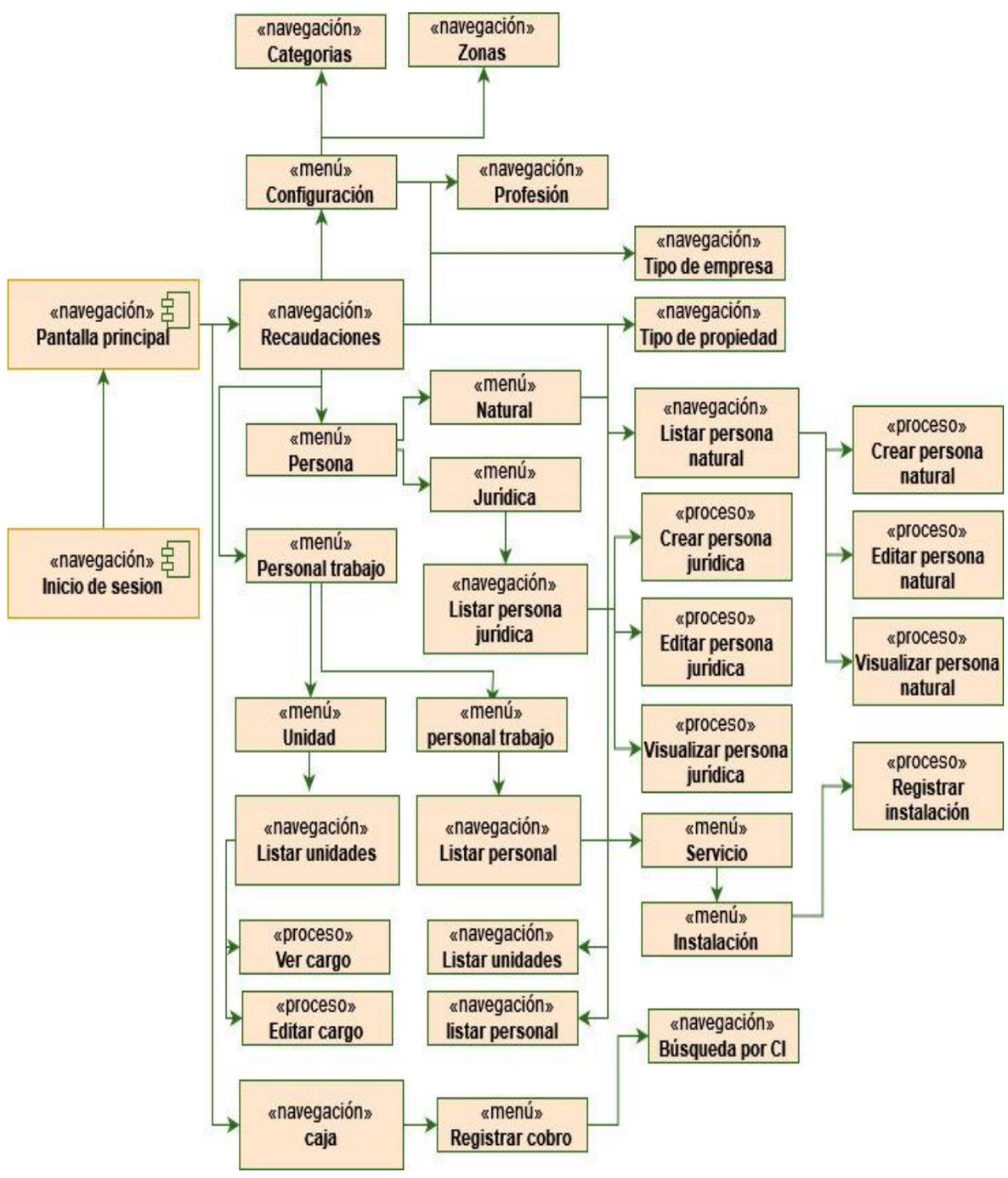


Nota. Modelo navegacional del rol administrador.

- Modelo de navegación – Técnico

Figura 29.

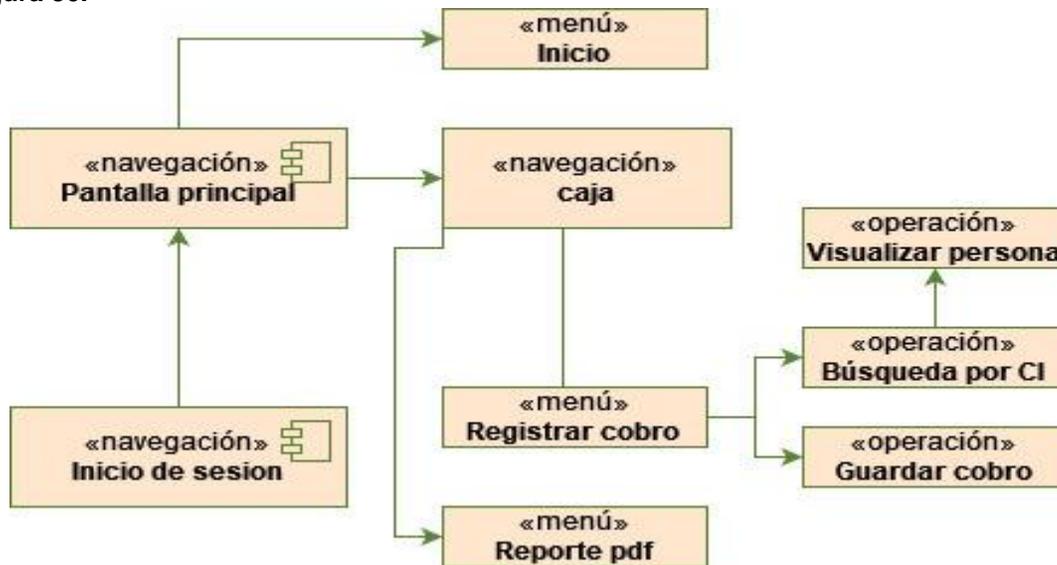
Modelo de navegación - técnico



Nota. Modelo navegacional del rol técnico.

- **Modelo de navegación – cajero**

Figura 30.



Modelos de navegación - cajero

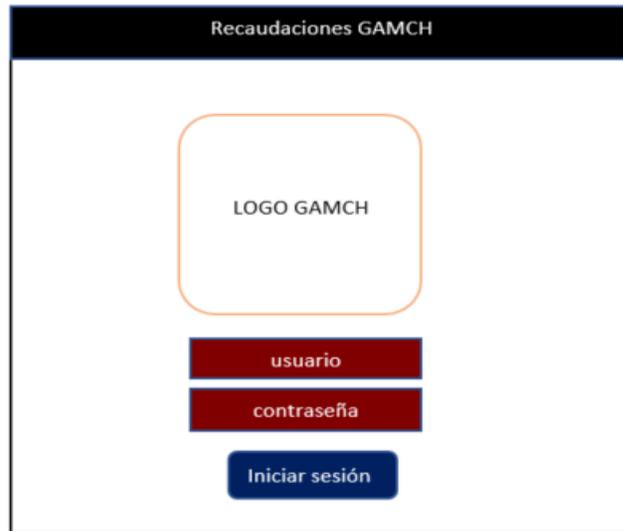
Nota. Modelo navegacional del cajero.

### 3.2.4.6. Modelo de presentación

El modelo de presentación tiene la visión general de la interfaz de usuario, describe la estructura de la interfaz que pueden contener texto, imágenes, enlaces, formularios utilizados para representar la forma de navegación.

**Figura 31.**

*Modelo de presentación - Login*



*Nota.* Elaboración propia.

**Figura 32.**

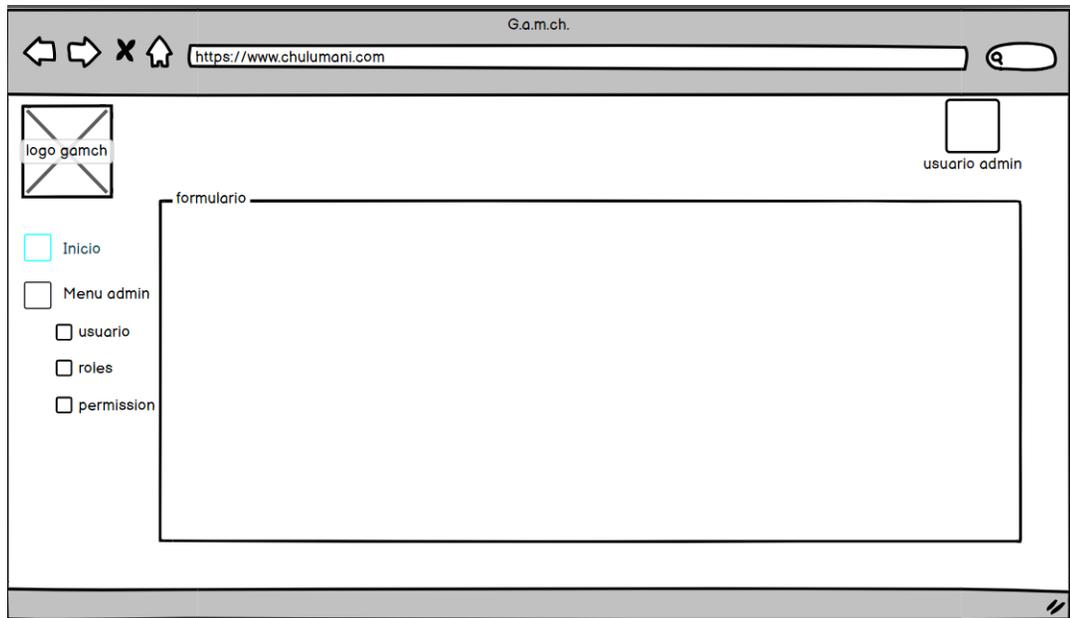
*Modelo de presentación - inicio*



*Nota.* Elaboración propia.

**Figura 33.**

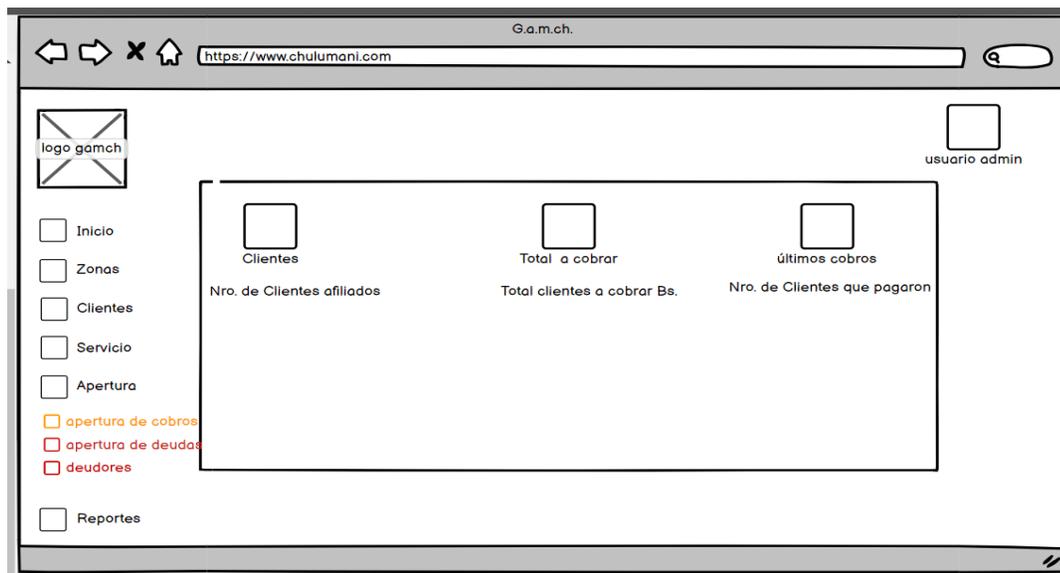
*Modelo de presentación - sistemas*



*Nota. Elaboración propia.*

**Figura 34.**

*Modelo de presentación - recaudaciones*



*Nota. Elaboración propia.*

**Figura 35.**

Categoría de cobro

Recaudaciones GAMCH

LOGO GAMCH

MENU

- Inicio
- Categoría**
- Personas
- Personal
- Servicio
- Reportes

CATEGORÍA DE COBRO DE AGUA

+ AGREGAR

NRO	TIPO CONSUMO	SUB CATEGORIA	ACCION
			EDITAR ELIMINAR

Nota. Elaboración propia.

**Figura 36.**

Registro de Zonas

Recaudaciones GAMCH

LOGO GAMCH

MENU

- Inicio configuración
- Tipo zona**
- Categoría
- Zonas
- Profesión
- Expedido
- Tipo de empresa
- Tipo propiedad

- Personas
- Personal
- Servicio
- Reportes

TIPOS DE ZONAS

+ AGREGAR

NRO	NOMBRE	ACCION
		EDITAR ELIMINAR

Nota. Elaboración propia.

**Figura 37.**

Registro de nuevas zonas

Recaudaciones GAMCH

LOGO GAMCH

MENU

- Inicio
- configuración
  - Categoría
  - Tipo zona
  - Zonas**
  - Profesión
  - Expedido
  - Tipo de empresa
  - Tipo propiedad
- Personas
- Personal
- Servicio
- Reportes

ZONAS

+ AGREGAR

NRO	NOMBRE	DESCRIPCION	FECHA DE CREACION	TIPO ZONA	ULTIMA MODIF.	ACCION
						EDITA ELIMINAR
						EDITA ELIMINAR
						EDITA ELIMINAR
						EDITA ELIMINAR

Nota. Elaboración propia.

**Figura 38.**

Registro datos profesiones

Recaudaciones GAMCH

LOGO GAMCH

MENU

- Inicio
- configuración
  - Categoría
  - Tipo zona
  - Zonas
  - Profesión**
  - Expedido
  - Tipo de empresa
  - Tipo propiedad
- Personas
- Personal
- Servicio
- Reportes

PROFESION

+ AGREGAR

NRO	DESCRIPCION	ACCION
		EDITA ELIMINAR

Nota. Elaboración propia.

**Figura 39.**

Registro tipo de empresa

Recaudaciones GAMCH

LOGO GAMCH

MENU

Inicio configuración

- Categoría
- Tipo zona
- Zonas
- Profesión
- Expedido
- **Tipo de empresa**
- Tipo propiedad

Personas

Personal

Servicio

Reportes

TIPO EMPRESA

+ AGREGAR

NRO	TITULO	DESCRIPCION	ACCION	
			EDITA	ELIMINAR

Nota. Elaboración propia.

**Figura 40.**

Registro tipo de propiedad

Recaudaciones GAMCH

LOGO GAMCH

MENU

Inicio configuración

- Categoría
- Tipo zona
- Zonas
- Profesión
- Expedido
- Tipo de empresa
- **Tipo propiedad**

Personas

Personal

Servicio

Reportes

TIPO DE PROPIEDAD

+ AGREGAR

NRO	TITULO	DESCRIPCION	ACCION	
			EDITA	ELIMINAR

Nota. Elaboración propia.

**Figura 41.**

*Afiliación persona natural*

Recaudaciones GAMCH

LOGO  
GAMCH

**VISTA AFILIACIÓN PERSONA NATURAL** + AGREGAR

NRO	CI	NOMBRES Y APELLIDOS	CELULAR	ACCION		
1				VER	EDITAR	ELIMINAR
2				VER	EDITAR	ELIMINAR
3				VER	EDITAR	ELIMINAR
4				VER	EDITAR	ELIMINAR

**MENU**

Inicio

Categoría

Personas

- Persona natural
- Persona jurídica

Personal

Servicio

Reportes

Nota. Elaboración propia.

**Figura 42.**

*Afiliación persona jurídica*

Recaudaciones GAMCH

LOGO  
GAMCH

**VISTA AFILIACION PERSONA JURIDICA** + AGREGAR

NRO	NOMBRE EMPRESA	EMAIL	NIT	NRO TESTIMONIO	TELEFONO	ACCION	
						EDITAR	ELIMINAR
						EDITAR	ELIMINAR
						EDITAR	ELIMINAR
						EDITAR	ELIMINAR

**MENU**

Inicio

Categoría

Personas

- Persona natural
- Persona jurídica

Personal

Servicio

Reportes

Nota. Elaboración propia.

**Figura 43.**

Vista unidades de trabajo



Nota. Elaboración propia.

**Figura 44.**

Vista para adicionar personal de trabajo en las áreas



Nota. Elaboración propia.

Figura 45.

Instalación de servicio

NRO	TIPO	FECHA DE INSTALACION	MONTO	ESTADO	ACCION
				<input type="checkbox"/>	<input type="button" value="EDITAR"/> <input type="button" value="VER"/> <input type="button" value="PDF"/>
				<input type="checkbox"/>	<input type="button" value="EDITAR"/> <input type="button" value="VER"/> <input type="button" value="PDF"/>
				<input type="checkbox"/>	<input type="button" value="EDITAR"/> <input type="button" value="VER"/> <input type="button" value="PDF"/>
				<input type="checkbox"/>	<input type="button" value="EDITAR"/> <input type="button" value="VER"/> <input type="button" value="PDF"/>

Nota. Elaboración propia.

### 3.2.4.7. Modelo de implementación

Figura 46.

Login

USUARIO MUNICIPAL DE CHULLUMANI

LOGIN

Usuario  
Ingrese su usuario

Contraseña  
Ingrese su contraseña

INGRESAR

NIIZ All Rights Reserved.

CHULLUMANI

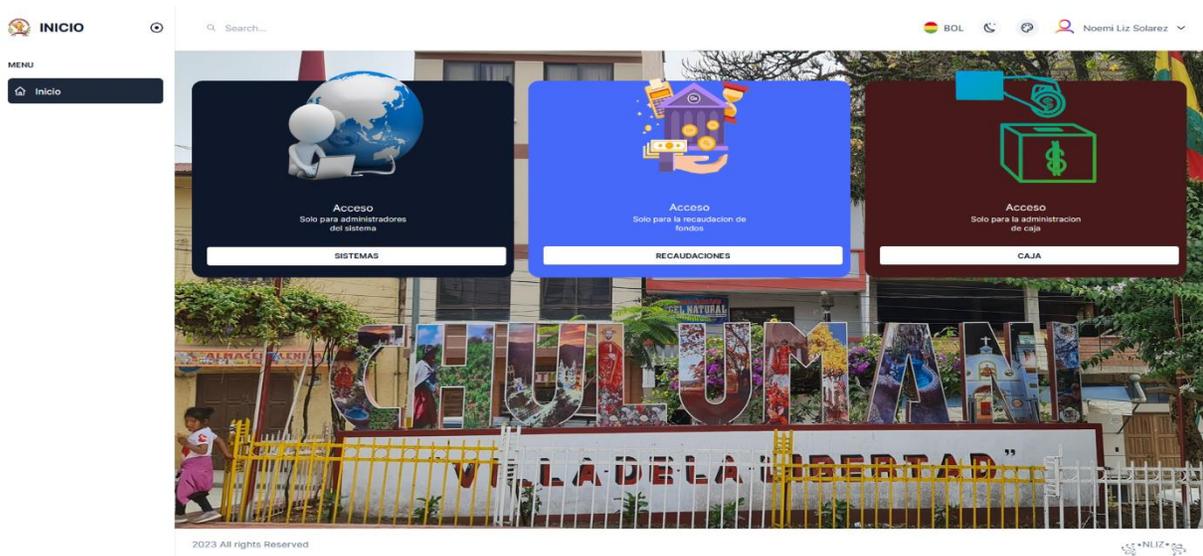
TODOS SOMOS CHULLUMANI

NIIZ

Nota. Elaboración propia.

Figura 47.

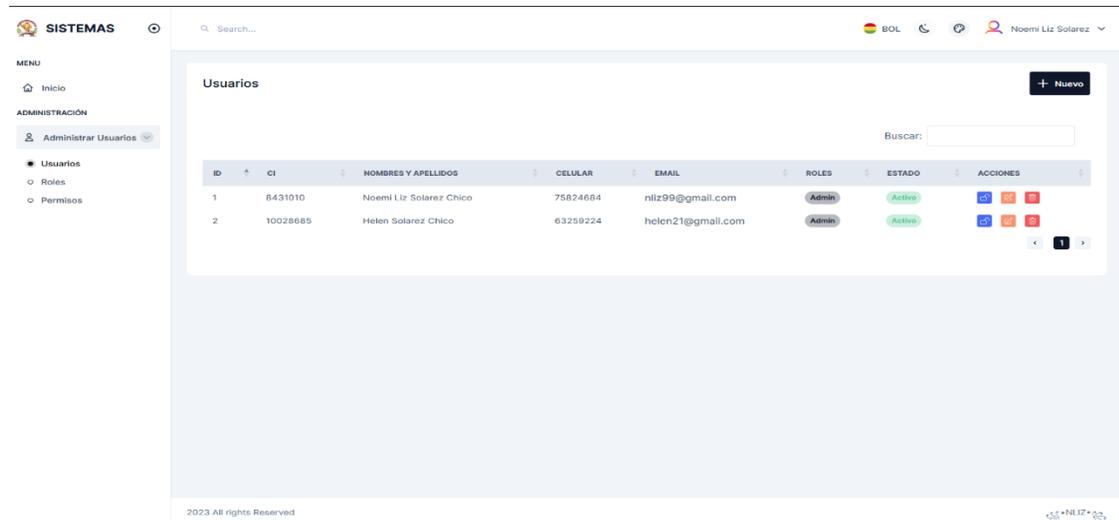
Seleccionar acceso



Nota. Elaboración propia.

Figura 48.

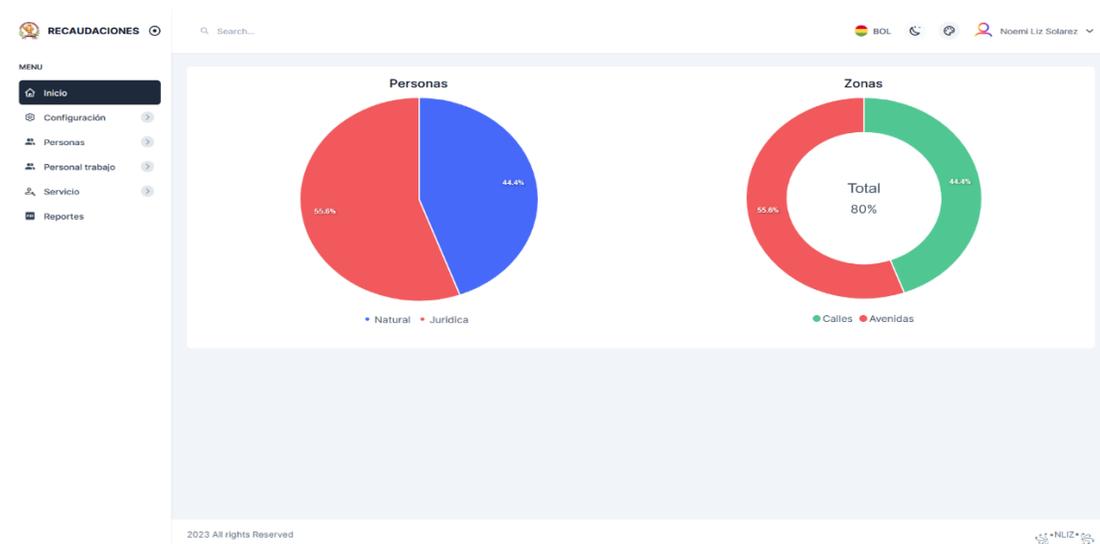
Administrar usuarios



Nota. Elaboración propia.

**Figura 49.**

*Administrar recaudaciones*



*Nota.* Elaboración propia.

**Figura 50.**

*Configuración de categorías*

The screenshot shows the configuration page for 'Categoría cobro de agua'. The table lists the following categories:

ID	TIPO DE CONSUMO	SUB CATEGORÍA	ACCIONES
1	Familia Básica	Sub Categoría	[Edit] [Delete]
2	Familiar	Sub Categoría	[Edit] [Delete]
3	Comercio	Sub Categoría	[Edit] [Delete]
4	Baño	Sub Categoría	[Edit] [Delete]
5	Piscina	Sub Categoría	[Edit] [Delete]

The interface includes a sidebar menu with options like 'Inicio', 'Configuración', 'Categorías', 'Tipo Zona', 'Zonas', 'Profesión', 'Expedido', 'Tipo de empresa', 'Tipo de propiedad', 'Personas', 'Personal trabajo', 'Servicio', and 'Reportes'. The top right corner shows the user 'Noemi Liz Solarez' and the country 'BOL'.

*Nota.* Elaboración propia.

**Figura 51.**

*Tipo de zonas*

ID	NOMBRE	ACCIONES
1	Reserva Natural	[Red -] [Red +]
2	Comunitaria	[Red -] [Red +]
3	Gubernamental	[Red -] [Red +]
4	Gastronómica	[Red -] [Red +]
5	Histórica	[Red -] [Red +]
6	Sanitaria	[Red -] [Red +]
7	Educativa	[Red -] [Red +]
8	Turística	[Red -] [Red +]
9	Rural	[Red -] [Red +]
10	Industrial	[Red -] [Red +]
11	Comercial	[Red -] [Red +]
12	Residencial	[Red -] [Red +]

*Nota.* Elaboración propia.

**Figura 52.**

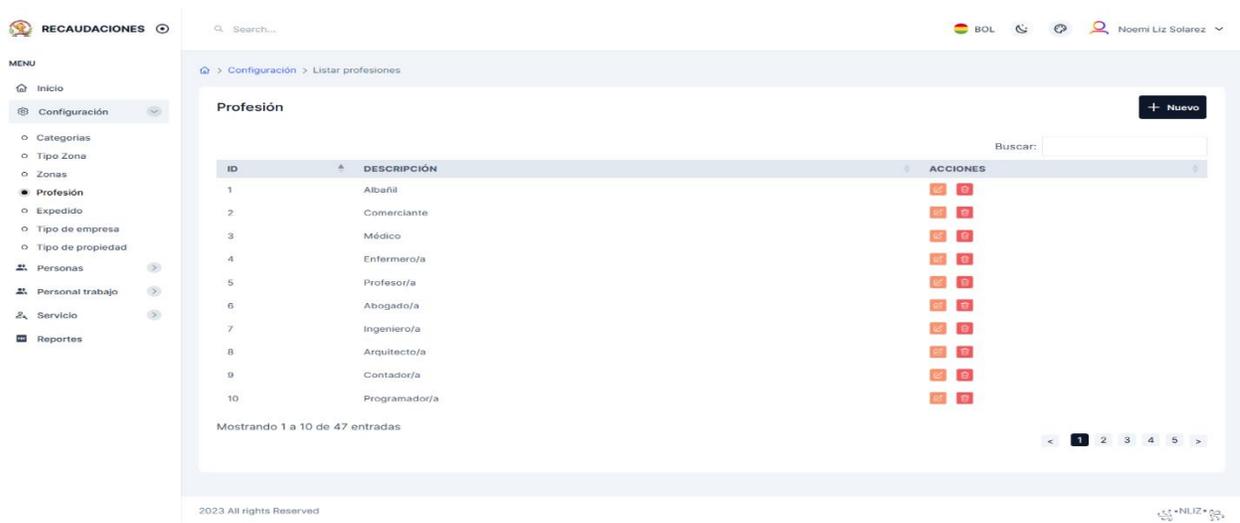
*Administración de zonas*

ID	NOMBRE	DESCRIPCIÓN	FECHA DE CREACIÓN	TIPO DE ZONA	ULTIMA MODIFICACIÓN	ACCIONES
1	Centro urbano	Es la zona principal de la ciudad, donde se encuentran las principales instituciones públicas, privadas y comerciales. Aquí también se encuentran la plaza principal, la iglesia católica y otros atractivos turísticos.	noviembre 3, 1997	Comercial	noviembre 7, 2023	[Red -] [Red +]
2	Barrios periféricos	Los barrios periféricos se encuentran alrededor del centro urbano. Son zonas residenciales donde viven la mayoría de los habitantes de Chulumani.	mayo 2, 2002	Residencial	noviembre 7, 2023	[Red -] [Red +]
3	Áreas rurales	Las áreas rurales se encuentran fuera del centro urbano. Estas zonas son dedicadas principalmente a la agricultura y la ganadería.	noviembre 25, 1999	Comunitaria	noviembre 7, 2023	[Red -] [Red +]

*Nota.* Elaboración propia.

**Figura 53.**

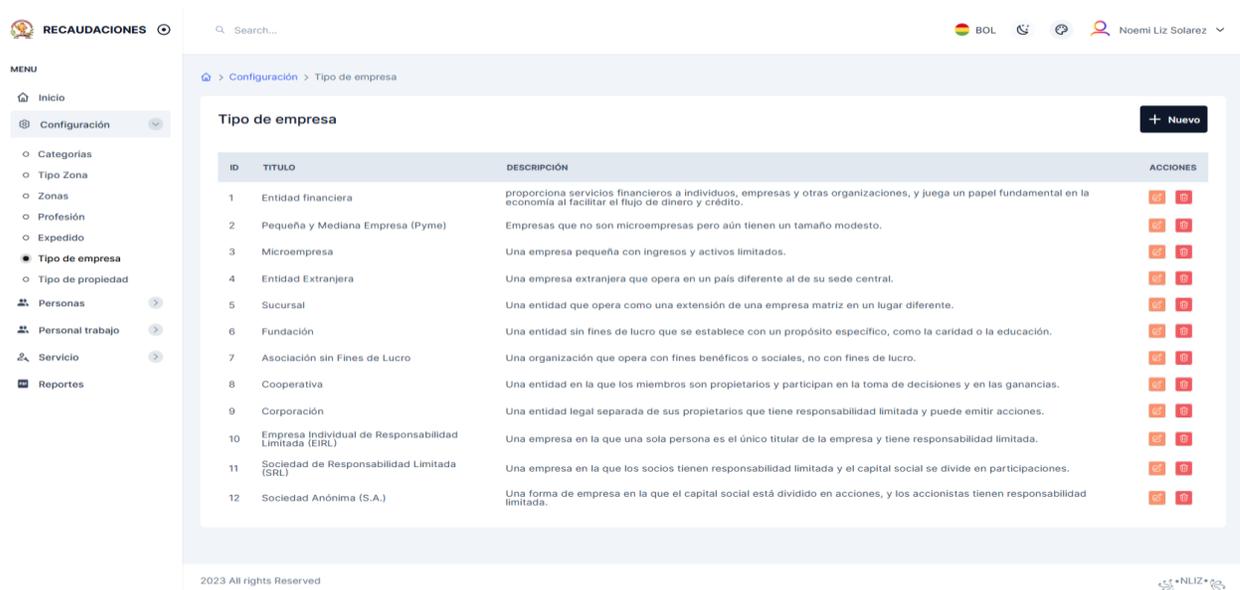
*Administrar profesiones*



*Nota. Elaboración propia.*

**Figura 54.**

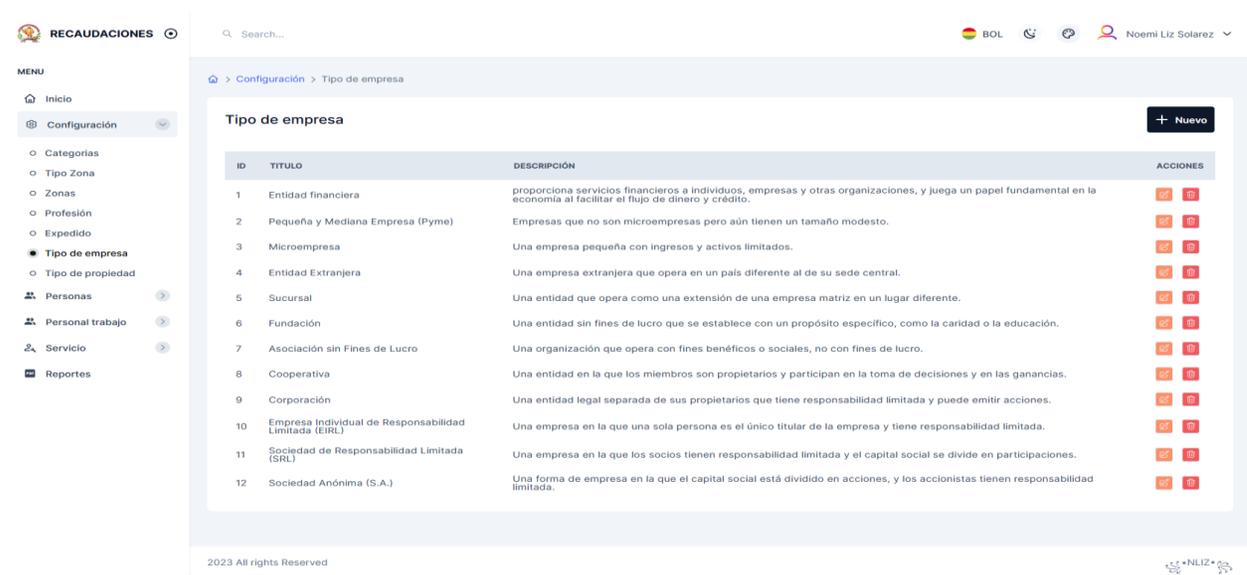
*Tipo de empresas*



*Nota. Elaboración propia.*

Figura 55.

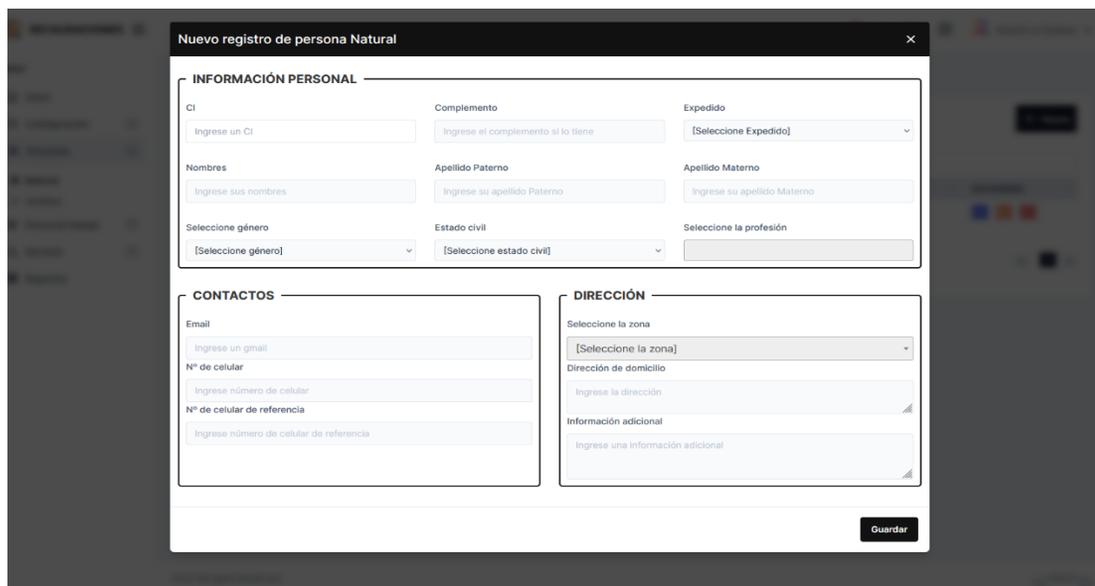
Tipo de propiedad



Nota. Elaboración propia.

Figura 56.

Registro de persona natural



Nota. Elaboración propia.

Figura 57.

Registro de persona jurídica

**Nuevo registro de persona Jurídica**

**PERSONA**

SELECCIONE EL TIPO DE EMPRESA: [Seleccione el tipo de empresa]

REPRESENTANTE LEGAL: Ingrese el nombre del representante legal

**DOCUMENTOS**

DESCRIPCIÓN DE LA ACTIVIDAD ECONOMICA: Describa la actividad económica

NÚMERO DE TESTIMONIO: Ingrese el número de testimonio

SELECCIONE EL TESTIMONIO: Examinar... No se ha seleccionado ningún archivo.

**INFORMACIÓN**

NIT: Ingrese el n° de NIT

NOMBRE DE LA EMPRESA: Ingrese el nombre de la empresa

Nº TELEFONO: Ingrese número de teléfono

Nº DE CELULAR: Ingrese número de celular

EMAIL: Ingrese número de celular

FECHA DE CONSTITUCIÓN: dd / mm / aaaa

Guardar

Nota. Elaboración propia.

Figura 58.

Registro de unidades

**RECAUDACIONES** Search... BOL Noemil Liz Solarez

MENU

- Inicio
- Configuración
- Personas
- Personal Trabajo
  - Unidad
  - Personal
  - Servicio
  - Reportes

Personal trabajo > Unidad

**Listado de unidad** + Nuevo

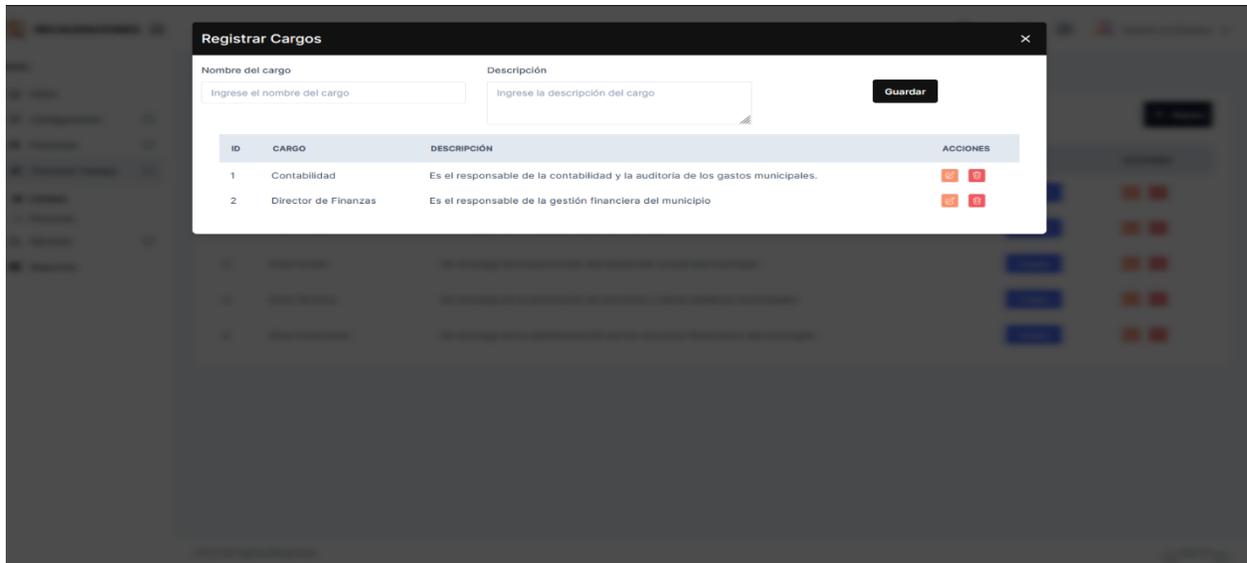
ID	NOMBRE	DESCRIPCIÓN	CARGOS	ACCIONES
1	Área Administrativa	Se encarga de la gestión administrativa del municipio	Cargos	[+][-]
2	Área Jurídica	Se encarga de la asesoría legal del municipio	Cargos	[+][-]
3	Área Social	Se encarga de la promoción del desarrollo social del municipio	Cargos	[+][-]
4	Área Técnica	Se encarga de la prestación de servicios y obras públicas municipales	Cargos	[+][-]
5	Área Financiera	Se encarga de la administración de los recursos financieros del municipio	Cargos	[+][-]

2023 All rights Reserved

Nota. Elaboración propia.

**Figura 59.**

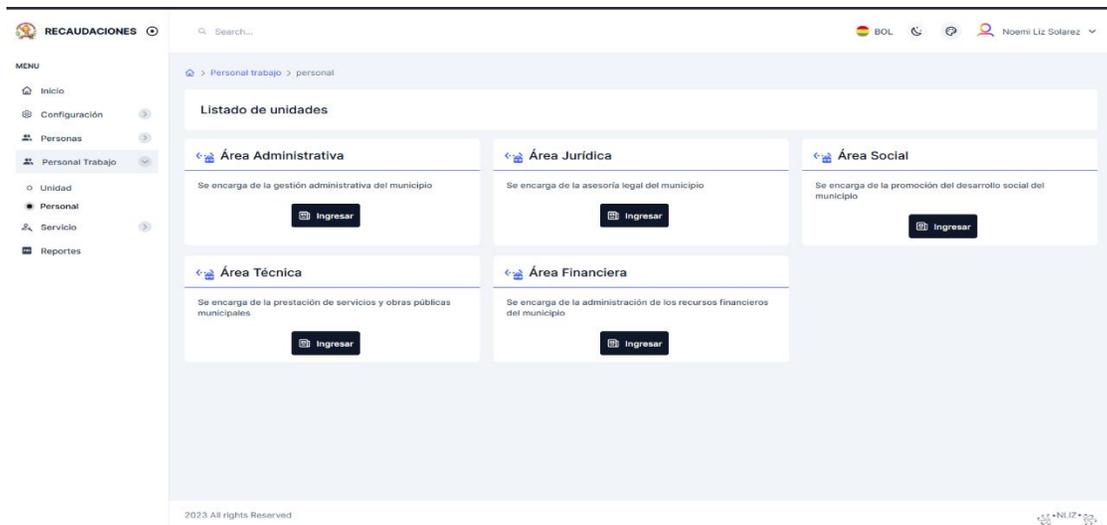
*Registro de cargos*



*Nota.* Elaboración propia.

**Figura 60.**

*Registro de personal*



*Nota.* Elaboración propia.

**Figura 61.**

*Registro del personal*

**PERSONA**

INGRESE CI DE PERSONA  
Ingrese ci persona

**INFORMACIÓN**

FECHA DE CONTRATACIÓN: dd / mm / aaaa

CARGO: [Seleccione cargo]

REFERENCIA LABORAL (TELEFONO O CELULAR): Ingrese referencia laboral

REFERENCIA LABORAL (NOMBRE DE LA PERSONA): Ingrese el nombre de la referencia laboral

Guardar

*Nota.* Elaboración propia.

**Figura 62.**

*Registrar instalación*

**NUEVO REGISTRO DE INSTALACIÓN**

**INSTALACIÓN**

INGRESE CI: Ingrese CI de la persona

SELECCIONE: [Seleccione persona jurídica]

**INSTALACIÓN**

SELECCIONE LA FECHA DE INSTALACIÓN: dd / mm / aaaa

SELECCIONE LA GESTIÓN: [Seleccione la gestión]

SELECCIONE LA CATEGORÍA: [Seleccione la categoría]

INGRESE EL MONTO DE INSTALACIÓN: Ingrese el monto de instalación

GLOSA: Describa la razón por la cual desea instalar el servicio

**DIRECCION**

SELECCIONE LA PROPIEDAD: [Seleccione el tipo de propiedad]

SELECCIONE LA ZONA: [Seleccione la zona]

DIRECCIÓN DONDE SE HARA LA INSTALACIÓN: Ingrese la dirección donde se realizara la instalación

Guardar

*Nota.* Elaboración propia.

### 3.2.5. Codificación.

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

Figura 63.

Codificación principal del software.

```
<?php
namespace App\Http\Controllers\Persona;

use App\Http\Controllers\Controller;
use App\Models\Configuracion\Expedido;
use App\Models\Configuracion\Profesion;
use App\Models\Configuracion\Tipo_empresa;
use App\Models\Configuracion\Zonas;
use App\Models\Persona\Juridica;
use App\Models\Persona\Natural;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Validator;
use Illuminate\Support\Facades\Auth;

class Controlador_persona extends Controller{
    /**
     * @version 1.0
     * @author Noemi Liz Solarez Chico <noemilizsolarez@gmail.com>
     * @param Controlador Administracion de la parte del registro de personas Naturales y Juridicas
     * ¡Muchas gracias por preferirnos! Esperamos poder servirte nuevamente
     */
    /**
     * PERSONAS NATURALES
     */
    public function persona_natural(){
        $data['menu'] = '11';
        $data['expedido'] = Expedido::get();
        $data['profesion'] = Profesion::get();
        $data['zonas'] = Zonas::get();
        $data['listar_persona_natural'] = Natural::get();
        return view('administrador.recaudaciones.persona.natural', $data);
    }

    //para listar las personas naturales
    public function personaNatural_listar(){
        $persona_natural = Natural::with(['expedido','profesion','zona'])->get();
        return response()->json($persona_natural);
    }
}
```

Nota. Elaboración propia.

Figura 64.

Codificación de datos mediante el ORM

```
//relacion reversa de expedido
public function expedido(){
    return $this->belongsTo(Expedido::class, 'id_expedido', 'id');
}

//relacion de muchos a muchos con profesion
public function profesion(){
    return $this->belongsToMany(Profesion::class, 'natural_profesion', 'id_persona_natural', 'id_profesion');
}

//relacion reversa con zona
public function zona(){
    return $this->belongsTo(Zonas::class, 'id_zona', 'id');
}

//relacion con persona juridica para el id_representante_legal
public function juridica_representante_legal(){
    return $this->hasMany(Juridica::class, 'id_representante_legal', 'id');
}
```

Nota. Elaboración propia.

**Figura 65.**

Codificación parte de la administración de usuarios.

```
public function ingresar(Request $request){
    //par arecuperar el cpacha de la session
    $captcha_recuperado = session()->get('captchaText_recuperado');
    //eliminar los estilos
    $captcha_texto = strip_tags($captcha_recuperado);

    $mensaje = "Usuario y contraseña invalidos";
    $validar = Validator::make($request->all(),[
        'usuario' => 'required',
        'password' => 'required',
        'captcha' => 'required'
    ]);
    if($validar->fails()){
        $data = mensaje_mostrar('error', 'Todos los campos son requeridos');
    }else{
        if($request->captcha == $captcha_texto){
            //comprobamos si el usuario existe o si esta activo
            $usuario = User::where('usuario', $request->usuario)->get();
            if(!$usuario->isEmpty()){
                $compara = Auth::attempt([
                    'usuario' => $request->usuario,
                    'password' => $request->password,
                    'estado' => 'activo',
                    'deleted_at' => NULL
                ]);
                if($compara){
                    $data = mensaje_mostrar('success', 'Inicio de session con éxito');
                    $request->session()->regenerate();
                }else{
                    $data = mensaje_mostrar('error', $mensaje);
                }
            }else{
                $data = mensaje_mostrar('error', $mensaje);
            }
        }else{
            $data = mensaje_mostrar('error', $mensaje);
        }
    }
    return response()->json($data);
}
```

Nota. Elaboración propia.

**CAPÍTULO IV**  
**PRUEBAS DE EVALUACIÓN**  
**Y RESULTADOS**

## 4. PRUEBAS DE EVALUACIÓN Y RESULTADOS

### 4.1. MÉTRICAS DE CALIDAD ISO 25000

La norma internacional ISO/IEC 25000, tiene por objetivo la creación de un marco de trabajo para definir los requisitos y evaluar la calidad del producto software, se desarrolla la evaluación de calidad del presente proyecto, para brindarle al usuario, facilidad, ahorro económico y dar al cliente seguridad en su información.

#### 4.1.1. Funcionalidad

La funcionalidad en esta técnica de estimación de software, se toma en cuenta 5 características:

- Número de entradas de usuario

**Tabla 13.**

*Número de entradas de usuario*

<b>Entradas de usuario</b>	<b>N.º</b>
Administración de usuarios	1
Administración de técnicos	1
Administración de cajeros	2
Administración de afiliados	26
<b>Total</b>	<b>30</b>

*Nota.* Se determina la cantidad de entradas de usuario.

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

**Tabla 14.**

*Número de salidas de usuario*

<b>Salidas de usuario</b>	<b>N.º</b>
Administración de usuarios	3
Administración de técnicos	3
Administración de cajeros	1
Administración de afiliados	20
<b>Total</b>	<b>25</b>

*Nota.* Se determina la cantidad de salidas de usuario.

- Número de peticiones de usuario: Son las entradas interactivas que genera una respuesta del software como salidas interactivas

**Tabla 15.**

*Peticiones de usuario*

<b>Peticiones de usuario</b>	<b>N.º</b>
Administración de usuarios	4
Administración de técnicos	3
Administración de cajeros	1
Administración de afiliados	6
<b>Total</b>	<b>14</b>

*Nota.* Se determina la cantidad de peticiones de usuario.

- Numero de archivos: Se toma en los grupos lógicos de datos o que son archivos independientes.

- Numero de interfaces externas: Es la suma de todas las interfaces que son leíbles por la máquina.

Parámetros de medición	Cuenta
Número de entradas de usuario	30
Número de salidas de usuario	25
Número de peticiones de usuario	14
Número de archivos	10
Número de interfaz Externa	1

*Nota.* Se determina de las peticiones externas.

Ya realizado el conteo del parámetro de función en la tabla se realiza el cálculo del punto de función sin ajustar en la siguiente tabla:

**Tabla 16.**

*Punto función sin ajustar*

Parámetro de medición	Cuenta	Factor	Total
Número de entradas de Usuario	30	7	210
Número de salidas de Usuario	25	5	125
Número de peticiones de Usuario	14	6	84
Número de Archivos	10	15	150
Número de Interfaz Externa	1	9	9
Cuenta total			578

*Nota.* Se determina el punto de función sin ajustar.

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

**Tabla 17.**

*Factores de Complejidad*

Importancia	0%	20%	40%	60%	80%	100%
Escala	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	4
¿Es crítico el rendimiento?						x 5
¿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	5
¿Son complejas las entradas, salidas, los archivos o las peticiones?	x	5
¿Es complejo el procesamiento interno del sistema?	x	5
¿Se ha diseñado el código para ser reutilizado?	x	4
¿Se ha diseñado el sistema para facilitar al usuario el trabajo y ayudarlos a encontrar la información?	x	5
TOTAL		55

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

Para realizar el cálculo de punto de función ajustado se utiliza la siguiente formula:

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

**Donde:**

$$\sum (fi)$$

- Sumatoria de los valore de los factores de ajuste

Se tiene:

$$\sum (fi) = 55$$

$$PF = 578 * [0.65 + 0.01 * 55]$$

$$PF = 693.6$$

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

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

$$PFmax = 578 * [0.65 + (0.01 * 70)]$$

$$Pfmax = 780.3$$

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

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

$$Funcionalidad = \frac{693.6}{780.3} * 100$$

$$Funcionalidad = 0.88 * 100$$

$$\mathbf{Funcionalidad = 88.89\%}$$

#### **4.1.2. Confiabilidad**

La confiabilidad del presente proyecto, se medirá de acuerdo a la probabilidad de operación libre de fallos, los cuales son:

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

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

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

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

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

Donde:

*f*: funcionalidad del sistema

$\mu$  = Probabilidad de error que puede tener el sistema

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

En un periodo de 18 días como tiempo de prueba se tiene de cada 10 ejecuciones 1 falla

$$F(t) = 0.88 * e^{-\frac{1}{10} * 18}$$

$$F(t) = 0.1454 * 100$$

$$F(t) = 14.54\%$$

Reemplazando:

$$P(T \leq t) = 0.1454 = 14.54\%$$

$$P(T \leq t) = 1 - 0.1454$$

$$P(T \leq t) = 0.8546 * 100\%$$

$$P(T \leq t) = \mathbf{85.46\%}$$

#### **4.1.3. Usabilidad**

Para determinar la usabilidad, se realiza una evaluación para conocer si el sistema satisface los requerimientos establecidos por el usuario, entonces para esto se medirá de acuerdo a encuestas planteadas a los usuarios del sistema, y los resultados son los siguientes.

**Tabla 18.**

*Escala de Calificación*

<b>Escala</b>	<b>Valor</b>
Muy Bueno	5
Bueno	4
Regular	3
Malo	2

Pésimo	1
--------	---

Nota. Elaboración propia.

En la siguiente tabla nos muestra los resultados, de la encuesta que se realizó.

**Tabla 19.**

*Cuestionario de evaluación*

No	Pregunta	Si	No	Evaluación
1	¿Puede utilizar con facilidad el sistema?	5	1	0,9
2	¿Puede controlar operaciones que el sistema solicite?	4	2	0,8
3	¿El sistema permitió la retroalimentación de información?	5	0	1
4	¿El sistema cuenta con interfaz gráfica agradable a la vista?	4	1	0,8
5	¿La respuesta del sistema es satisfactoria?	4	1	0,8
6	¿Le parecen complicadas las funciones del sistema?	4	2	0,8
7	¿Le hace difícil aprender a manejar el sistema?	5	0	1
8	¿Puede fácilmente navegar por las distintas opciones que tiene el sistema?	5	0	1
9	¿Los resultados que proporciona el sistema facilitan el trabajo?	5	0	1
10	¿Durante el uso del sistema producto errores?	2	4	0,8

---

TOTAL

8,9

---

*Nota.* Se desarrollo el cuestionario de evaluación..

Para determinar la usabilidad del sistema se utiliza la siguiente ecuación:

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

Donde:

- **Xi** = Valores de evaluación de cada pregunta
- **n** = Numero de preguntas

Realizando los cálculos con los datos obtenidos:

$$FU = \frac{8,9}{10} * 100\% = 89 \%$$

De acuerdo a estos cálculos, se tiene un 89% de usabilidad, eso quiere decir una comprensión o entendimiento de los usuarios en el porcentaje mencionado.

#### **4.1.4. Eficiencia**

La eficiencia mide si el sistema emplea óptimamente los recursos del sistema, lo cual evalúa la eficiencia del sistema y las llamadas del sistema de aplicación.

Para calcular el grado de eficiencia que tiene el sistema se toma en cuenta la siguiente escala:

**Tabla 20.**

*Escala de valores de eficiencia*

<b>Escala</b>	<b>Valor</b>
Muy Bueno	5
Bueno	4

Regular	3
Malo	2
Pésimo	1

*Nota.* Elaboración propia.

Dada la siguiente tabla nos muestra el valor en la eficiencia del sistema.

**Tabla 21.**

*Valoración en la eficiencia del sistema*

No	Pregunta	%
1	¿Las operaciones jerárquicas se organiza de manera en que se minimizan para la navegación del usuario, para que otra dicha tarea se ejecute de manera normal?	5
2	¿En la secuencia de entrada de datos se realiza con mayor facilidad?	5
3	¿para la distribución de estilo de la interfaz de usuario permite a la eficiencia de la operación con la información?	4
4	¿Los datos de salida se representa de la manera en que se entiende de inmediato?	4
5	¿procesa y responde adecuadamente cuando se realiza alguna consulta o búsqueda?	5
TOTAL		23

*Nota.* Elaboración propia.

Para la medición de dicho computo de elementos de datos es:

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

**Donde:**

$\Sigma xi$  = sumatoria de los valores de eficiencia

**n** = Numero de preguntas

Reemplazando los datos tenemos que:

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

$$E = 87\%$$

por lo tanto, obtenemos el resultado que la eficiencia del sistema es del **87%**

#### **4.1.5. Mantenibilidad**

El mantenimiento se da de acuerdo a las modificaciones del sistema, esto quiere decir a los nuevos requerimientos y actualizaciones según los usuarios del sistema. Por lo cual el índice de madurez del software se determina de acuerdo a la siguiente ecuación.

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

Donde:

**Mt**: Número de módulos en la versión actual

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

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

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

En este sistema se tienen los siguientes valores de acuerdo a las variables ya mencionadas, se muestra la información en la siguiente tabla:

<b>Variable</b>	<b>Valor</b>
Mt	16
Fa	0

Fb	2
Fc	0

Realizamos los cálculos:

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

$$IMS = 0.875 * 100\%$$

$$IMS = 87\%$$

Por lo cual el sistema tiene un 87% de índice de madurez de software, que es la facilidad de mantenimiento, y el 13% corresponde al margen de error a los cambios y modificaciones.

#### **4.1.6. Portabilidad**

El presente proyecto de acuerdo a la metodología está diseñado en un entorno de acceso vía web, por lo cual la portabilidad por el lado del servidor y por el lado del cliente, entonces la portabilidad se medirá bajo los siguientes aspectos:

- Software del Servidor
- Hardware del Servidor
- Sistema Operativo del Servidor

En conclusión, de acuerdo a estas características el sistema es portable tanto en hardware como en software.

Para obtener la portabilidad, se tiene la siguiente formula:

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

Reemplazando a la fórmula se obtiene el siguiente resultado:

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

$$Portabilidad = 0.875 * 100$$

$$Portabilidad = 87.5\%$$

## Resultados:

De acuerdo a los resultados obtenidos se puede establecer la calidad total del sistema en base a parámetros medidos anteriormente.

**Tabla 22.**

*Resultados*

<b>Características</b>	<b>Resultados</b>
Funcionalidad	88.89%
Confiabilidad	85.46%
Usabilidad	89%
Eficiencia	87%
Mantenibilidad	87%
Portabilidad	87.5%
<b>Evaluación total</b>	<b>87.48%</b>

*Nota.* Evaluación de calidad total del sistema es de un 87.48%.

### **4.1.7. Seguridad del software ISO - 27001**

Según la ISO 27001, se considera a la información como un activo valioso para la entidad, el estándar se baso en el Sistema de Gestión de Seguridad de la información (SGSI) que permite evaluar todo tipo de riesgos y amenazas y luego tratarlos sistemáticamente.

### **4.1.8. Seguridad Lógica**

Para la seguridad del sistema se aplicó en el sistema las siguientes precauciones

- **Copias de seguridad:** Se hacen copias automáticas de la información importante para asegurarse de que no se pierda en caso de algún problema.
- **Encriptación:** La encriptación de la información ayuda a protegerla de ser accedida por personas no autorizadas. En este caso, el sistema permite encriptar las rutas y las contraseñas de los usuarios con hash.
- **Autenticación:** El uso de contraseñas encriptadas y sesiones controladas ayuda a proteger la información de accesos no autorizados. En este caso, el sistema tiene cuatro niveles de usuario, cada uno con diferentes privilegios. Esto ayuda a garantizar que solo las personas autorizadas tengan acceso a la información confidencial.
- **Base de datos:** Se hacen copias de seguridad regularmente para tener una versión segura de los datos, en caso de que se pierdan o estropeen.
- **Código CAPTCHA:** Se usan códigos CAPTCHA para evitar que programas automáticos puedan acceder al sistema. Esto evita que robots o programas maliciosos ingresen.

La información en una institución es extremadamente valiosa. Por eso, es crucial protegerla. La conexión a la base de datos se establece y se cierra automáticamente para asegurarse de que la información esté segura en todo momento. Esto garantiza que nadie pueda acceder a los datos sin permiso y que estén protegidos cuando no se estén utilizando.

#### **4.1.9. Seguridad Física**

Se establecen las siguientes medidas en cuanto al acceso se refiere:

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

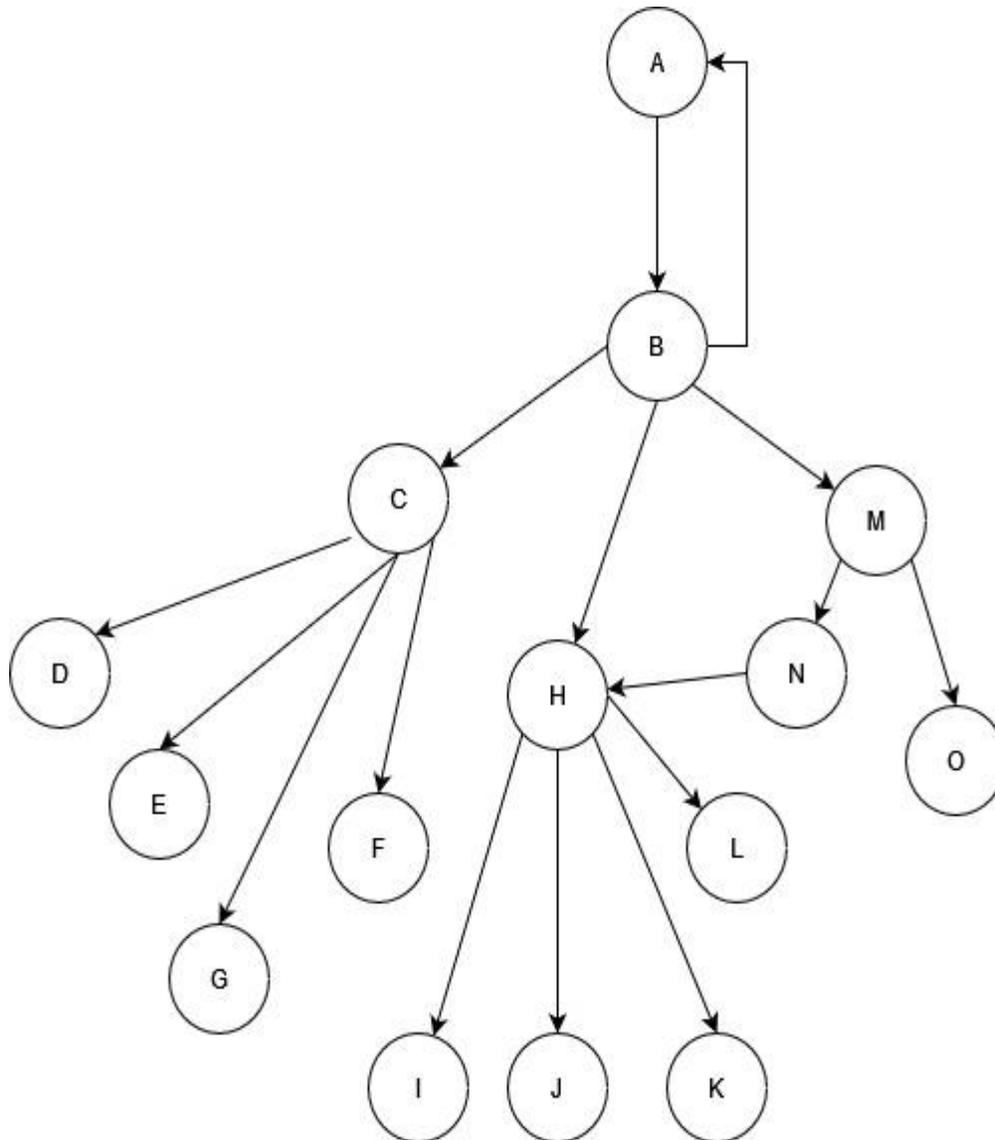
#### **4.2. PRUEBAS DE FUNCIONAMIENTO**

### 4.2.1. Prueba de caja blanca

Se vio necesario realizar pruebas de caja blanca para el módulo de recaudaciones, debido al flujograma de que caminos se debe seguir.

**Figura 66.**

*Flujograma de recaudaciones*



*Nota:* Elaboración propia.

Lista de nodos que formaran el grafo de flujo

- Inicio del sistema (A)

- Login usuario y contraseña (B)
- Módulo de sistemas (C)
- Administración de usuarios (D)
- Asignación de usuarios (E)
- Cambiar estado (F)
- Reset usuario y contraseña (G)
- Módulo de recaudaciones (H)
- Registro de persona Natural (I)
- Registro de persona Jurídica (J)
- Habilitar el servicio de instalación (K)
- Generar reporte (L)
- Módulo de caja (M)
- Búsqueda de persona (N)
- Registrar el cobro del servicio (O)

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

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

Donde:

A = Es el número de aristas (Lados)

N = Es el número de nodos (Vértices)

Por tanto:

$$V(G) = 16 - 15 + 2$$

$$V(G) = 3$$

El código tiene una complejidad ciclomática de 3, eso quiere decir que debemos realizar 2 pruebas para asegurarnos de que cada instrucción se ejecute una vez.

#### 4.2.2. Prueba de caja negra

Para realizar las pruebas de caja negra, se hace el uso de la técnica de partición equivalencia que es una de las más efectivas para analizar las interfaces del sistema.

**Tabla 23.**

*Caso de prueba - ingreso al sistema*

<b>Caso de prueba de ingreso al sistema</b>	
Código	PCN-01
Descripción de la prueba	Ingresar al sistema mediante usuario y contraseña, después de que se realizó la validación.
Condiciones de ejecución	El usuario debe estar primero registrado en el sistema y debe estar activo
Pasos de ejecución	El sistema muestra un login de autenticación, donde se pide el usuario y contraseña para el ingreso correspondiente
Resultado esperado	El sistema valida los datos ingresados, si ese fuera el caso ingresa al sistema de acuerdo al rol y permiso asignado en caso contrario mostrara un mensaje de 'Usuario y contraseña inválidos'
Evaluación de prueba	Se realiza el ingreso al sistema con validación y verificación previa y mostrara un mensaje de se inició la sesión con éxito

*Nota:* Se describe el caso de prueba de ingreso al sistema.

**Tabla 24.***Caso de prueba - Administrar persona natural y jurídica*

<b>Caso de prueba Administrar persona natural y jurídica</b>	
Código	PCN-2
Descripción de la prueba	Para el administrador de recaudaciones tiene todos los privilegios del registro de las personas naturales y jurídicas
Condiciones de ejecución	El usuario debe estar autenticado en el sistema como administrador o responsable
Pasos de ejecución	El sistema muestra un formulario donde debe registrar primero a la persona natural y después a la persona jurídica dando como responsable a una persona natural
Resultado esperado	El sistema valida los datos ingresados de la persona natural y guarda en la base de datos
Evaluación de prueba	Se realizó el registro de nueva persona natural o jurídica

*Nota:* Elaboración propia.**Tabla 25.***Caso de prueba Administrar cobros*

<b>Caso de prueba Administrar cobros</b>	
Código	PCN-3
Descripción de la prueba	Para administrar la parte de cobro del servicio

---

Condiciones de ejecución	El usuario debe estar autenticado en el sistema como administrador, responsable, técnico o cajera
Pasos de ejecución	El sistema muestra un formulario donde debe ingresar el número de carnet de la persona para realizar el cobro adecuado
Resultado esperado	El sistema valida los datos ingresados de la persona y emite la factura
Evaluación de prueba	Se cobro del servicio con éxito

---

*Nota:* Elaboración propia.

#### **4.2.3. Pruebas de estrés**

Las pruebas de estrés consisten en probar los límites que un sistema pueda soportar. En este tipo de pruebas se suelen enviar más peticiones de las que el software podría atender normalmente para saber el comportamiento de la aplicación.

La herramienta se evalúa al rendimiento del sitio web **Geekflare**.

- Sitio web: control del proyecto\_10\_grado.net (3.147.76.244)
- Estado de alojamiento en Hostinger International Limited.
- Puntuación de desempeño con un 81%.
- Puntuación de mejores prácticas con un 86%.
- Carga del sitio web: 1.9 segundo.
- Primer byte en 90mseg.
- Primera pantalla con el contenido de 1.6 segundo.
- Tamaño de pagina 0.9 Megabyte.
- Tiempo total de bloqueo de 0 mseg.

## Auditorias de desempeño.

**Tabla 26.**

*Auditoria de desempeño.*

Elimina los recursos que bloquean el moderizado.	Ahorro potencial de 450 ms
Imágenes de tamaño adecuado.	Ahorro potencial de 16 KiB
Minimizar CSS.	Ahorro potencial de 14 KiB
Minimizar JavaScript.	Ahorro potencial de 35 KiB
Eliminar de CSS no utilizado.	Ahorro potencial de 30 KiB
Eliminar JavaScript no utilizado.	Ahorro potencial de 47 KiB
El tiempo de respuesta inicial del servidor fue corto.	El documento raíz tardó 80 ms.
Mantenga los recuerdos de solicitudes bajos y los tamaños de transferencia pequeños.	79 solicitudes * 793 KiB
Sirva activos estáticos con una política de cache eficiente.	55 recursos encontrados
Evita grandes cambios de diseño.	2 elementos encontrados.
Evita encadenar solicitudes críticas.	70 cadenas encontradas.
Tiempo de ejecución de JavaScript.	0,0 s
Minimiza el trabajo del hilo principal.	0,3 s

*Nota:* Elaboración propia.

# **CAPÍTULO V**

## **ESTIMACIÓN DE COSTO**

## 5. ESTIMACIÓN DE COSTO

### 5.1. COCOMO II

Para el desarrollo de la estimación de costos COCOMO II, la estimación del sistema ha sido desarrollado bajo las líneas de código "LDC" (líneas de código), con este dato obtenemos "KL" (cantidad de líneas de código en miles).

$$LCD = 4595$$

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

$$KLCD = \frac{4595}{1000}$$

$$KLCD = 4.595$$

**Tabla 27.**

*Coefficientes de a, b, c, d de COCOMO II*

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

Verificamos que el proyecto no supera las 50000 líneas de código se utilizaran los coeficientes para proyecto orgánicos.

**Tabla 28.**

*Ecuaciones de Cocomo II*

Variable	Ecuación	Tipo / Unidad
Esfuerzo requerido por el proyecto	$E = a * (KLDC)^b * FAE$	Persona/mes

Tiempo requerido por el proyecto	$T = c * (E)^d$	Mes
Número de personas requeridos para el proyecto	$NP = \frac{E}{P}$	Personas
Costo Total	$CT = SueldoMes * NP * T$	\$us

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

**Tabla 29.**

*Cálculo de Atributos FAE*

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

Restricciones de tiempo de desarrollo	1,22	1,08	<b>1,00</b>	1,04	1,10
---------------------------------------	------	------	-------------	------	------

$$FAE = 1.15 * 1 * 1.15 * 1 * 1.06 * 1 * 1 * 0.71 * 0.91 * 0.86 * 1 * 0.95 * 0.91 * 0.91 * 1$$

$$= 0.6127$$

Reemplazando valores a la fórmula de esfuerzo, se tiene:

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

$$E = 3.2 * 4.595^{1.05} * 0.6127(\text{persona/mes})$$

$$E = 9.72(\text{persona/mes})$$

Cálculo del tiempo

$$T = c * Esfuerzo^d(\text{meses})$$

$$T = 2.5 * 9.72^{0.38}(\text{meses})$$

$$T = 5.93(\text{meses}) \text{ esto equivale a 6 meses}$$

Calculando el personal requerido

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

$$NP = \frac{9.72}{5.93} \text{ personas}$$

$$NP = 1.63 \cong 2 \text{ personas}$$

Calculando el costo total

$$CT = SueldoMes * NP * T$$

$$CT = 350 * 2 * 6$$

$$CT = 4200 \$us$$

En resumen, se requirió 2 personas en un tiempo de 6 meses y con un costo total de 4200 \$us, 6.91 Equivalente en bolivianos a 29022 Bs

**Tabla 30.**

*Resumen de costo*

Esfuerzo	9 personas
Tiempo	5 meses
Cantidad de personas por mes	2 personas
Costo total	29022Bs

*Nota.* El costo total estimado es de **29022 Bs.**

**CAPÍTULO VI**  
**CONCLUSIONES Y**  
**RECOMENDACIONES**

## **6. CONCLUSIONES Y RECOMENDACIONES**

### **6.1. CONCLUSIONES**

El objetivo general del proyecto se logró satisfactoriamente con el "Sistema Web para la Administración de Información y Control de la Recaudación de Recursos del Suministro de Agua Potable" destinado a la Unidad de Dirección de Recaudaciones del Gobierno Autónomo Municipal de Chulumani.

- Se ha completado un análisis exhaustivo en los procesos del funcionamiento y requisitos del sistema, identificando los flujos de trabajo, roles y procesos fundamentales necesarios para el funcionamiento del sistema.
- Se ha desarrollado con éxito una Base de Datos centralizada que cumple con los requisitos de almacenamiento de datos de la Institución, permitiendo una gestión eficiente y organizada de la información.
- El sistema de control del servicio de agua potable ha sido diseñado en consonancia con las necesidades esenciales y los requisitos identificados durante el diagnóstico del flujo de trabajo, asegurando así la alineación con los estándares necesarios.
- Se ha implementado un reporte dinámico que mejorará la atención al cliente al proporcionar la información correcta, la cual se encuentra almacenada en una base de datos centralizada y gestionada de manera eficiente.
- Las pruebas realizadas han confirmado que el sistema cumple con las expectativas deseadas, validando su funcionalidad y confirmando la capacidad de satisfacer las necesidades identificadas durante el proceso de desarrollo del sistema.

De esta manera se tiene que el software cumple y ayuda en los procesos de administración de información, como también en los procesos de control y pago del suministro agua potable, sea eficiente al proporcionar datos centralizados al usuario para así tener un buen control del servicio vital en el Gobierno Autónomo Municipal de Chulumani, lo cual se logró optimizar los procesos de ejecución en un tiempo mínimo.

## 6.2. RECOMENDACIONES

El presente proyecto se proponen las siguientes recomendaciones, con el fin de buscar el mejoramiento del sistema.

- Realizar la administración continua y periódica del sistema web para poder atender oportunamente el control cobro del servicio
- Se recomienda cambiar continuamente las contraseñas para la seguridad del sistema y proteger el acceso a personas ajenas.
- Considerar que la información del sistema sea manipulada únicamente por el administrador o técnicos y mas no por el cajero.
- La revisión periódica es recomendable para la eficiencia de un funcionamiento adecuado del sistema.
- Se recomienda trabajar con lenguajes de programación compatibles con el Framework Laravel, para tener las versiones actualizadas a futuro.

## 7. BIBLIOGRAFÍA

- DeerDev . (s.f). *Modelo cliente servidor*. Obtenido de Conoce a detalle el la utilidad de las aplicaciones cliente-servidor: <https://deer.dev/blog/cliente-servidor>
- Perez Porto, J. (2012). *Ciencias de la Informacion*. Obtenido de <http://csdelainformacionvzla.blogspot.com/2017/10/valoraciones-fundamentales-para.html>
- Robbins, S. (2012). *Fundamentos de administracion financiera*. Obtenido de Red Tercer Milenio: <http://biblioteca.udgvirtual.udg.mx/>
- Salim, J. (septiembre de 2009). *PROYECCIÓN, ACOMPAÑAMIENTO Y ANÁLISIS DEL COMPORTAMIENTO DE LA RECAUDACIÓN TRIBUTARIA*. Obtenido de Recaudacion: [https://www.ciat.org/Biblioteca/SerieTematica/Espanol/2009\\_sept\\_N4\\_recaudacion.pdf](https://www.ciat.org/Biblioteca/SerieTematica/Espanol/2009_sept_N4_recaudacion.pdf)
- , Stoner J.; , Freeman R. (1995). *DEFINICIÓN DE ADMINISTRACIÓN*.
- Agarwal. (2022). *Universidad Nacional del Este Facultad Politécnica*. Obtenido de proceso de estimacion.
- Allen Collier, D., & Evans , J. (2019). *Definición de cadena de suministro*. Obtenido de <https://www.emprendedorinteligente.com/que-es-la-cadena-de-suministro-segun-autores/>
- Anonimo. (2012). *UWE - UML-based WEB Engineering*. Obtenido de Ingenieria WEB: <https://sites.google.com/site/ingenieriawebuc/home/contenido/uwe>
- Anonimo. (s.f.). *Estadísticas sobre el Suministro y Saneamiento del Agua. Metodología*. Obtenido de Medio ambiente y desarrollo sostenible: <https://www.ine.es/DEFIne/es/concepto.htm?c=4756&op=30087&p=2&n=20>
- Apaza Bilbao, M. (2021). *SISTEMA PRESUPUESTARIO DE INFORMACIÓN PARA PROYECTOS DE CONSTRUCCIÓN. CASO: EMPRESA MULTIDISCIPLINARIA ADUS.R.L*. Obtenido de PROYECTO DE GRADO: <https://repositorio.umsa.bo/bitstream/handle/123456789/29647/PG-3820.pdf?sequence=1&isAllowed=y>
- Araujo, F., & Lorena, O. (2021). *Desarrollo de una nueva metodología derivada del COCOMO II agregando pautas de calidad*. Obtenido de Maestria en informatica y computacion.
- Arias Estupiñán, J. D. (2022). *Diseño e implementación back-end de un software libre para el monitoreo y análisis de información en bases de datos enfocado a contribuir con el crecimiento de pequeñas y medianas empresas*. Obtenido de (Doctoral dissertation, Universidad Santo Toma).
- Arias, M. A. (2013). *Introducción a PHP*. IT Campus Academy. Campus Academy.

- Bachelor's thesis y Quito, B. C. (2015, 2014). *Evaluación de calidad de productos de software en empresas de desarrollo de software aplicando la norma ISO/IEC 25000*. Obtenido de <https://bibdigital.epn.edu.ec/handle/15000/9113>
- Barners. (1990). *la World Wide Web (WWW)*. Obtenido de <http://virtual.urbe.edu/>
- Betina. (22 de 03 de 2013). *Diferencia entre Cliente y Servidor*. Obtenido de Diferencia entre Cliente y Servidor: <http://diferenciaentre.net/diferencia-entre-cliente-y-servidor/>
- Blogeer Cruz. (01 de 08 de 2017). *evaluacion de software*. Obtenido de Norma ISO/IEC 25000: <http://cursoevaluaciondesoftwareunad.blogspot.com/2017/08/norma-isoiec-25000.html>
- Boté, J y Juanjo Boté. (2013). *Aprende HTML efectivo*:. Obtenido de Conceptos básicos para crear páginas web: [https://books.google.es/books?hl=es&lr=&id=CcoRL\\_3mRI8C&oi=fnd&pg=PT6&dq=html+concepto&ots=zx0NErsAmY&sig=boOm46FluFb1LBQKCBdt03ZG23Y#v=onepage&q=html%20concepto&f=false](https://books.google.es/books?hl=es&lr=&id=CcoRL_3mRI8C&oi=fnd&pg=PT6&dq=html+concepto&ots=zx0NErsAmY&sig=boOm46FluFb1LBQKCBdt03ZG23Y#v=onepage&q=html%20concepto&f=false)
- Broehm , B. (2000). *Modelo de Arquitectura COCOMO II*. Obtenido de Cocomo II.
- Burgos, Franklin E.;. (2012). *TRABAJO ESPECIAL DE GRADO DEFINICIÓN DE UN PLAN DE GESTIÓN DE LA CALIDAD, BASADO EN EL MODELO ISO/IEC (9126)*. Obtenido de PARA PROYECTOS DE DESARROLLO DE SOFTWARE: <http://biblioteca2.ucab.edu.ve/anexos/biblioteca/marc/texto/AAS3851.pdf>
- Cabello, J. C. (2010). *Diseño de páginas Web con XHTML, JavaScript y CSS*. Ra-Ma Editorial. RedUsers.
- Caisa Guayta, C. J., & Semblantes Chicaiza, L. V. (octubre y noviembre de 2008 y 2009). *Implementación de Pruebas Caja Negra y Caja Blanca aplicables al Sistema Escolástico del Colegio Nacional "Primero de Abril" de la Ciudad de Latacunga Provincia de Cotopaxi*". Obtenido de en el periodo Octubre 2: <http://repositorio.utc.edu.ec/handle/27000/1166>
- Caisa Guayta, C. J., & Semblantes, C. (2 de Octubre de 2011). *Implementación de Pruebas Caja Negra y Caja Blanca aplicables al Sistema Escolástico del Colegio Nacional "Primero de Abril" de la Ciudad de Latacunga Provincia de Cotopaxi*". Obtenido de REVISTA COLOMBIANA DE COMPUTACIÓN: <https://revistas.unab.edu.co/index.php/rcc/article/view/1833/1662>
- Chavez Arias, M. (2006). *Ingeniería de requerimientos*.
- Chavez, M. A. (2005). La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software. *InterSedes: Revista de las Sedes Regionales*, 1 a 15.
- Chen, C. (2019). *Qué es Sistema de información*. Obtenido de <https://www.studocu.com/latam/document/universidad-federada-san-judas-tadeo/informatica/sistemas-de-informacion-principales/14619940>

- Chiavenato. (2006). *Administración de Mercadotecnia*.
- Chiavenato, I. (1999). *ADMINISTRACIÓN DE RECURSOS HUMANOS*". Editorial Mc Graw Hill.
- Clavadetscher, C. (31 de 08 de 2022). *Autorización en PostgreSQL*. Obtenido de Documento de apoyo para cursos.
- Cruz Valle, Jorge ;. (2005). *Ingeniería de requerimientos*. Obtenido de Academia.edu:  
[https://www.academia.edu/17176029/Ingenieria\\_de\\_requerimientos](https://www.academia.edu/17176029/Ingenieria_de_requerimientos)
- Cubero, H. (2020). Lenguaje unificado de modelado (UML). *Miembro de la red Ilumino*, 3.
- Domiinguez Betancourt, J. R. (09 de septiembre de 2011). *DIFERENTES CARACTERÍSTICAS DE LA ARQUITECTURA CLIENTE-SERVIDOR*. Obtenido de Arquitectura cliente-servidor:  
[http://juanitocuirindichapio.blogspot.com/2011/09/diferentes-caracteristicas-de-la\\_09.html](http://juanitocuirindichapio.blogspot.com/2011/09/diferentes-caracteristicas-de-la_09.html)
- García Hoyos, J. L. (2019). *Migración del software kactus-hcm de una arquitectura cliente-servidor a una arquitectura cliente-contenedor*. Obtenido de  
[http://repositoriodspace.unipamplona.edu.co/jspui/bitstream/20.500.12744/656/1/Garc%c3%ada\\_2019\\_TG.pdf](http://repositoriodspace.unipamplona.edu.co/jspui/bitstream/20.500.12744/656/1/Garc%c3%ada_2019_TG.pdf)
- Gomez Castro, J. A. (20 de agosto de 2018). *Metodologías de programación MODELO ITERATIVO INCREMENTAL*. Obtenido de Ingeniería de Software — Vol I:  
<https://josegomezdev.medium.com/proceso-estructurado-afb0ea9bcf8a>
- Gómez, A; López, M; Migan, S; Otazu, A, . (2010). *Un Modelo de Estimación de proyectos de Software*. Obtenido de Estimacion del esfuerzo.
- Gómez, Mario Rubiales. (2022). *CURSO DE DESARROLLO WEB*:. Obtenido de HTML, CSS Y JAVASCRIPT.
- Guevara, A. (2007). *Ingeniería Web: e-Learning*. Obtenido de Cultura Científica:  
[http://sedici.unlp.edu.ar/bitstream/handle/10915/4172/Documento\\_completo\\_.pdf?sequence=1&isAllowed=y](http://sedici.unlp.edu.ar/bitstream/handle/10915/4172/Documento_completo_.pdf?sequence=1&isAllowed=y)
- Guillermina, B. (2003). *La Recaudación*. Obtenido de  
[http://catarina.udlap.mx/u\\_dl\\_a/tales/documentos/ledf/lievano\\_m\\_o/capitulo2.pdf](http://catarina.udlap.mx/u_dl_a/tales/documentos/ledf/lievano_m_o/capitulo2.pdf)
- información, ISO/IEC 27001 Seguridad de la. (s.f.). *Asesoría ISO RCR*. Obtenido de [https://asesorias-iso.cl/iso-iec-27001-seguridad-de-la-informacion/?gclid=Cj0KCQjwhY-aBhCUARIsALNIC05hplOWnxFgSv4zgXfmfY63qepu3tQEX1Mz3ZmNx5MUxO6TMVbm8zAaApNqEALw\\_wcB](https://asesorias-iso.cl/iso-iec-27001-seguridad-de-la-informacion/?gclid=Cj0KCQjwhY-aBhCUARIsALNIC05hplOWnxFgSv4zgXfmfY63qepu3tQEX1Mz3ZmNx5MUxO6TMVbm8zAaApNqEALw_wcB)
- Iturralde, O. J. (s.f.). *Arquitectura Cliente-Servidor*. En O. J. Iturralde, *Introducción a la arquitectura de software*. primera edición. Obtenido de Arquitectura Cliente-Servidor:  
<https://reactiveprogramming.io/blog/es/estilos-arquitectonicos/cliente-servidor>

- Koontz, H., & Weihrich, H. (2013). Un enfoque internacional. En K. H. Heinz, *Elementos de la Administracion* (pág. 484). Mexico: OCTAVA EDICION.
- Koontz, H., Weihrich, H., & Donnell, C. (1991). *Elementos de administración*. Quinta Edicion.
- Kuhn, T. (s.f.).
- Largo Garcia, C., & Marin Mazo, E. (2003). *Aplicacion del modelo ISO 9126*. Obtenido de Guia tecnica para evaluacion de software: <https://es.slideshare.net/abetancur/guia-tecnica-para-evaluacin-de-software>
- Lionsty, A. (2016). *Definición de Cadena de suministro*. Obtenido de segun autores.
- Lockhart, T. (2022). *Tutorial de PostgreSQL*. Obtenido de El equipo de desarrollo de PostgreSQL, 10.
- López, F. R. (s.f). *INGENIERÍA DEL SOFTWARE I*. Obtenido de tema 8: <https://www.ctr.unican.es/asignaturas/is1/is1-t08-trans.pdf>
- Mantilla Guerra, A. (29 de diciembre de 2017). Gestión de seguridad de la información con la norma ISO 27001:2013. *Espacios*, 7. Obtenido de <https://www.revistaespacios.com/a18v39n18/a18v39n18p05.pdf>
- Marcos, J., & Arroyo, A. y. (2008). La norma ISO/IEC 25000 y el proyecto KEMIS para su automatización con software libre. *Revista Española de Innovación, Calidad e Ingeniería del Software*, 133-144.
- Martínez, A. (2022). *Definición de Sistema*.
- Mesquida, L. A. (2010). istema de Gestión Integrado según las normas ISO 9001, ISO/IEC 20000 e ISO/IEC 27001. *REICIS. Revista Española de Innovación, Calidad e Ingeniería del Software*, 25 y 34.
- Moran, L. (2001).
- Murugesan. (2022). *Introducción a la Ingeniería Web*. Obtenido de Procesos y Métodos de Modelado para: <https://repositorio.grial.eu/bitstream/grial/2533/1/Tema1.pdf>
- Perez, J., & Gardey, A. (2012). *CIENCIAS DE LA INFORMACIÓN*. Obtenido de informacion: <http://csdelainformacionvzla.blogspot.com/2017/10/valoraciones-fundamentales-para.html>
- Periañez, J. y. (2004). *La Administracion*.
- Pincolini, F. J. (2012). *SUMINISTRO DE AGUA POTABLE*. Obtenido de Trabajo de Investigación: [https://bdigital.uncu.edu.ar/objetos\\_digitales/5158/pincolinisuministroaguapotable.pdf](https://bdigital.uncu.edu.ar/objetos_digitales/5158/pincolinisuministroaguapotable.pdf)
- Pressman. (2006). *Ingenieria de Requerimientos*.

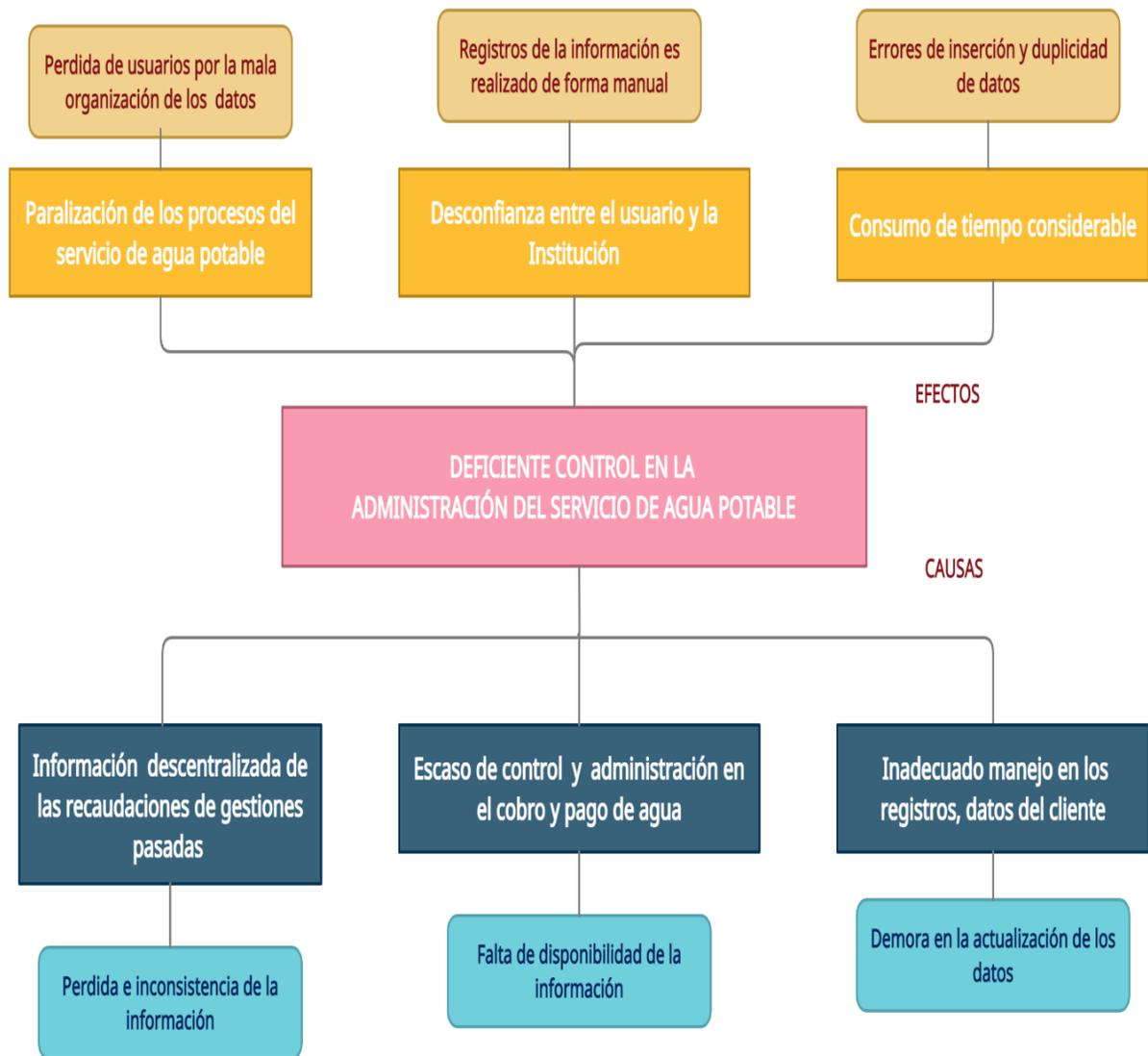
- Pressman, R. S. (2005). *Ingeniería de Software*. Obtenido de Modelos de Desarrollo de Software: <http://isw-udistrital.blogspot.com/>
- Richos Organization. (2018). *Laravel*.
- Roa , P., Morales, C., & Gutiérrez , P. (28 de 09 de 2015). *Norma iso/iec 25000*. Obtenido de Tecnología Investigación y Academia: <https://geox.udistrital.edu.co/index.php/tia/article/view/8373/11349>
- Robbins. (1996). *PROCESO DE FABRICACIÓN*. Obtenido de El control como fase del proceso administrativo: [http://www.juntadeandalucia.es/averroes/centros-tic/14002984/helvia/aula/archivos/repositorio/1500/1663/html/web/jose\\_pablo/frabricacion.htm](http://www.juntadeandalucia.es/averroes/centros-tic/14002984/helvia/aula/archivos/repositorio/1500/1663/html/web/jose_pablo/frabricacion.htm)
- Roberto Caroca. (21 de 08 de 2020). *Frontend y Backend*:. Obtenido de La analogía con un restaurante.: <https://es.linkedin.com/pulse/frontend-y-backend-analog%C3%ADa-del-restaurante-roberto-caroca>
- Robles , R. (2012). *Fundamentos de administración financiera*.
- Rojas, J. E. (2023). *Diseño de casos de prueba, pruebas de caja blanca y del camino básico*. Obtenido de Ingeniería de Software: <https://ingsoftwarejefer.webcindario.com/unidad-3/tecnicas-de-prueba-del-software/diseño-de-casos-de-prueba-pruebas-de-la-caja-blanca-y-del-camino-basico.html>
- Rojas, M., & García, J. (03 de 2021). *Introducción y principios básicos del desarrollo de software basado en componentes*. Obtenido de Colombia2004. ISBN. DOI [Citado.
- Rojo, S. D. ((2012).). *Requerimientos No funcionales para aplicaciones Web (Doctoral dissertation, Universidad Nacional de La Plata)*. Obtenido de Tesis Doctoral: <http://sedici.unlp.edu.ar/handle/10915/33033>
- Rotta, D., Pallott, G., Klikailo, H., & Belloni, E. (2016). *Un caso de estudio sobre la aplicación de UWE para la generación de Sistemas Web*. Obtenido de Concurso de Trabajos Estudiantiles: [http://sedici.unlp.edu.ar/bitstream/handle/10915/58143/Documento\\_completo.pdf-PDFA.pdf?sequence=1&isAllowed=y](http://sedici.unlp.edu.ar/bitstream/handle/10915/58143/Documento_completo.pdf-PDFA.pdf?sequence=1&isAllowed=y)
- Sánchez Peño, J. M. (2015). *Pruebas de Software, fundamentos y técnicas*. Madrid: Universidad Politécnica de Madrid.
- Scanlan. ( 1987). *sistema de control*. Obtenido de [https://www.ecotec.edu.ec/material/material\\_2019D1\\_ADM106\\_012\\_122713.pptx](https://www.ecotec.edu.ec/material/material_2019D1_ADM106_012_122713.pptx)
- Schiaffarino, Andrés. (12 de marzo de 2019). *Modelo cliente servidor*. Obtenido de infranetworking: <https://blog.infranetworking.com/modelo-cliente-servidor/>

- Silvana del Valle, R. (abril de 2012). *REQUERIMIENTOS NO FUNCIONALES PARA APLICACIONES WEB*. Obtenido de Trabajo Integrador presentado para obtener el grado de especialista en Ingeniería de Software”.
- Solano Fernández, E., & Porras Alfaro, D. (2020). *El modelo iterativo e incremental para el desarrollo de la aplicación de realidad aumentada Amón\_RA*. Obtenido de Artículo científico: [https://revistas.tec.ac.cr/index.php/tec\\_marcha/article/view/5518](https://revistas.tec.ac.cr/index.php/tec_marcha/article/view/5518)
- Sommerville . (2005). *Que es un sistema de informacion*. Obtenido de <https://www.coursehero.com>
- Suárez. ( 2009). *Proceso Administrativo*. Obtenido de <https://es.slideshare.net/TeGaVe/proceso-administrativo-27532222>
- Tom MCCabe. (s.f.). *Pruebas de Software*. Obtenido de Prueba de camino basico: <https://coggle.it/diagram/X6gU6RIMvfQ7FndU/t/pruebas-de-software>
- Unknown. (2016). *¿Qué es la ISO?* Obtenido de ISO9126: [http://iso9126uts.blogspot.com/p/i\\_8.html](http://iso9126uts.blogspot.com/p/i_8.html)
- Valero Duran, J. D. (2013). *Seguimiento y Evaluación para un Software de Administración para un Sistema de Gestión de Seguridad de la Información Basado en la Norma ISO 27001*. Obtenido de PROYECTO DE GRADO PARA OPTAR POR EL TÍTULO DE ESPECIALISTA EN: <https://repository.udistrital.edu.co/bitstream/handle/11349/4786/valeroduranjuandaniel2015.pdf?sequence=1&isAllowed=y>
- Varas y Pradenas, J. (2000). Hacia la definición de métricas de calidad para esquemas conceptuales de bases de datos. *Revista Electrónica del DIICC, edición (6)*.
- Zambrano, P., Vélez Vélez , L., & Almeida Zambrano, E. (2020). Desarrollo e implementación de aplicación móvil para la difusión de publicaciones de la Editorial ULEAM. *Revista Científica INGENIAR: Ingeniería, Tecnología e Investigación. ISSN: 2737-6249.*, 23. Obtenido de Artículo de investigación. Revista Científica INGENIAR: Ingeniería, Tecnología e Investigación.
- Zapata, C., Cardona Velásquez, M., & De Jesús, C. (2021). *Comparación de las características de algunas herramientas de software para pruebas de carga. 8(2)*. Obtenido de Revista Avances en Sistemas e Informática: <https://www.redalyc.org/pdf/1331/133119867014.pdf>

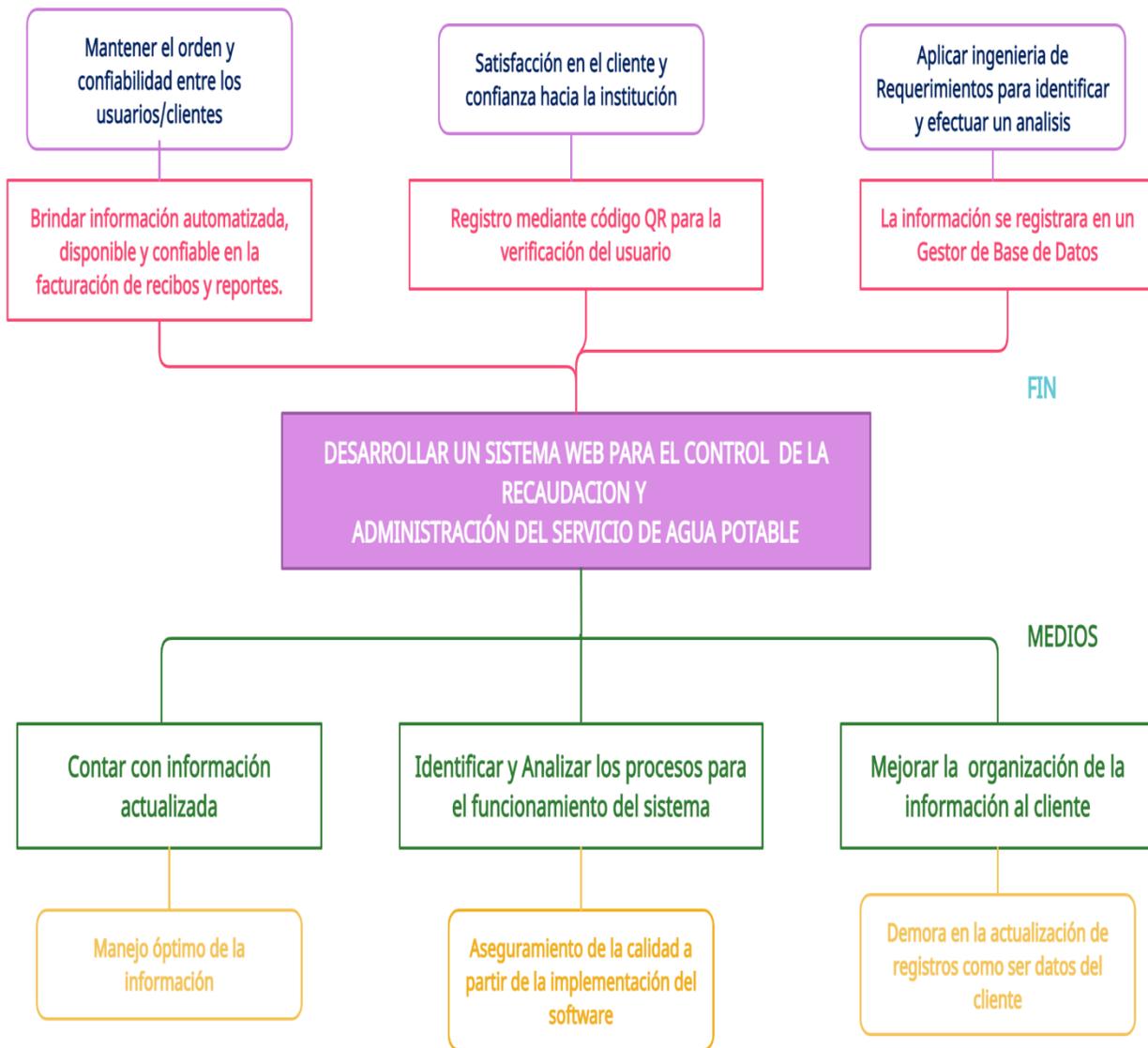
# **ANEXOS**

# **ANEXO A**

# ÁRBOL DE PROBLEMAS



# ÁRBOL DE OBJETIVOS



# **ANEXO B**

# Gobierno Autónomo Municipal de Chulumani

Villa de la Libertad

Capital de la Provincia Sud Yungas - La Paz - Bolivia

Chulumani, 24 de noviembre 2023

Señor:

Lic. Ing. Dionicio Henry Pacheco Rios

**TUTOR METODOLÓGICO**

**TALLER DE GRADO II**

Presente. -

## REF. AVAL DE CONFORMIDAD

Distinguido tutor metodológico:

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

**TITULO:** SISTEMA WEB PARA LA ADMINISTRACIÓN DE INFORMACIÓN Y CONTROL DE LA RECAUDACIÓN DE RECURSOS DEL SUMINISTRO DE AGUA POTABLE.

**CASO:** GOBIERNO AUTÓNOMO MUNICIPAL DE CHULUMANI

**MODALIDAD:** PROYECTO DE GRADO

**Univ.:** NOEMI LIZ SOLAREZ CHICO

**Registro Universitario:** 200017206

**Cedula de identidad:** 8431010 Lp.

De tal forma cabe recalcar que el **SISTEMA/PROYECTO** satisface los requerimientos de la institución de esta forma se dio cumplimiento de los objetivos del presente.

El presente **SISTEMA/PROYECTO** fue **IMPLEMENTADO** satisfactoriamente y se realizó las capacitaciones necesarias en la institución.

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

Atentamente,



*[Handwritten signature]*  
Ramon Viana Avendaño  
Ing. en Sistemas  
C.A. U.P.E.A.  
Calle 10 de Mayo - La Paz

Villa de la Libertad, Ing. Calle 10 de Mayo, No. 1000  
Municipio Chulumani  
Tel. 3040711 - 3040712  
Email: info@chulumani.gob.bo

*Todos somos Chulumani...*

Señor:  
Lic. Ing. William Roque Roque  
**DIRECTOR DE CARRERA  
INGENIERÍA DE SISTEMAS**  
Presente. –

El Alto, 27 de noviembre de 2023

**REF. AVAL DE CONFORMIDAD**

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

**TITULO: SISTEMA WEB PARA LA ADMINISTRACIÓN DE INFORMACIÓN Y CONTROL DE LA RECAUDACIÓN DE RECURSOS DEL SUMINISTRO DE AGUA POTABLE.**

**CASO: GOBIERNO AUTÓNOMO MUNICIPAL DE CHULUMANI**

**MODALIDAD: PROYECTO DE GRADO**

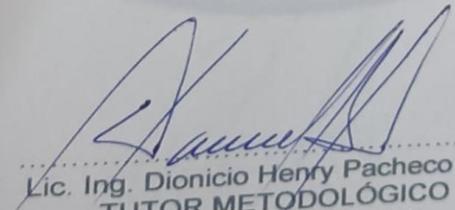
**Univ. NOEMI LIZ SOLAREZ CHICO**

**Registro Universitario: 200017206**

**Cedula de Identidad: 8431010 Lp.**

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

Atentamente,

  
.....  
Lic. Ing. Dionicio Henry Pacheco Rios  
**TUTOR METODOLÓGICO  
TALLER DE GRADO II**

Señor:  
Lic. Ing. Dionicio Henry Pacheco Rios  
**TUTOR METODOLÓGICO**  
**TALLER DE GRADO II**  
Presente. —

El Alto, 27 de noviembre de 2023

**REF. AVAL DE CONFORMIDAD**

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

**TITULO:** SISTEMA WEB PARA LA ADMINISTRACIÓN DE INFORMACIÓN Y CONTROL DE LA RECAUDACIÓN DE RECURSOS DEL SUMINISTRO DE AGUA POTABLE.

**CASO:** GOBIERNO AUTÓNOMO MUNICIPAL DE CHULUMANI

**MODALIDAD:** PROYECTO DE GRADO

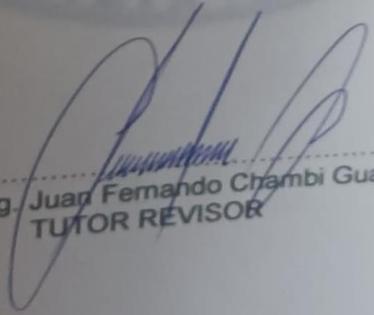
Univ. NOEMI LIZ SOLAREZ CHICO

Registro Universitario: 200017206

Cedula de Identidad: 8431010 Lp.

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

Atentamente,

  
M. Sc. Lic. Ing. Juan Fernando Chambi Guachalla  
**TUTOR REVISOR**

Señor:  
Lic. Ing. Dionicio Henry Pacheco Rios  
**TUTOR METODOLÓGICO**  
**TALLER DE GRADO II**  
Presente. –

El Alto, 27 de noviembre de 2023

**REF. AVAL DE CONFORMIDAD**

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

**TITULO:** SISTEMA WEB PARA LA ADMINISTRACIÓN DE INFORMACIÓN Y CONTROL DE LA RECAUDACIÓN DE RECURSOS DEL SUMINISTRO DE AGUA POTABLE.

**CASO:** GOBIERNO AUTÓNOMO MUNICIPAL DE CHULUMANI

**MODALIDAD:** PROYECTO DE GRADO

Univ. NOEMI LIZ SOLAREZ CHICO

**Registro Universitario:** 200017206

**Cedula de Identidad:** 8431010 Lp.

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

Atentamente,



Ing. Rolando Ulo Mollo  
**TUTOR ESPECIALISTA**

# **ANEXO C**



# 2023

# MANUAL DE

# USUARIO

---

**GOBIERNO AUTÓNOMO MUNICIPAL DE**  
**CHULUMANI**





## SISTEMA WEB DE ADMINISTRACION Y CONTROL DE LA RECAUDACION DEL SUMINISTRO DE AGUA POTABLE

El sistema fue desarrollado pensando en la constante demanda de servicios informáticos, mediante el módulo de Control de Ingreso podrá tener una mejor atención, seguimiento y estabilidad respecto a las visitas al ingresar al módulo.

### LOGIN DEL SISTEMA

- 1.- para ingresar el nombre de usuario con lo que se registró en el sistema.
- 2.- Ingrese la contraseña con lo que se registró en el sistema.
- 3.- Ingresar el captcha de seguridad generado en el sistema.

Una vez ingresado los datos correctos se le presentaran los siguientes módulos.

The image shows a web login form with the following fields and elements:

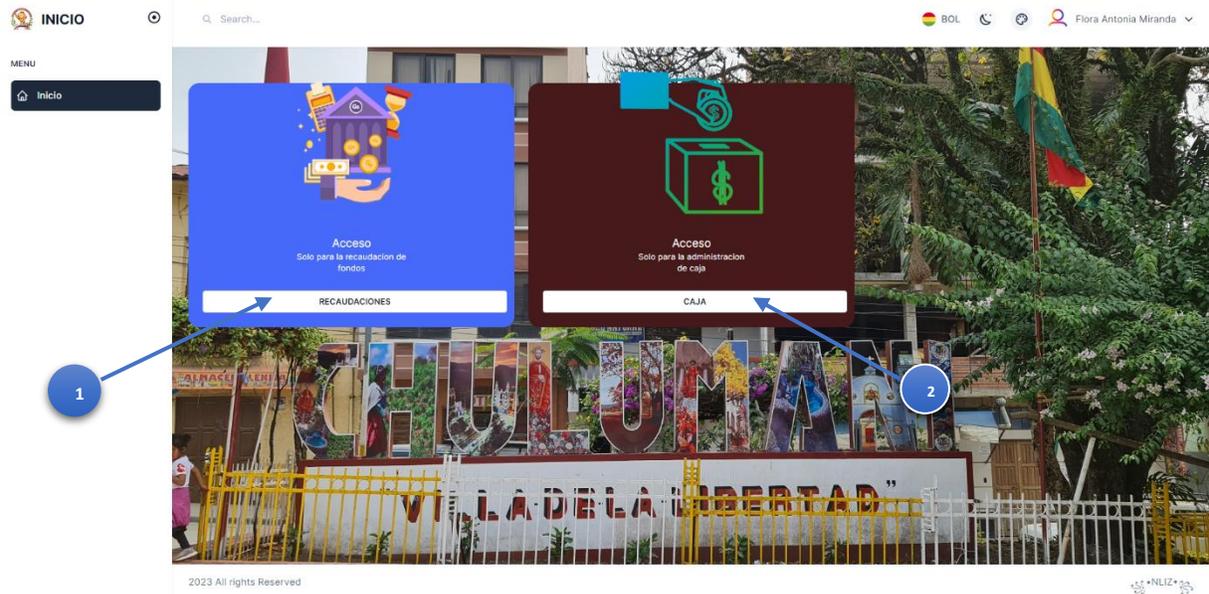
- Usuario:** A text input field with the placeholder "Ingrese su usuario".
- Contraseña:** A text input field with the placeholder "Ingrese su contraseña".
- Capcha:** A text input field with the placeholder "Ingrese capcha".
- INGRESAR:** A dark button at the bottom of the form.
- Captcha:** A small image showing the characters "4E2qz" and a red square icon.

Three blue arrows with numbers 1, 2, and 3 point to the Usuario, Contraseña, and Capcha fields respectively.





GOBIERNO AUTÓNOMO MUNICIPAL DE CHULUMANI  
UNIDAD RECAUDACIONES



1.- Modulo de la unidad de recaudaciones, donde se realizará nuevas afiliaciones, del servicio del suministro de agua potable.

2.- Modulo acceso a la caja, donde se realizará los cobros y pagos una vez pasado en el proceso de afiliación de los nuevos usuarios.

Nº	NOMBRES Y APELLIDOS	CI	Nº CELULAR	DIRECCIÓN	ACCIONES
1	HUEGO, AYLLON	8956237 LP	78956230	avenida circunvalacion	[Vista detallada] [editar] [eliminar]
2	ANTONIO BURGOA CHULUMANI	89562341 LP	78945628	zona central	[Vista detallada] [editar] [eliminar]
3	MARCELO RENZO CHULUMANI	3605484 CBB	78945612	avenida circunvalacion	[Vista detallada] [editar] [eliminar]
4	JHONNY RICAR CHACAU ARCADI	6025089 LP	78524610	avenida circunvalacion	[Vista detallada] [editar] [eliminar]
5	JUAN CARLOS ARAMAYO MERIDA	4797431 LP	75623014	avenida circunvalacion	[Vista detallada] [editar] [eliminar]
6	ISAAC ANTON GORENA ESPINOZA	2789126 LP	78945613	avenida circunvalacion	[Vista detallada] [editar] [eliminar]
7	JAIME TELLERIA VISCARRA	9685321 LP	78922554	avenida circunvalacion	[Vista detallada] [editar] [eliminar]
8	MOISES CRUZ QUENALLATA	69325001 LP	68943256	avenida circunvalacion	[Vista detallada] [editar] [eliminar]
9	JUANA KAPA	99456230 LP	78920113	avenida circunvalacion	[Vista detallada] [editar] [eliminar]
10	LUIS CALAMANI ALI	69305420 LP	78524106	calle alianza	[Vista detallada] [editar] [eliminar]

Una vez ingresado al modulo de recaudaciones tenemos las siguientes opciones en registro de nueva afiliación de usuarios tanto como persona natural y jurídica.





GOBIERNO AUTÓNOMO MUNICIPAL DE CHULUMANI  
UNIDAD RECAUDACIONES

A continuación, nos muestra un formulario para el registro de nuevo afiliado de una persona natural.

The screenshot shows a web form titled "Nuevo registro de persona Natural". It is divided into three main sections: "INFORMACIÓN PERSONAL", "CONTACTOS", and "DIRECCIÓN".

- INFORMACIÓN PERSONAL:** Includes fields for CI (Cédula de Identidad), Complemento, Expedido, Nombres, Apellido Paterno, Apellido Materno, Selección de género, Estado civil, and Selección de profesión.
- CONTACTOS:** Includes fields for Email, N° de celular, and N° de celular de referencia.
- DIRECCIÓN:** Includes a dropdown for "Selección de la zona", a field for "Dirección de domicilio", and a field for "Información adicional".

Callouts point to various elements: "Datos requeridos de la nueva afiliación." points to the CI field; "Agregar datos del afiliado lugar donde vive actualmente" points to the "Dirección" section; "Agregar los contactos del nuevo afiliado" points to the "CONTACTOS" section; "Click agregar" points to the "Guardar" button at the bottom right.

Seguidamente, tenemos los siguientes datos para el registro de una persona Jurídica.

The screenshot shows a web interface for "RECAUDACIONES" with a sidebar menu and a main content area. The main content area displays a "Listado de Persona jurídica" table. Callouts point to "agregar nuevo afiliado" (a button), "Vista detallada" (a button), and "editar" (a button).

Nº	NIT	NOMBRE EMPRESA	EMAIL	CELULAR	NÚMERO DE TESTIMONIO	ACCIONES
1	1020589022	Banco de Desarrollo Productivo Sociedad Anónima Mixta (BDP- S.A.M.)	bancodesarrollo@gmail.com	68953001	2311	[Vista detallada] [editar] [eliminar]
2	1028415020	Banco Unión S.A.	bancounion@gmail.com	69800		[Vista detallada] [editar] [eliminar]
3	1020601026	DIACONIA	diaconia@gmail.com	69822001	78	[Vista detallada] [editar] [eliminar]





GOBIERNO AUTÓNOMO MUNICIPAL DE CHULUMANI  
UNIDAD RECAUDACIONES

A continuación, nos muestra un formulario para el registro de nuevo afiliado de una persona jurídica.

The screenshot shows a web form titled "Nuevo registro de persona Jurídica". It is divided into three main sections: PERSONA, DOCUMENTOS, and INFORMACIÓN. The PERSONA section includes a dropdown for "SELECCIONE EL TIPO DE EMPRESA" and a text field for "REPRESENTANTE LEGAL". The DOCUMENTOS section includes a text field for "DESCRIPCIÓN DE LA ACTIVIDAD ECONOMICA", a text field for "NÚMERO DE TESTIMONIO", and a file upload area for "SELECCIONE EL TESTIMONIO". The INFORMACIÓN section includes text fields for "NIT", "NOMBRE DE LA EMPRESA", "N° TELEFONO", "N° DE CELULAR", "EMAIL", and "FECHA DE CONSTITUCIÓN". A "Guardar" button is at the bottom right. Callouts point to various fields with instructions: "Seleccione el tipo de empresa" (points to the dropdown), "Ingresar el CI de una persona natural para el representante legal de la empresa." (points to the legal representative field), "agregar datos de los doc. de la empresa" (points to the testimonio upload area), and "Se agrega los datos de la empresa NIT y nombre de la empresa" (points to the NIT and company name fields).

Una vez ingresado los datos requeridos del formulario, se procede a guardar todos los datos del formulario de una persona jurídica.

The screenshot shows the "Listado de unidad" page in the RECAUDACIONES system. The page has a sidebar menu on the left with options like "Inicio", "Configuración", "Personas", "Personal Trabajo", "Unidad", "Personal", "Servicio", and "Caja actual". The main content area shows a table with columns: "N°", "NOMBRE DE LA UNIDAD", "DESCRIPCIÓN", "CARGOS", and "ACCIONES". The table lists six units: 1. Unidad de Intendencia, 2. Área Administrativa, 3. Área Jurídica, 4. Área Social, 5. Área Técnica, and 6. Área Financiera. Callouts point to: "agregar nueva unidad" (points to a "+ Nuevo" button), "vista para añadir nueva unidad y personal resp." (points to the table header), "agregar nuevos cargos en la unidad" (points to "Cargos" buttons in the table), and "Click editar" (points to an "editar" button in the actions column).





GOBIERNO AUTÓNOMO MUNICIPAL DE CHULUMANI  
UNIDAD RECAUDACIONES

Registro de nuevos cargos de las unidades correspondientes.

The screenshot shows a modal window titled "Registrar Cargos" with a close button (X). It contains two input fields: "Nombre del cargo" (with placeholder "Ingrese el nombre del cargo") and "Descripción" (with placeholder "Ingrese la descripción del cargo"). A "Guardar" button is located below these fields. Below the form is a table with columns "N°", "CARGO", "DESCRIPCIÓN", and "ACCIONES". The table contains two rows of data. The "ACCIONES" column for each row contains two icons: a pencil (edit) and a trash can (delete). Annotations with blue arrows point to the "Guardar" button (labeled "Guardar datos"), the edit icon (labeled "Click editar"), and the delete icon (labeled "Click eliminar").

N°	CARGO	DESCRIPCIÓN	ACCIONES
1	tecnico	personal encargado en la supervisión del trabajo en terreno	[edit] [delete]
2	comisario	responsable para la verificación de la matriz del agua	[edit] [delete]

Seguidamente se tiene la vista de las unidades competentes al trabajo de la unidad.

The screenshot shows a web interface for "RECAUDACIONES". The left sidebar has a menu with "Personal Trabajo" selected. The main content area is titled "Listado de unidades" and displays a grid of six units, each with a description and an "Ingresar" button. Annotations with blue arrows point to the "Ingresar" button of the "Unidad de Intendencia" (labeled "Ingresar datos de los nuevos personales a cargo en la ejecución del trabajo de la instalación") and the "Unidad de personal técnico" in the sidebar (labeled "Unidad de personal técnico").

Unidad	Descripción	Acción
Unidad de Intendencia	son responsables del suministro de bienes y servicios a una organización, institución o gobierno	Ingresar
Área Administrativa	Se encarga de la gestión administrativa del municipio	Ingresar
Área Jurídica	Se encarga de la asesoría legal del municipio	Ingresar
Área Social	Se encarga de la promoción del desarrollo social del municipio	Ingresar
Área Técnica	Se encarga de la prestación de servicios y obras públicas municipales	Ingresar
Área Financiera	Se encarga de la administración de los recursos financieros del municipio	Ingresar





GOBIERNO AUTÓNOMO MUNICIPAL DE CHULUMANI  
UNIDAD RECAUDACIONES

A continuación, tenemos los personales activos para dicho trabajo técnico que se lleva en el proceso de la instalación.

The screenshot shows a web application interface for 'RECAUDACIONES'. The main content area is titled 'Listado de personal activo' and contains a table with the following data:

Nº	CI	NOMBRE Y APELLIDOS	FECHA DE CONTRATACIÓN	CARGO	ACCIONES
1	2428960-1U LP	OCTAVIO RAMIREZ PACO	2015-01-01	tecnico	[+], [✎], [✖]
2	7590091 BEN	LUIZ ENRIQUE SUAREZ HURTADO	2023-09-01	comisario	[+], [✎], [✖]

Callouts in the image point to various elements: 'Vista personal inactivo' points to a tab; 'Agregar nuevo personal de trabajo según a la instrucción de RRHH' points to a '+ Nuevo' button; 'Click eliminar' points to the red minus icon in the actions column; 'Click actualizar' points to the green plus icon; and 'Click editar' points to the blue pencil icon.

Registro de nuevo personal técnico en la instalación del suministro.

The screenshot shows a form titled 'Nuevo registro de personal de trabajo'. It is divided into two sections: 'PERSONA' and 'INFORMACIÓN'. The 'PERSONA' section has a field for 'INGRESE CI DE PERSONA' with the value '8431013' and a preview box showing 'Nombres y apellidos : JESUS PRUEVA CHAMBI CALLE' and 'Nº de celular : 78945601'. The 'INFORMACIÓN' section has fields for 'FECHA DE CONTRATACIÓN' (with a date picker), 'CARGO' (a dropdown menu), 'REFERENCIA LABORAL (TELEFONO O CELULAR)', and 'REFERENCIA LABORAL (NOMBRE DE LA PERSONA)'. A 'Guardar' button is at the bottom right. Callouts point to the CI field, the preview box, the date field, and the cargo dropdown.





GOBIERNO AUTÓNOMO MUNICIPAL DE CHULUMANI  
UNIDAD RECAUDACIONES

A continuación, se procede al llenado del formulario de la instalación ya sea un afiliado natural o jurídica.

**Nuevo registro de Instalación**

**INSTALACIÓN**

INGRESE CI: 8431015 SELECCIONE: [Selección persona jurídica]

**MOSTRAR DATOS**

Nombres y apellidos : JESUS PRUEVA CHAMBI CALLE  
Nº de celular : 78945601

**INSTALACIÓN**

SELECCIONE LA FECHA DE INSTALACIÓN: dd/mm/aaaa  
SELECCIONE LA CATEGORIA: [Selección la categoría]  
SELECCIONE LA SUB-CATEGORIA: [Selección la sub-categoría]

INGRESE EL MONTO DE INSTALACIÓN: Ingrese el monto de instalación  
GLOSA: Describa la razón por la cual desea instalar el servicio

**DIRECCION**

Guardar

Annotations:

- Fecha de inicio en la instalación del servicio.
- Selección de la categoría de cobro.
- Selección de la subcategoría del servicio
- Costo del servicio en la instalación.

Continuando con el formulario del proceso de la instalación del agua.

**Nuevo registro de Instalación**

**INSTALACIÓN**

SELECCIONE LA FECHA DE INSTALACIÓN: dd/mm/aaaa  
SELECCIONE LA CATEGORIA: [Selección la categoría]  
SELECCIONE LA SUB-CATEGORIA: [Selección la sub-categoría]

INGRESE EL MONTO DE INSTALACIÓN: Ingrese el monto de instalación  
GLOSA: Describa la razón por la cual desea instalar el servicio

**DIRECCION**

SELECCIONE LA PROPIEDAD: [Selección el tipo de propiedad]  
SELECCIONE LA ZONA: [Selección la zona]  
DIRECCIÓN DONDE SE HARÁ LA INSTALACIÓN: Ingrese la dirección donde se realizará la instalación

**ASIGNAR UNIDAD DE RESPONSABLE**

SELECCIONE UNIDAD RESPONSABLE: [Selección la unidad responsable]  
SELECCIONE CARGO: [Selección cargo]  
FUNCIONARIO RESPONSABLE: [Selección funcionario responsable]

Guardar

Annotations:

- Asignación de la unidad.
- Selección al cargo del personal principal.
- Selección al personal técnico.
- Guardar datos





GOBIERNO AUTÓNOMO MUNICIPAL DE CHULUMANI  
UNIDAD RECAUDACIONES

Clasificación del servicio ya instalados.

**RECAUDACIONES**

Inicio > Servicio > Listado de registros

**Servicios**

Servicio Activo | Servicio Inactivo

**Listado de servicio activo**

N°	TIPO	CI	NOMBRES Y APELLIDOS	FECHA DE INSTALACIÓN	FECHA FIN INSTALACIÓN	ESTADO	ACCIONES
1	Juridica	3605484 CBB	MARCELO RENZO CHULLUMANI	2015-01-01	2015-01-15	Activo	[+]
2	Natural	8431019 LP	JESUS PRUEVA CHAMBI CALLE	2023-11-23	2023-11-24	Activo	[+]
3	Natural	9685321 LP	JAIME TELLERIA VISCARRA	2016-02-02	2023-11-23	Activo	[+]
4	Juridica	3605484 CBB	MARCELO RENZO CHULLUMANI	2023-11-23	2023-11-24	Activo	[+]
5	Natural	2789126 LP	ISAAC ANTON GORENA ESPINOZA	2017-01-23	2023-11-24	Activo	[+]

Mostrando 1 a 5 de 5 entradas

2023 All rights Reserved

A continuación, se ve el reporte de las deudas y cobros pendientes de pago.

**RECAUDACIONES**

Inicio > Caja actual

**Monto total en caja**  
1,750.00 Bs  
Mil Setecientos Cincuenta Bolivianos

**Monto por la instalacion del servicio**  
1,750.00 Bs  
Mil Setecientos Cincuenta Bolivianos

**Monto por cobro del servicio**  
0.00 Bs  
CERO

Imprimir Detalle

Imprimir personas con deudas pendientes de una gestión específica

[Seleccione Gestión]

- 2015
- 2016
- 2017
- 2018
- 2019
- 2020
- 2021
- 2022
- 2023

Imprimir

Impresión del reporte de las deudas pendientes

2023 All rights Reserved





GOBIERNO AUTÓNOMO MUNICIPAL DE CHULUMANI  
UNIDAD RECAUDACIONES

Modulo caja.

Ingresar el CI para el cobro y pago del servicio

Busqueda por CI

+ Ingrese CI

8431019

CI:

8431019 LP

NOMBRES Y APELLIDOS:

JESUS PRUEVA CHAMBI CALLE

Natural

Nombre persona : JESUS PRUEVA CHAMBI CALLE

Ingresar

Vista previa de los datos del usuario

Ingresar al tipo de persona para el respectivo cobro del servicio

A continuación, se procede al proceso de cobro del servicio.

Click cobros por gestión

Click cobros por mes

Click en impresión comprobante de pago

CAJA

Search...

Corbos

Inicio

Cobros

Corbos específicos

DATOS - NATURAL

CI - NOMBRES Y APELLIDOS :

8431019 LP [ JESUS PRUEVA CHAMBI CHAMBI ]

CATEGORIA :

FAMILIA BASICA [ 10.00 Bs ]

AÑO Y MES DE INICIO A COBRAR :

Diciembre del 2023

ID	GESTIÓN	COBROS POR GESTIÓN	COBROS POR MESES	IMPRIMIR
1	2015	Cobrar Anual	Cobrar Mensual	Imprimir
2	2016	Cobrar Anual	Cobrar Mensual	Imprimir
3	2017	Cobrar Anual	Cobrar Mensual	Imprimir
4	2018	Cobrar Anual	Cobrar Mensual	Imprimir
5	2019	Cobrar Anual	Cobrar Mensual	Imprimir





GOBIERNO AUTÓNOMO MUNICIPAL DE CHULUMANI  
UNIDAD RECAUDACIONES

A continuación, se visualiza de manera detallada en el cobro mensual del servicio.

**Cobro mensual**

GESTIÓN : 2015

Selección en las casillas de todos los meses.

Sumatorio total a cobrar: Seleccionados Bs 40,00

Click en impresión comprobante de pago

Guardar datos

ID	MES	MONTO	MARCAR	ESTADO	ACCIONES
1	Enero	10.00 Bs	<input checked="" type="checkbox"/>	Pendiente	
2	Febrero	10.00 Bs	<input checked="" type="checkbox"/>	Pendiente	
3	Marzo	10.00 Bs	<input checked="" type="checkbox"/>	Pendiente	
4	Abril	10.00 Bs	<input checked="" type="checkbox"/>	Pendiente	
5	Mayo	10.00 Bs	<input type="checkbox"/>	Pendiente	
6	Junio	10.00 Bs	<input type="checkbox"/>	Pendiente	
7	Julio	10.00 Bs	<input type="checkbox"/>	Pendiente	
8	Agosto	10.00 Bs	<input type="checkbox"/>	Pendiente	
9	Septiembre	10.00 Bs	<input type="checkbox"/>	Pendiente	
10	Octubre	10.00 Bs	<input type="checkbox"/>	Pendiente	
11	Noviembre	10.00 Bs	<input type="checkbox"/>	Pendiente	
12	Diciembre	10.00 Bs	<input type="checkbox"/>	Pendiente	

Guardar





# 2023

# MANUAL

# TÉCNICO

**GOBIERNO AUTÓNOMO  
MUNICIPAL DE CHULUMANI**

V 1.0

Noemi Liz Solarez Chico





## SISTEMA WEB DE ADMINISTRACION Y CONTROL DE LA RECAUDACION DEL SUMINISTRO DE AGUA POTABLE

El sistema fue desarrollado pensando en la constante demanda de servicios informáticos, mediante el módulo de Control de Ingreso podrá tener una mejor atención, seguimiento y estabilidad respecto a las visitas al ingresar al módulo.

### Herramientas.

- PHP versión 8 o superior
- PostgreSQL en la versión 13.
- Laravel en su versión 10
- Bootstrap
- JavaScript
- Spatie.
- Visual Studio Code
- Sweet Alert.

### Instalación.

- Composer
- Navicat

### Configuración.

- Docker
- GitHub
- pgAdmin, PostgreSQL

### Seguridad.

- Capcha
- Encriptación de URL.
- *GetactivityLogOptions* (para en caso de registro de auditoria).





GOBIERNO AUTÓNOMO MUNICIPAL DE CHULUMANI  
UNIDAD RECAUDACIONES

El sistema cuenta con 3 módulos que tienen información y las diferentes acciones a realizar.

- ✚ Administración.
- ✚ Configuración de categorización de servicio.
- ✚ Control de la Recaudación.
- ✚ Caja.
- ✚ Reportes.





# 2023 MANUAL DE ADMINISTRACIÓN

---

**GOBIERNO AUTÓNOMO MUNICIPAL  
DE CHULUMANI**

V 1.0

Noemi Liz Solarez Chico





## LOGIN DEL SISTEMA

- 1.- para ingresar el nombre de usuario con lo que se registró en el sistema.
- 2.- Ingrese la contraseña con lo que se registró en el sistema.
- 3.- Ingresar el captcha de seguridad generado en el sistema.

Una vez ingresado los datos correctos se le presentara el módulo de administrador del sistema.

LOGIN

Usuario  
Ingrese su usuario

Contraseña  
Ingrese su contraseña

Captcha  
Ingrese captcha

INGRESAR



A continuación, tendrá el acceso total del sistema.

INICIO

Search...

BOL

Noemi Liz Solarez

MENU

Inicio

Acceso  
Solo para administradores del sistema  
SISTEMAS

Acceso  
Solo para la recaudación de fondos  
RECAUDACIONES

Acceso  
Solo para la administración de caja  
CAJA

Click e ingresar a la administración de usuarios

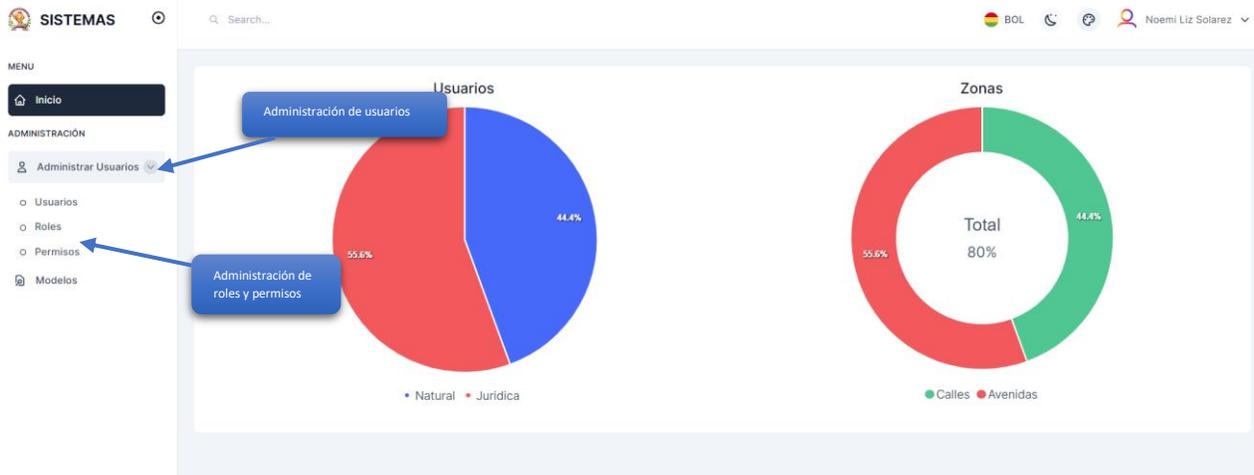
2023 All rights Reserved

NLIZ





### Panel de visualización administrativa de usuarios, roles y permisos.



### Vista de agregar nuevos usuarios.

The screenshot shows the 'Agregar nuevo usuario' interface. A table lists existing users with columns for ID, CI, Name, Cell, Email, Roles, and Status. Annotations point to various UI elements:

- 'Agregar nuevo usuario' button at the top right.
- 'Click en cambiar el estado' pointing to the 'ESTADO' column.
- 'Click en editar' pointing to the 'ACCIONES' column.
- 'Click en eliminar' pointing to the delete icon in the actions column.
- 'Nuevo' button with a plus sign.

N°	CI	NOMBRES Y APELLIDOS	CELULAR	EMAIL	ROLES	ESTADO	ACCIONES
1	4939207	Gustavo Isaias Flores Quino			Responsable	Activo	[Icons]
2	6805946	Merubia Eliana Salinas Ibañez			Encargada	Activo	[Icons]
3	6908251	Flora Antonia Miranda Allaga			Técnico Caja Responsable	Activo	[Icons]
4	8431010	Noemi Liz Solarez Chico	75824684	nliz99@gmail.com	Admin	Activo	[Icons]





## Vista Roles.

The screenshot shows the 'ROLES' management page. On the left is a sidebar with 'ADMINISTRACIÓN' containing 'Administrar Usuarios', 'Usuarios', 'Roles', 'Permisos', and 'Modelos'. The main area displays five role cards: 'Responsable', 'Encargada', 'Caja', 'Técnico', and 'admin'. Each card has 'Editar' and 'Eliminar' buttons. A '+ Nuevo' button is at the top right. Annotations include: 'Asignación de un nuevo rol' pointing to the '+ Nuevo' button; 'Click en editar rol' pointing to the 'Editar' button of the 'Caja' role; and 'Click en eliminar rol' pointing to the 'Eliminar' button of the 'Caja' role.

## Vista Permisos.

The screenshot shows the 'Permisos' management page. The sidebar is similar to the 'Roles' page, but 'Permisos' is selected. The main area features a table with columns 'ID', 'PERMISO', and 'ACCIONES'. The table contains three rows: ID 1 (actualizar), ID 2 (editar), and ID 3 (crear). Each row has 'Editar' and 'Eliminar' icons. A '+ Nuevo' button is at the top right. Annotations include: 'Asignación de un nuevo permiso' pointing to the '+ Nuevo' button; 'Click en editar el permiso' pointing to the 'Editar' icon of the first row; and 'Click en eliminar permiso' pointing to the 'Eliminar' icon of the first row.

ID	PERMISO	ACCIONES
1	actualizar	
2	editar	
3	crear	

