

UNIVERSIDAD PÚBLICA DE EL ALTO

CARRERA INGENIERÍA DE SISTEMAS



PROYECTO DE GRADO

SISTEMA DE INFORMACIÓN PARA EL CONTROL DE PAGOS POR EL SUMINISTRO DE AGUA POTABLE CASO: COMITÉ DE AGUA POTABLE LAKA PUCARA

Para optar al título de Licenciatura en Ingeniería de Sistemas
Mención: **INFORMÁTICA Y COMUNICACIONES**

POSTULANTE: MARCOS COLQUE TICONA

TUTOR METODOLÓGICO: M.Sc. ING. ENRIQUE FLORES BALTAZAR

TUTOR ESPECIALISTA: ING. JUAN REGIS MUÑOZ SIRPA

TUTOR REVISOR: ING. GABRIEL REYNALDO SIRPA HUAYHUA

EL ALTO – BOLIVIA

2020

DEDICATORIA

“El presente proyecto va dedicado con mucho cariño a mis queridos padres Fernando y Damiana, que siempre me dieron todo su apoyo y colaboración incondicional.”

AGRADECIMIENTOS

Agradezco a Dios por la fuerza y salud que me da para seguir adelante, agradezco a toda mi familia, por el apoyo que me brindo a lo largo de toda mi vida, e impulsarme a seguir y no rendirme.

A mi Tutor Metodológico por la enseñanza y colaboración brindada en Taller de Licenciatura II, las cuales me ayudaron en este proyecto.

A mi Tutor Especialista, por brindarme su tiempo, y por su colaboración durante todo el proceso de desarrollo de este proyecto.

A mi Tutor Revisor, por brindarme su tiempo, y por guiarme paso a paso durante todo el proceso de Taller de Licenciatura II.

A todos mis amigos y amigas por brindarme la ayuda y apoyarme en los momentos más difíciles.

¡Muchas Gracias a todos!!

RESUMEN

El proyecto “SISTEMA INFORMACIÓN DE CONTROL DE PAGOS POR EL SUMINISTRO DE AGUA POTABLE CASO: COMITÉ DE AGUA POTABLE LAKA PUCARA”, fue realizado para la Comunidad Laka Pucara, dicha Comité de Agua Potable tiene la función de proveer agua, control de pagos, mantenimiento y reparación en general.

El presente proyecto de grado se pretende automatizar los procesos de cobros de consumo de agua potable, la centralización de la información en una base de datos que permite acceder de manera inmediata a la información de los clientes, ya que estos procesos son de vital importancia para los de comité de agua potable.

La metodología empleada en el presente proyecto es UWE, que brinda un desarrollo de software orientado al web, que proporciona una serie de diagramas adecuados que facilitan la comprensión del desarrollo del sistema.

Para el desarrollo del sistema se utilizaron lenguajes como PHP y JavaScript, para la administración e base de datos es bajo el entorno MySQL, utilizando el servidor Apache XAMPP.

La medición de calidad del sistema se realizó basada en la NORMAS ISO 9126, que maneja los parámetros de usabilidad, funcionalidad, confiabilidad, mantenibilidad y portabilidad.

PALABRA CLAVE: control, pagos, suministro, agua, comité.

SUMMARY

The project "INFORMATION SYSTEM OF CONTROL OF PAYMENTS FOR THE SUPPLY OF DRINKING WATER CASE: LAKA PUCARA POTABLE WATER COMMITTEE", was carried out for the Laka Pucara Community, said Drinking Water Committee has the function of providing water, payment control, maintenance and repair in general.

The present degree project is intended to automate the processes of charges for drinking water consumption, the centralization of information in a database that allows immediate access to customer information, since these processes are of vital importance to those of the drinking water committee.

The methodology used in this project is UWE, which provides web-oriented software development, which provides a series of appropriate diagrams that facilitate understanding of system development.

For the development of the system languages such as PHP and JavaScript were used, for the administration of the database it is under the MySQL environment, using the Apache XAMPP server.

The quality measurement of the system was carried out based on the ISO 9126 STANDARDS, which manages the parameters of usability, functionality, reliability, maintainability and portability.

KEYWORD: *control, payments, supply, wáter, committee.*

ÍNDICE

	Págs.
DEDICATORIA	i
AGRADECIMIENTOS	ii
RESUMEN	iii
SUMMARY.....	iv
ÍNDICE	v
ÍNDICE DE FIGURAS	xi
ÍNDICE DE TABLAS.....	xiii
CAPITULO I	1
MARCO PRELIMINAR.....	1
1.1. INTRODUCCIÓN.....	2
1.2. ANTECEDENTES	3
1.2.1. Antecedentes de la Institución	3
1.3. PLANTEAMIENTO DEL PROBLEMA	5
1.3.1. Problema Principal	5
1.3.2. Problema Secundario	5
1.4. OBJETIVOS.....	6
1.4.1. Objetivo General	6
1.4.2. Objetivo Específicos	6
1.5. JUSTIFICACIÓN	7
1.5.1. Justificación Social.....	7
1.5.2. Justificación Económica.....	8
1.5.3. Justificación Técnica	8
1.6. METODOLOGÍA	8
1.6.1. Metodología de Ingeniería Web.....	8
1.6.2. Métricas de Calidad ISO 9126.....	9
1.6.3. Métricas de costos COCOMO II	9
1.6.4. ISO 27001 Seguridad de la Información	10
1.7. HERRAMIENTAS.....	10
1.8. LÍMITES Y ALCANCES	13

1.8.1. Límites	13
1.8.2. Alcances	13
1.9. APORTES.....	14
CAPITULO II	15
MARCO TEÓRICO	15
2.1. INTRODUCCIÓN.....	16
2.2. SISTEMA	16
2.3. DATO	16
2.4. INFORMACIÓN	17
2.5. FACTURACIÓN	17
2.6. SISTEMA DE INFORMACIÓN	18
2.7. CONTROL.....	19
2.7.1. Control de gestión	19
2.7.2. Control de financiero.....	19
2.7.3. Control de operaciones	20
2.7.4. Control de calidad	20
2.8. SUMINISTRO.....	20
2.9. INGENIERÍA WEB	21
2.10. SOFTWARE	23
2.11. INGENIERÍA DE SOFTWARE	23
2.11.1. Modelos para el desarrollo de software.....	24
2.11.1.1. El modelo de desarrollo evolutivo (espiral)	25
2.11.1.2. Arquitectura Cliente – Servidor	26
2.12. METODOLOGÍA DE INGENIERÍA WEB	27
2.12.1. Metodología uwe	27
2.12.2. Características de UWE	29
2.12.3. Actividades de modelado de UWE	29
2.12.4. Etapas de modelado.....	29
2.12.4.1. Modelo de requerimientos.....	30
2.12.4.2. Modelo de contenido	31
2.12.4.3. Modelo de navegación	32
2.12.4.4. Modelo de presentación	34

2.12.4.5. Modelo de procesos	36
2.12.5. Fases de la metodología UWE	36
2.12.5.1. Captura, análisis y especificación de requisito	36
2.12.5.2. Diseño del sistema.....	36
2.12.5.3. Codificación del software	37
2.12.5.4. Pruebas.....	37
2.12.5.5. La Instalación o Fase de Implementación	37
2.12.5.6. El Mantenimiento.....	37
2.13. HERRAMIENTAS DE DESARROLLO DE SOFTWARE	37
2.13.1. Servidor Apache	37
2.13.2. Base de Datos.....	39
2.13.3. Gestor de base de datos MySql	39
2.13.4. MySQL Workbench.....	40
2.13.5. Lenguaje de programación “PHP”	40
2.13.6. HTML	41
2.13.7. CSS3.....	41
2.13.8. Javascript.....	42
2.13.9. Framework Bootstrap	42
2.13.10. Framework Codeigniter.....	43
2.14. INGENIERÍA DE REDES	44
2.15. MÉTRICAS DE CALIDAD DE SOFTWARE	46
2.15.1. Calidad de Software	46
2.15.2. La Norma de Evaluación ISO/IEC 9126	46
2.15.2.1. Funcionalidad	48
2.15.2.2. Confiabilidad	49
2.15.2.3. Usabilidad.....	50
2.15.2.4. Eficiencia	52
2.15.2.5. Capacidad de mantenimiento	53
2.15.2.6. Portabilidad	54
2.15.2.7. Calidad de uso	55
2.16. COSTOS	56
2.16.1. Cocomo	56

2.16.2. Cocomo II	57
2.16.2.1. Características	57
2.16.2.2. Modelos de COCOMO II	58
2.16.2.3. Modelos de Estimación	58
2.17. ISO 27001 SEGURIDAD DE LA INFORMACIÓN	63
CAPITULO III	66
MARCO APLICATIVO.....	66
3.1. INTRODUCCIÓN.....	67
3.2. DESCRIPCIÓN GENERAL DE COMITÉ DE AGUA POTABLE LAKA PUCARA	67
3.3. ESTRUCTURA DE COMITÉ DE AGUA POTABLE LAKA PUCARA.....	68
3.4. ESQUEMA DEL SISTEMA	68
3.5. OBTENCIÓN DE REQUERIMIENTOS	70
3.6. INGENIERÍA DE REQUERIMIENTOS.....	71
3.6.1. Requerimientos funcionales.....	72
3.6.2. Requerimientos no funcionales.....	73
3.7. MODELADO DEL SISTEMA POR MEDIO DE LA METODOLOGÍA UWE	73
3.7.1. Modelo Caso de Uso	73
3.7.1.1. Identificación de actores	73
3.7.1.2. Caso de uso general	74
3.7.1.3. Caso de Uso: Acceso al Sistema.....	75
3.7.1.4. Caso de Uso: Gestión de Usuarios	77
3.7.1.5. Caso de Uso: Gestión de Clientes	79
3.7.1.6. Caso de Uso: Registro de Lectura de Medidor de Agua.....	80
3.7.1.7. Caso de Uso: Consulta y Pago de Planillas	82
3.7.1.8. Caso de Uso: Emisión de Comprobante de pago	83
3.7.1.9. Caso de Uso: Gestión de Tarifas	85
3.7.1.10. Caso de Uso: Informes e Reportes.....	86
3.8. MODELO CONCEPTUAL	87
3.9. MODELO DE NAVEGACIÓN.....	89
3.10. MODELO DE PRESENTACIÓN.....	92
3.11. MODELO DE IMPLEMENTACIÓN	97
3.11.1. Acceso al Sistema	97

3.11.2. Menú Principal del Usuario Administrador	99
3.11.3. Menú Principal del Usuario Tesorera	100
3.11.4. Menú Principal del Usuario Operador	100
3.11.5. Modulo Usuarios del Usuario Administrador	101
3.11.6. Modulo Clientes del Usuario Administrador	102
3.11.7. Módulo de Lecturas del Usuario Operador	105
3.11.8. Módulo de Planillas de Consumo del Usuario Tesorera	106
3.11.9. Módulo de Reportes del Usuario Administrador	110
3.12. CONFIGURACIÓN DE LA RED DE COMPUTADORAS	112
CAPITULO IV	114
MÉTRICAS DE CALIDAD Y COSTOS	114
4.1. MÉTRICAS DE CALIDAD DE SOFTWARE	115
4.1.1. Funcionalidad	116
4.1.2. Confiabilidad	122
4.1.3. Usabilidad	123
4.1.4. Mantenibilidad	123
4.1.5. Portabilidad	124
4.2. MÉTODO DE ESTIMACIÓN DE COSTO DE SOFTWARE	125
4.2.1. Método de estimación de COCOMO	125
4.2.1.1. Costo y estimación de esfuerzo del desarrollo del proyecto	127
4.2.1.2. Estimación de esfuerzo del proyecto	127
4.2.1.3. Costo de desarrollo	129
4.3. SEGURIDAD DE LA INFORMACIÓN DEL SISTEMA	129
4.3.1. Seguridad de base de datos	130
4.3.2. Seguridad de autenticación	130
4.3.3. Seguridad de la aplicación	130
CAPITULO V	131
PRUEBAS Y RESULTADOS	131
5.1. INTRODUCCIÓN	132
5.2. REGISTRO DE CLIENTES	132
5.3. REGISTRO DE LECTURAS	133
5.4. GENERACIÓN DE PLANILLAS DE CONSUMO DE AGUA	133

5.5. INFORME ANUAL DE PAGOS	134
CAPITULO VI.....	135
CONCLUSIONES Y RECOMENDACIONES.....	135
6.1. CONCLUSIONES	136
6.2. RECOMENDACIONES	137
BIBLIOGRAFÍA.....	138
ANEXOS.....	143

ÍNDICE DE FIGURAS

<i>Figura 2.1. Esquema del Modelo Cliente Servidor</i>	27
<i>Figura 2.2. Descripción general de los modelos UWE</i>	30
<i>Figura 2.3. Diagrama de Caso de Uso de la Metodología UWE</i>	31
<i>Figura 2.4. Diagrama de Contenido UWE</i>	32
<i>Figura 2.5. Estereotipos e Iconos de Diagrama de Navegación</i>	33
<i>Figura 2.6. Diagrama de Navegación UWE</i>	34
<i>Figura 2.7. Estereotipos e Iconos de Diagrama de Presentación</i>	35
<i>Figura 2.8. Diagrama de Presentación UWE</i>	35
<i>Figura 2.9. Servidor Apache</i>	38
<i>Figura 2.10. MVC de Codeigniter</i>	44
<i>Figura 2.11. Norma de Evaluación ISO/IEC 9126</i>	47
<i>Figura 2.12. Evaluación Interna, Externa y Calidad de Uso ISO/IEC 9126</i>	48
<i>Figura 2.13. Característica de Funcionalidad ISO/IEC 9126</i>	49
<i>Figura 2.14. Característica de Confiabilidad ISO/IEC 9126</i>	50
<i>Figura 2.15. Característica de Usabilidad ISO/IEC 9126</i>	51
<i>Figura 2.16. Característica de Eficiencia ISO/IEC 9126</i>	52
<i>Figura 2.17. Característica de Mantenimiento ISO/IEC 9126</i>	53
<i>Figura 2.18. Característica de Portabilidad ISO/IEC 9126</i>	54
<i>Figura 2.19. Característica de Calidad de Uso ISO/IEC 9126</i>	55
<i>Figura 2.20. Clasificación de Puntos Objetos</i>	59
<i>Figura 2.21. Puntos Función Determinación del Peso</i>	62
<i>Figura 2.22. Peso del Factor de Complejidad</i>	62
<i>Figura 3.23. Estructura Organizacional de CAPyS Laka Pucara</i>	68
<i>Figura 3.24. Esquema del Sistema</i>	69
<i>Figura 3.25. Diagrama de Caso de Uso General</i>	75
<i>Figura 3.26. Diagrama de Caso de Uso Acceso al Sistema</i>	75
<i>Figura 3.27. Diagrama de Caso de Uso Gestión de Usuarios</i>	77
<i>Figura 3.28. Diagrama de Caso de Uso Gestión de Clientes</i>	79
<i>Figura 3.29. Diagrama de Caso de Uso Registro de Medidor</i>	80
<i>Figura 3.30. Diagrama de Caso de Consulta de Planilla</i>	82
<i>Figura 3.31. Diagrama de Caso de Comprobante de pago</i>	83
<i>Figura 3.32. Diagrama de Caso de Gestión de Tarifas</i>	85
<i>Figura 3.33. Diagrama de Caso de Informe e Reportes</i>	86
<i>Figura 3.34. Modelo de Contenido</i>	88
<i>Figura 3.35. Modelo Navegación de Administración de Usuarios</i>	89
<i>Figura 3.36. Modelo Navegación de Administración de Clientes</i>	90
<i>Figura 3.37. Modelo Navegación de Administración de Detalle Consumo</i>	90
<i>Figura 3.38. Modelo Navegación de Administración de Lecturas</i>	91
<i>Figura 3.39. Modelo Navegación de Administración de Reportes</i>	91
<i>Figura 3.40. Modelo de Presentación de Autenticación del Sistema</i>	92
<i>Figura 3.41. Modelo de Presentación de Inicio (Administrador)</i>	93
<i>Figura 3.42. Modelo de Presentación de Inicio (Operador)</i>	93
<i>Figura 3.43. Modelo de Presentación Administración de Usuarios (Administrador)</i>	94
<i>Figura 3.44. Modelo de Presentación Administración de Clientes (Administrador)</i>	95
<i>Figura 3.45. Modelo de Presentación Administración de Lecturas (Operador)</i>	96

Figura 3.46. Modelo de Presentación Administración de Detalle Consumo (Tesorera)	96
Figura 3.47. Modelo de Presentación Administración de Reportes (Administrador)	97
Figura 3.48. Interfaz de Autenticación de Usuario	98
Figura 3.49. Interfaz Menú Principal	99
Figura 3.50. Interfaz Menú Principal del Tesorera	100
Figura 3.51. Interfaz Menú Principal del Operador	100
Figura 3.52. Interfaz Registro de Usuarios	101
Figura 3.53. Interfaz Administración de Usuarios	102
Figura 3.54. Interfaz de Registro Nuevos Clientes	103
Figura 3.55. Interfaz de administración de clientes	104
Figura 3.56. Interfaz de Registro de Lecturas.	105
Figura 3.57. Interfaz de Administración de Lecturas	106
Figura 3.58. Interfaz de Detalle de Consumo	107
Figura 3.59. Interfaz Aviso de Cobranza	107
Figura 3.60. Interfaz de Formulario de Cobranza	108
Figura 3.61. Interfaz de Boleta Comprobante de Pagos	109
Figura 3.62. Interfaz de Reportes	110
Figura 3.63. Informe de Recaudaciones por Meses	111
Figura 3.64. Router wifi para Adsl Telefónica	112
Figura 3.65. Implementación de Red inalámbrica	113
Figura 3.66. Instrucciones para Conectarse desde un Notebook	113

ÍNDICE DE TABLAS

Tabla 2.1. <i>Peso de un Punto Objeto</i>	60
Tabla 2.2. <i>Productividad para el modelo Composición de Aplicación</i>	60
Tabla 3.3. <i>Tareas de Obtención de Requisitos</i>	70
Tabla 3.4. <i>Requerimientos Funcionales</i>	72
Tabla 3.5. <i>Requerimientos No Funcionales</i>	73
Tabla 3.6. <i>Descripción de Actores del Sistema</i>	74
Tabla 3.7. <i>Descripción de Acceso al Sistema</i>	76
Tabla 3.8. <i>Descripción de Gestión de Usuarios</i>	77
Tabla 3.9. <i>Descripción de Gestión de Clientes</i>	79
Tabla 3.10. <i>Descripción de Lectura de Medidor</i>	81
Tabla 3.11. <i>Descripción de Consulta y Pago de Planillas</i>	82
Tabla 3.12. <i>Descripción de Comprobante de pago</i>	84
Tabla 3.13. <i>Descripción de Gestión de Tarifas</i>	85
Tabla 3.14. <i>Descripción de Reporte e Informe</i>	87
Tabla 4.15. <i>Entrada de Usuario</i>	116
Tabla 4.16. <i>Salidas de Usuario</i>	117
Tabla 4.17. <i>Peticiones de Usuario</i>	117
Tabla 4.18. <i>Numero de archivos</i>	118
Tabla 4.19. <i>Numero de Interfaces Externas</i>	119
Tabla 4.20. <i>Parámetros de medición según Punto de Función</i>	119
Tabla 4.21. <i>Ponderación de Ajuste de Complejidad</i>	120
Tabla 4.22. <i>Ponderación de Usabilidad</i>	123
Tabla 4.23. <i>Calidad Global</i>	125
Tabla 4.24. <i>Coficiente COCOMO</i>	127
Tabla 5.25. <i>Prueba y Resultados Registro de Clientes</i>	132
Tabla 5.26. <i>Prueba y Resultados Registro de Lecturas</i>	133
Tabla 5.27. <i>Prueba y Resultados Generación de Planillas</i>	133
Tabla 5.28 <i>Prueba y Resultados Informe Pagos</i>	134

A thick, dark blue vertical bar is positioned on the left side of the page. From the bottom of this bar, several thin, curved lines in shades of blue and grey extend upwards and to the right, creating an abstract, organic shape.

CAPITULO I

MARCO PRELIMINAR

1.1. INTRODUCCIÓN

En la actualidad el avance tecnológico se ha convertido en un elemento fundamental en el desarrollo y crecimiento de las empresas sean éstas públicas o privadas, gracias a la tecnología se ha llegado a resolver los problemas a través de sistemas innovadoras que son adaptables a las necesidades de cada una, que permiten un fácil manejo, acceso y tratamiento de la información.

El presente proyecto consiste en desarrollar un sistema de información, que permitirá de llevar el proceso de control de pagos y seguimiento a la información, beneficiando a comité de agua potable y saneamiento (CAPyS) y a la comunidad optimizando la administración de pagos por el consumo de agua potable.

Comité de agua potable y saneamiento (CAPyS) es la entidad conformada por los usuarios al servicio de la comunidad, de llevar un cobro adecuado sobre todo la rendición de cuentas a la población.

La metodología empleada en el presente proyecto es UWE (UML- Based Web Engineering), para el desarrollo del sistema se utilizarán lenguajes como PHP5 y el uso de framework CodeIgniter, para la administración de base de datos es bajo el entorno MySQL, utilizando el servidor Apache XAMPP.

1.2. ANTECEDENTES

1.2.1. Antecedentes de la Institución

La comunidad Laka Pucara fue fundada desde muchos años atrás y posteriormente refrendado con personería jurídica por el Gobierno Autónomo Municipal de Corocoro con resolución N° 29/96 en fecha 11/11/1996, se encuentra en el departamento de La Paz, provincia Pacajes, primera Sección Corocoro a 110 kilómetros de la sede de gobierno.

Las autoridades de la comunidad y conjuntamente en coordinación con las autoridades del municipio, gobernación y gobierno central a través de Ministerio de Medio Ambiente y Agua, lograron garantizar a la población el servicio de agua potable que son un derecho humano, se inició con la perforación de agua en el año 2016, esta ha tenido éxito teniendo una aproximada de bombeo de 5 lt/seg, luego de haber realizado la perforación se ha dado la siguiente fase que es la distribución de agua mediante cañerías a domicilio a la población en general, la administración está a cargo de Comité de Agua Potable y Saneamiento (CAPyS) que es una entidad conformada por los usuarios al servicio a la comunidad, el comité se organiza diferenciando dos unidades que corresponde a las funciones básicas de una entidad prestadora de servicios de agua y saneamiento:

- Gestión técnica (operación y mantenimiento de los servicios).
- Gestión administrativa y comercial (manejo administrativo y cobranzas por los servicios).

Para el control de consumo de agua a los usuarios se realiza a través de un medidor de agua, las tarifas son aprobadas en la asamblea tomando en cuenta todos los costos de administración, operación y mantenimiento, mínimo de consumo es de menor o igual a 10m³ por cliente que es equivalente a un valor de 8,00 Bs al final de cada año rinde cuentas a la Asamblea, el Comité de Agua Potable.

Trabajos a fines al proyecto.

- **A Nivel Internacional**

- (Gonzales & Velastegui, 2008) **“ESTUDIO DE PROTOCOLOS ENTRE PDA/PC Y SU APLICACIÓN EN EL SISTEMA DE FACTURACIÓN DE LAS EAPA SAN MATEO”**. Su objetivo general es Desarrollar un sistema que optimice la extracción de información e ingreso de datos de lectura de consumo de agua potable al Sistema Comercial de la Empresa de Agua Potable y alcantarillado “San Mateo”. Realizado en la Escuela Superior Politécnica, Chimborazo, Ecuador.
- (Figueroa, 2018). **“SISTEMA DE FACTURACIÓN UTILIZANDO LOS FRAMEWORK ANGULAR 5, LARAVEL 5.5 PARA LA JUNTA ADMINISTRADORA DE AGUA POTABLE SAN JOSÉ DE CHORLAVI DE LA CIUDAD DE IBARRA - ECUADOR”**. Su objetivo general es Desarrollar un Sistema de facturación utilizando los framework Angular 5, Laravel 5.5 para la junta administradora de agua potable San José de Chorlovi de la ciudad de Ibarra – Ecuador.

- **A Nivel Nacional**

- (Dipp, 2010) **“SISTEMA DE CONTROL DE COBROS DE AGUA POTABLE PARA COMITÉS AFILIADOS UTILIZANDO FRAMEWORK CAKE”**. Su objetivo general es Desarrollar un Sistema de Control de Cobros de Agua Potable para Comités de Agua Afiliados utilizando Framework CAKEPHP. Realizado en la Universidad Mayor San Simón, Cochabamba, Bolivia.
- (Aramayo Canazas, 2009). **“SISTEMA DE FACTURACIÓN DE CONSUMO DE AGUA POTABLE PARA LA PROVINCIA INQUISIVI”**. El objetivo general es 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. Realizado en la Universidad Mayor de San Andrés, La Paz, Bolivia.

1.3. PLANTEAMIENTO DEL PROBLEMA

1.3.1. Problema Principal

La institución encargada de control de agua, se rige bajo una norma emanada por las autoridades del gobierno a través del Ministerio de Medio Ambiente y Agua que tiene como propósito de regular los costos de operación para brindar el servicio y su propia administración a través de un sistema de tarifario que muchas veces no resulta eficiente.

Los comités de agua potable de la comunidad Laka Pucara, toda la información que obtienen durante la lectura de los medidores en los domicilios de cada cliente, el registro de planillas de consumo, la elaboración de reportes y la entrega de información al cliente, se viene realizando de forma manual, mediante el uso de formatos pre impresos y los cálculos se realizan utilizando una calculadora, factor que incide en la pérdida de tiempo tanto para el cliente y la persona encargada de este proceso, a veces suele confundirse o perder la información, lo que genera inconformidad por parte de los clientes en algunas de sus tarifas a cancelar, esto proceso de una manera que no brinda garantías necesarias al momento de entregar la información.

Otro factor determinante es no contar con respaldos de la información dando como resultado reportes incompletos en CAPyS, generando dificultades al momento de tomar decisiones factor que incide en pérdida económica a la institución.

1.3.2. Problema Secundario

- Falta de información con certeza en la cantidad del consumo de agua que realizan mensualmente los consumidores la cual influye en toma de decisiones.
- No se lleva el registro adecuado de los pagos y multas realizado por los clientes.

- Comité de agua potable tienen errores en generar reportes de recaudación por el consumo de agua.
- Falta de organización de los datos.
- Cuando los clientes se acercan a las oficinas de CAPyS a cancelar sus valores, suele darse el inconveniente de que no les alcanza el dinero.

Nos planteamos la siguiente interrogante:

¿De qué manera se puede mejorar el proceso de registro de planillas y control de pagos por el consumo de agua en el comité de agua potable Laka Pucara?

1.4. OBJETIVOS

1.4.1. Objetivo General

Desarrollar un sistema de información para el control de pagos por el suministro de agua potable, que permita un mejor proceso de recaudación por el consumo en el comité Laka Pucara, situada en provincia pacajes, departamento de La Paz.

1.4.2. Objetivo Específicos

- Realizar el análisis de la situación actual.
- Implementar un módulo de administración de usuarios.
- Diseñar los componentes del sistema de recaudación de tarifas el cual permita llevar un registro actualizado de información.
- Realizar el diseño conceptual lógico y físico de la base de datos, que permita responder a los requerimientos del usuario.
- Diseñar un módulo de facturación y control de pagos.
- Implementar módulo de reportes sobre la información generada.
- Evaluar los resultados obtenidos con el sistema utilizando la norma ISO 9126.

1.5. JUSTIFICACIÓN

1.5.1. Justificación Social

En las instituciones es importante, contar con los servicios de los sistemas de información hoy en día, se han convertido en el pilar fundamental en cualquier empresa u organización, y en concreto los que están orientados al manejo de los servicios básicos como el agua, sin duda son muy necesario diseñarlo, desarrollarlo e implementarlo, orientados para que se optimice los procesos de administración de pagos.

El sistema beneficiará disminuyendo el trabajo de forma manual, durante los procesos de operación de cálculo de tarifas de consumo, elaboración de informes en cuanto a los pagos mensuales. Además, beneficiará directamente al cliente, ya que podrá realizar procesos de consulta de consumo de agua de manera rápida y confiable, tendrá la posibilidad de realizar su seguimiento de consumo a través del internet, evitando ir a la oficina de CAPyS.

Todo ello representa ventajas y facilidades de una forma u otra mejora la administración de comité de agua potable Laka Pucara para un mejor servicio a la sociedad en su conjunto, beneficiando de la siguiente manera.

- **Beneficiarios Directos**

- Comité de Agua Potable
- Autoridades de la comunidad Laka Pucara
- Los usuarios de Agua Potable

- **Beneficiarios Indirectos**

- La población en conjunto de la comunidad Laka Pucara en sus 3 zonas.

1.5.2. Justificación Económica

El sistema de información es un recurso muy importante para la institución que maneja datos, al proponer un sistema de información se reducirá los errores de registro de usuarios y cobranzas, así también mejorara el procedimiento y manejo de la información el cual reducirá el tiempo de elaboración de las planillas y buen control de pagos.

La comunidad y comité de agua potable CAPyS será beneficiada con el desarrollo de software de manera gratuita, lo que implica que las herramientas utilizadas no tienen costo alguno es decir el lenguaje de programación (PHP), gestor de Base de Datos (MySQL) y el servidor de páginas Web (Apache) no requieren licencias.

1.5.3. Justificación Técnica

El presente trabajo de investigación se justifica debido a que se proyecta diseñar un sistema de información para el control de pagos por el suministro de agua potable, para tal propósito me apoyare en la metodología de desarrollo UWE, basada en la herramienta UML (Lenguaje de Modelo Unificado).

En la actualidad cuentan con la tecnología y el equipamiento necesario de hardware, que viabiliza la realización del presente proyecto. De las cuales se puede evidenciar una computadora de escritorio con procesador de Core Duo y sistema operativo Windows 7, que cumple con lo necesario para la implementación del sistema a desarrollarse.

1.6. METODOLOGÍA

1.6.1. Metodología de Ingeniería Web

UML-Based Web Engineering (UWE) es una metodología de desarrollo de aplicaciones web, utilizada en la ingeniería web, dedicado a la sistematización y personalización, es decir realizar sistemas adaptativos.

UWE es un proceso, iterativo e incremental, incluye flujos de trabajo y puntos de control, las fases coinciden con las propuestas en el Proceso Unificado de Modelado y UML, pero adaptada a la web.

El lenguaje UWE posee definiciones que representan características específicas y necesarias para el diseño de modelos en el dominio Web y el hecho de ser una ramificación del lenguaje UML le provee de la flexibilidad necesaria para la definición en este dominio. Como el lenguaje UML es un lenguaje de amplio uso en la mayoría de las herramientas CASE y en la ingeniería de software en general, la aplicación de UWE es de fácil entendimiento y de simple utilización. (Metodología UWE, 2015)

1.6.2. Métricas de Calidad ISO 9126

Es un estándar internacional para evaluación de la calidad de software. El estándar está dividido en cuatro partes las cuales dirigen, realidad, métricas externas, métricas internas y calidad en las métricas de uso y expendido. El modelo de calidad establecido en la primera parte del estándar, ISO 9126-1, clasifica la calidad del software en un conjunto estructurado de características y subcaracterísticas de la siguiente manera: Funcionalidad, Fiabilidad, Usabilidad, Eficiencia, Mantenibilidad, Portabilidad, Calidad de Uso. (Medina Prieto, 2017)

1.6.3. Métricas de costos COCOMO II

COConstructive COst MOdel II (COCOMO II) es una modelo que permite estimar el costo, el esfuerzo, y el horario en la planificación de una nueva actividad de desarrollo de software. COCOMO II es la última extensión importante a la COCOMO originales (COCOMO 81) modelo publicado en 1981. Se compone de tres sub-modelos, cada uno que ofrece una mayor fidelidad el más largo de uno está en la planificación del proyecto y el proceso de diseño. Enumerado aumento de la fidelidad, estos sub-modelos se llaman la Composiciones Aplicaciones, Diseño temprana, y los modelos posteriores a la arquitectura. (Administración de Proyectos de Software, 2015)

1.6.4. ISO 27001 Seguridad de la Información

La ISO 27001 es una norma internacional de Seguridad de la Información que pretende asegurar la confidencialidad, integridad y disponibilidad de la información de una organización y de los sistemas y aplicaciones que la tratan. Este estándar ha sido desarrollado por la Organización Internacional de Normalización (ISO: “International Organization for Standardization”) y por la Comisión Electrotécnica Internacional (IEC: “International Electrotechnical Commission”).

Hay que garantizar su confidencialidad (sólo las personas autorizadas pueden acceder a esta), su integridad (no ha sido manipulada de manera no autorizada) y su disponibilidad (la información puede ser accedida por las personas autorizadas cuando lo necesitan), mediante una gestión de los riesgos que considera a las personas, procesos y sistemas de TIC (Tecnología de la Información y las Comunicaciones) relacionados con la misma. (Unir Revista, 2019)

1.7. HERRAMIENTAS

Para el desarrollo del sistema de información de control de pagos por el suministro de agua potable los recursos utilizados en el desarrollo de software; PHP como lenguaje de programación, con el framework Codeigniter, MySQL para la base de datos y servidor web Apache, para el lado del cliente JavaScript y estilos CSS.

- **Modelo de la Base de Datos**

MySQL Workbench es una herramienta visual unificada para arquitectos de bases de datos, desarrolladores y DBA. MySQL Workbench proporciona modelado de datos, desarrollo de SQL y herramientas de administración integrales para la configuración del servidor, la administración de usuarios, la copia de seguridad y mucho más. (MySQL, s.f.)

- **Servidor de Aplicación Web**

Apache es un software especializado en ofrecer servicios de servidor web. Es versátil, ligero y muy útil, además de ser completamente gratuito y de código abierto. Aunque se le conoce así, su nombre completo es Apache HTTP Server, y sus responsables tienen también un nombre similar: Apache Software Foundation.

Lleva en activo desde el año 1995, tiempo más que suficiente para erigirse como el estándar que es en la actualidad. Fiable, robusto y muy flexible, permite al dueño de cualquier web publicar el contenido que desea en esta, como también gestionar todos sus ficheros de forma fácil y sencilla. (Neo Attack, s.f.)

- **Lenguaje de Programación**

PHP se utiliza principalmente para crear páginas web, para crear contenido dinámico y para trabajar con bases de datos y HTML.

PHP son las siglas en inglés del acrónimo Hypertext Pre-Processor, es decir, pre-procesador de hipertexto. Es un lenguaje de programación de propósito general que se ejecuta en el lado del servidor. Es un lenguaje interpretado. Tiene múltiples formas de utilizarse, ya que puede utilizarse con scripts, de forma estructurada o programación en objetos. Está creado con la licencia de software libre PHPv3_01, que es una licencia Open Source. (Arenols Solano, 2019)

- **Gestor de Base de Datos**

MySQL es el sistema de gestión de bases de datos relacional más extendido en la actualidad al estar basada en código abierto.

MySQL presenta algunas ventajas que lo hacen muy interesante para los desarrolladores. La más evidente es que trabaja con bases de datos relacionales, es decir, utiliza tablas múltiples que se interconectan entre sí para almacenar la información y organizarla correctamente. (Robledano, 2019)

- **Bootstrap para Front - end**

Bootstrap es conjunto conceptos, prácticas y criterios (framework) desarrollado por Mark Otto y Jacob Thornton dentro de Twitter con la intención de estandarizar el conjunto de herramientas que utilizaban todos los involucrados en el desarrollo del front-end.

Bootstrap nos ayuda a maquetar un sitio web con rapidez y, sobretodo, ayudándonos a que el diseño sea correcto y usable tanto en dispositivos convencionales como en los táctiles (responsive web design). Para hacerlo, nos ofrece una serie de estilos CSS y librerías JavaScript que nos ayudarán de una manera rápida a desarrollar nuestro sitio web y sobretodo es recomendable para el desarrollo de prototipos y tener un tiempo de respuesta realmente bueno. (Ruiz, 2014)

- **Codeigniter para Back-end**

Es un entorno de desarrollo web escrito en PHP que presume de acelerar y optimizar el desarrollo de aplicaciones web gracias a un compacto diseño de software. La compañía de software norteamericana EllisLab fue la encargada de su creación y de la publicación de su primera versión en febrero de 2006.

El diseño orientado al rendimiento de este framework de desarrollo web se revela en su parca arquitectura, pues se basa en el patrón Modelo-Vista-Controlador (MVC). El principio fundamental que sustenta a la arquitectura de desarrollo MVC es la estricta separación entre el código y la presentación, gracias a una estructura modular de software y a la externalización del código PHP. (Digital Guide IONOS, 2020)

- **Lenguaje JavaScript**

JavaScript es uno de los lenguajes de programación más utilizados y conocidos, ya que este permite crear páginas dinámicas y llamativas en las que se puede interactuar más con los usuarios.

Como, por ejemplo, gracias a este lenguaje la experiencia visual del usuario más atractiva al momento de actualizar tus estados de Facebook, Instagram, Twitter y otras redes sociales. Cabe resaltar que JavaScript se ejecuta en el ordenador del usuario y actualmente también se ejecuta en el servidor. JavaScript es uno de los lenguajes de programación más utilizados y la razón de su popularidad es sencilla: permite desarrollos web tanto simples como complejos. Dicho de otra manera, con él podemos desde desarrollar aplicaciones web, hasta validar formularios de contacto. (Barrera, s.f.)

1.8. LÍMITES Y ALCANCES

1.8.1. Límites

- El desarrollo del sistema será diseñado e implementado para el Comité de Agua Potable Laka Pucara (CAPyS) ubicado en la Provincia Pacajes, Municipio Corocoro, con las características y requerimientos propios para el sistema de información de control de pagos por el suministro de agua potable, no tomando en cuenta otro tipo de información.
- El sistema será administrado por los usuarios del comité de agua potable, cada encargado tendrá un tipo de usuario de acceso al sistema.

1.8.2. Alcances

El Sistema de información de control de pagos por el suministro de agua potable será una aplicación Web a la que se podrá acceder con los usuarios con acceso autorizado, estos podrán registrar y obtener información de acuerdo a sus roles, los módulos que integrarán el sistema y sus correspondientes funcionalidades son:

- Módulo de registros de clientes de agua potable, que el almacenamiento se realizara en la base de datos mediante formulario.
- Módulo de pagos permitirá generar comprobante de pago por los conceptos de consumo de agua potable.

- Módulo de lecturas permite que el operador registre la lectura de medidor de cada cliente por el consumo de agua potable.
- Módulo de multas, permitirá sancionar el incumplimiento de la normativa vigente en la institución.
- Módulo de administración de usuarios, dónde se otorgará privilegios de acceso al sistema.
- Módulo de reportes e informes, permitirá la generación de todo los informes y reportes.

1.9. APORTES

El aporte que se desea brindar con el presente proyecto es la implementación de un sistema web logrando colaborar:

- A la población en conjunta de la comunidad Laka Pucara, con el sistema tendrán confiabilidad y rapidez en el manejo de la información para mejor administración de agua potable.
- A los de comité de agua potable, con el sistema se automatizará la entrega de comprobante del consumo de agua.
- A los usuarios, el sistema propuesto permite almacenar en línea toda la información del servicio de agua, permitiendo una atención más integral, rápida y oportuna; trayendo consigo altos grados de satisfacción.

A thick, dark blue vertical bar is positioned on the left side of the page. From the bottom of this bar, several thin, light blue lines curve upwards and outwards, creating an abstract, organic shape.

CAPITULO II

MARCO TEÓRICO

2.1. INTRODUCCIÓN

En este capítulo se expone los fundamentos teóricos, como ser conceptos, estableceremos metodologías, técnicas, métodos, y herramientas a usar durante el desarrollo de sistema, en el presente proyecto.

2.2. SISTEMA

Un sistema es un conjunto de partes o elementos organizados y relacionados que interactúan entre sí para lograr un objetivo. Los sistemas reciben datos, energía o materia del ambiente (entrada) y proveen información, energía o materia (salida). Cada sistema puede ser estudiado con el objetivo de comprender el funcionamiento del mismo, descubrir sus límites/fronteras visibles y/o no visibles, entender el objetivo del mismo y cómo interactúa con otros sistemas externos. (Alegsa, 2018)

Es un conjunto de elementos relacionados entre sí que contribuyen a un determinado fin. Donde cada elemento tiene su propia función y en conjunto generan una nueva función. Se caracteriza: Elementos que interactúan jerárquicamente, función de cada elemento, ramas, finalidad y fruto de la creación. (Dip, 2011)

Sistema es un conjunto de elementos interrelacionados entre sí, conformados por un bloque de entrada, uno de proceso y uno de salida, que cuenta con una línea retroalimentación; todo ello operando dentro de un determinado entorno. (Ramirez, 2019)

2.3. DATO

En informática, los datos son representaciones simbólicas (vale decir: numéricas, alfabéticas, algorítmicas, etc.) de un determinado atributo o variable cualitativa o cuantitativa, o sea: la descripción codificada de un hecho empírico, un suceso, una entidad.

Los datos son, así, la información (valores o referentes) que recibe el computador a través de distintos medios, y que es manipulada mediante el procesamiento de los algoritmos de programación. Su contenido puede ser prácticamente cualquiera: estadísticas, números, descriptores, que por separado no tienen relevancia para los usuarios del sistema, pero que en conjunto pueden ser interpretados para obtener una información completa y específica. (Raffino M. E., 2020)

En informática el concepto de dato se refiere a las magnitudes numéricas, conjunto de símbolos, frases, valores cualitativos, imágenes, sonidos, entre otros, que son procesados por diversos sistemas o mecanismos informáticos a fin de generar una información de manera automática y eficiente. (Morales, 2019)

2.4. INFORMACIÓN

En informática se entiende por información al conjunto de datos ordenados, secuenciados, procesados por un algoritmo de programación, que permiten recomponer un referente, como un hecho concreto o algún sentido real.

La recuperación de la información a partir de los paquetes o conjuntos de datos es, así, el objetivo final de la computación, dado que los sistemas informáticos codifican y representan la información a través de distintos mecanismos y lenguajes que les permiten comunicarse entre sí de manera veloz y eficiente. (Raffino M. E., 2020)

2.5. FACTURACIÓN

La factura es un documento legal que constituye y autentifica que se ha prestado o recibido un servicio o se ha comprado o vendido un producto. En la factura se incluyen todos los datos referentes a la operación y, la emisión de la misma, es de obligado cumplimiento en operaciones mercantiles. (Reviso, 2020)

La factura es el documento contable que acredita e informa de la venta o prestación de un producto o servicio. Funciona como una prueba física de la realización de una operación comercial de forma legal. Además, certifica la validez de la operación ya que es un comprobante del pago de la misma. (Vendus, s.f.)

2.6. SISTEMA DE INFORMACIÓN

Por definición es un conjunto de elementos que interactúan entre sí con un fin común; que permite que la información esté disponible para satisfacer las necesidades en una organización, un sistema de información no siempre requiere contar con recursos de computación, aunque la disposición del mismo facilita el manejo e interpretación de la información por los usuarios.

Los elementos que interactúan entre sí son: el equipo computacional (cuando esté disponible), el recurso humano, los datos o información fuente, programas ejecutados por las computadoras, las telecomunicaciones y los procedimientos de políticas y reglas de operación.

Un Sistema de Información realiza cuatro actividades básicas:

- Entrada de información: proceso en el cual el sistema toma datos que requiere.
- Almacenamiento de información: puede hacerse por computadora o archivos físicos para conservar la información.
- Procesamiento de la información: permite la transformación de los datos fuente en información que puede ser utilizada para la toma de decisiones.
- Salida de información: es la capacidad del sistema para producir la información procesada o sacar los datos de entrada al exterior.

Los usuarios de los sistemas de información tienen diferente grado de participación dentro de un sistema y son el elemento principal que lo integra, así se puede definir usuarios primarios quienes alimentan el sistema, usuarios indirectos que se benefician de los resultados pero que no interactúan con el sistema, usuarios gerenciales y directivos quienes tienen responsabilidad administrativa y de toma de decisiones con base a la información que produce el sistema. (El Instituto de Nutrición de Centro América y Panamá, s.f.)

Un sistema de información se alimenta de datos, que son su materia prima para procesarla y transformarla en información, la cual es útil para cualquier empresa en la toma de decisiones. Los datos por sí solos no nos dicen nada, ya que son datos aislados sin ningún tipo de conexión. (Ramirez, 2019)

2.7. CONTROL

Control es el proceso de verificar el desempeño de distintas áreas o funciones de una organización. Usualmente implica una comparación entre un rendimiento esperado y un rendimiento observado, para verificar si se están cumpliendo los objetivos de forma eficiente y eficaz. El control permite tomar acciones correctivas cuando sea necesario.

El control es una de las principales actividades administrativas de las organizaciones. El control se relaciona con la planeación, porque el control busca que el desempeño se ajuste a los planes. El proceso administrativo, desde el punto de vista tradicional, es un proceso circular que se retroalimenta.

El control se ejerce en todos los niveles de las organizaciones desde los niveles superiores o jerárquicos, hasta los niveles inferiores u operativos. (Zona Económica, 2019)

2.7.1. Control de gestión

El control de gestión se refiere a la evaluación de las actividades administrativas que se realizan en los distintos niveles. Se enfoca en los niveles superiores y medios de la organización. Puede incluir elementos de largo plazo, como la estrategia de inversiones o los procesos de planificación.

2.7.2. Control de financiero

El control financiero tiene en cuenta información financiera y contable, como los ratios financieros, para evaluar la capacidad de la organización de pagar sus deudas y disponer de recursos financieros adecuados para la operatoria de la empresa.

2.7.3. Control de operaciones

El control de operaciones verifica que las actividades principales de la empresa se estén desarrollando de acuerdo a lo planeado. Se concentra en los niveles inferiores y medio de la organización, y en el corto plazo. Suele estar estandarizado, es decir, que las observaciones o mediciones se realizan periódicamente (en forma horaria, diaria, semanal, etc.).

2.7.4. Control de calidad

El control de calidad es un proceso integral que busca lograr que la actividad principal de la empresa cumpla con ciertos parámetros de calidad. El control de calidad no sólo verifica la calidad del producto o servicio terminado, sino que también controla los procesos intermedios y las materias primas. Un aspecto importante del control de calidad es que también tiene en cuenta a las personas involucradas en la producción. Aspectos como la capacitación de los recursos humanos, su entrenamiento y el ambiente de trabajo impactan en la calidad. (Zona Económica, 2019)

2.8. SUMINISTRO

La definición de suministros está vinculada con la acción de suministrar, de proveer a alguien de algo que necesita. En términos económicos se utiliza el concepto de suministros como sinónimo de abastecimiento.

Consiste en una actividad que se desarrolla con el propósito de satisfacer las necesidades de consumo de una estructura económica, ya sea empresa, familia, etc. Dicho suministro debe efectuarse en tiempo y forma.

Existen industrias y empresas que se ocupan de suministrar una serie de artículos y servicios. El concepto de suministro no sólo afecta a los alimentos, aunque sean los más demandados y necesitados por parte de los usuarios, ya que existe una amplia variedad de productos de uso concreto en un sector o de demanda general que también se suministran a los distintos espacios habilitados para que puedan llegar al consumidor final. (Economíasimple.net, 2016)

2.9. INGENIERÍA WEB

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

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.

Desde que esto empezó a suceder el Internet se volvió más que una diversión y empezó a ser tomado más en serio, ya que el aumento de publicaciones y de informaciones hizo que la Web se volviera como un desafío para los (Ingeniería del software) ingenieros del software, a raíz de esto se crearon enfoques disciplinados, sistemáticos y metodologías donde tuvieron en cuenta aspectos específicos de este nuevo medio.

Uno de los aspectos más tenidos en cuenta, en el desarrollo de sitios web es sin duda alguna el diseño gráfico y la organización estructural del contenido. En la actualidad la web está sufriendo grandes cambios, que han obligado a expertos en el tema a utilizar herramientas y técnicas basadas en la ingeniería del software, para poder garantizar el buen funcionamiento y administración de los sitios web.

Para garantizar el buen funcionamiento y mantenimiento de los sitios web, este debe contar con ciertos atributos y características que en conjunto forman un concepto muy importante, para alcanzar el éxito en cualquier organización, herramienta, y todo aquello que se pueda considerar como servicio. Dicho concepto es la calidad, que con atributos como, usabilidad, navegabilidad, seguridad, mantenibilidad, entre otros, hace posible por un lado la eficiencia del artefacto web y por ende la satisfacción del usuario final.

Pero para tener artefactos de calidad, a esa misma se le debe planificar, programar y controlar, es decir la calidad no podrá ser agregada a un artefacto web o a cualquier otro producto, al final del proceso de desarrollo, si no que se deberá implementar durante todo el ciclo de vida del desarrollo. Para finalizar el resultado de un proceso de calidad, podría arrojar recomendaciones para introducir mejoras, y la decisión final podría consistir en lanzar una nueva versión del sitio web o en modificar algunos atributos ausentes o pobremente diseñados.

Cabe destacar que la ingeniería de la web hace una diferencia entre un sitio web y un aplicativo, ya que la ingeniería de la web no se dedica a la construcción de sitios web si no a la construcción de aplicativos web, la principal característica que los distingue (aplicativos de sitios web) es que los sitios web son sitios en la web en donde se publica contenido generalmente estático o un muy bajo nivel de interactividad con el usuario, mientras que los aplicativos son lugares con alto contenido de interactividad y funcionalidades que bien podrían ser de un software convencional, el aplicativo web más sencillo seria uno que contenga formularios y subiendo de nivel encontramos los que realizan conexión con bases de datos remotas, y administradores de contenidos entre otras.

Entonces la ingeniería de la Web es la aplicación de metodologías sistemáticas, disciplinadas y cuantificables al desarrollo eficiente, operación y evolución de aplicaciones de alta calidad en la World Wide Web. En este sentido, la ingeniería de la Web hace referencia a las metodologías, técnicas y herramientas que se utilizan en el desarrollo de aplicaciones Web complejas y de gran dimensión en las que se apoya la evaluación, diseño, desarrollo, implementación y evolución de dichas aplicaciones. (Lopez, 2010)

2.10. SOFTWARE

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

Muchas personas asocian el término software con los programas de computadora. Sin embargo, yo prefiero una definición más amplia donde el software no son solo programas, sino todos los documentos asociados y la configuración de datos que se necesitan para hacer que estos programas operen de manera correcta. Por lo general, un sistema de software consiste en diversos programas independientes, archivos de configuración que se utilizan para ejecutar estos programas, un sistema de documentación que describe la estructura del sistema, la documentación para el usuario que explica cómo utilizar el sistema y sitios web que permitan a los usuarios descargar la información de productos recientes. (Sommerville, 2005)

2.11. INGENIERÍA DE SOFTWARE

La ingeniería de software es una especialidad que consiste en sistemas, instrumentos y técnicas que se emplean en el desarrollo de los programas informáticos.

La ingeniería de software, también, incorpora el análisis precedente de la situación, el bosquejo del proyecto, el desarrollo del software, el ensayo necesario para comprobar su funcionamiento correcto y poner en funcionamiento el sistema.

Se debe señalar, que el desarrollo del software va unido a lo que se conoce en el campo del software “ciclo de vida del software” que consiste en cuatro etapas que se conocen como: concepción, elaboración, construcción y transición.

La concepción determina la repercusión del proyecto y diseña el modelo de negocio; la elaboración precisa la planificación del proyecto, especificando las características y apoya la arquitectura; la construcción es la elaboración del producto; y la transición es la entrega del producto terminado a los usuarios.

Al culminar este ciclo, comienza el mantenimiento del software, el cual consiste en una etapa en la que el software ofrece soluciones a errores que son denunciados por los usuarios, principalmente y se incorporan actualizaciones para hacer frente a los nuevos requisitos. (Mi carrera universitaria, s.f.)

2.11.1. Modelos para el desarrollo de software

Un modelo para el desarrollo de software es una representación abstracta de un proceso. Cada modelo representa un proceso desde una perspectiva particular y así proporcione información parcial sobre el proceso. Éstos modelos generales no son descripciones definitivas de los procesos del software más bien son abstracciones de los procesos que se pueden utilizar para el desarrollo del software. Puede pensarse en ellos como marcos de trabajo del proceso y que pueden ser adaptados para crear procesos más específicos. Los modelos que mencionaremos en este punto son:

- a) **El modelo en cascada.** Considera las actividades fundamentales del proceso especificación, desarrollo, validación y evolución. Los representa como fases separadas del proceso, tales como la especificación de requerimientos, el diseño del software, la implementación, las pruebas, etcétera.

- b) **El modelo de desarrollo evolutivo (espiral).** Este enfoque entrelaza las actividades especificación, desarrollo y validación. Es decir, surge de un sistema inicial que se desarrolla rápidamente a partir de especificaciones abstractas. Basándose en las peticiones del cliente para producir un sistema que satisfaga sus necesidades.
- c) **El modelo de desarrollo basado en componentes.** Éste enfoque se basa en la existencia de un número significativo de componentes reutilizables. El proceso de desarrollo se enfoca en integrar estos componentes en el sistema más que en desarrollarlos desde cero. Estos tres modelos se utilizan ampliamente en la práctica actual de la ingeniería del software, no se excluyen mutuamente y a menudo se utilizan juntos especialmente para el desarrollo de grandes sistemas.

2.11.1.1. El modelo de desarrollo evolutivo (espiral)

El modelo en espiral que Boehm propuso es un modelo de proceso de software evolutivo que conjuga la naturaleza iterativa de la construcción de prototipos con los aspectos controlados y sistemáticos del modelo en cascada. Cuando se aplica este modelo en espiral, el software se desarrolla en una serie de entregas evolutivas. Cada una de las actividades del marco de trabajo representan un segmento de la ruta en espiral.

Este modelo se basa en la idea de desarrollar una implementación inicial, exponiéndola a los comentarios del usuario y refinándola a través de las diferentes versiones que se generan hasta que se desarrolle un sistema adecuado.

Las actividades de especificación, desarrollo y validación se entrelazan en vez de separarse, con una rápida retroalimentación entre estas. Existen dos tipos de desarrollo evolutivo:

1. Desarrollo exploratorio, en este caso el objetivo del proceso es trabajar con el cliente para explorar sus requerimientos y entregar un sistema final. El desarrollo empieza con las partes del sistema que se comprenden mejor. El sistema evoluciona agregando nuevos atributos propuestos por el cliente.
2. Prototipos desechables, el objetivo de este proceso de desarrollo evolutivo es comprender los requerimientos del cliente para así desarrollar una definición mejorada de los requerimientos para el sistema. El prototipo se centra en experimentar los requerimientos del cliente que no se comprenden del todo.

Haciendo referencia a la producción del software, un enfoque evolutivo suele ser más efectivo que el enfoque en cascada, ya que satisface las necesidades inmediatas de los clientes. La ventaja de un software que se basa en un enfoque evolutivo es que las especificaciones se pueden desarrollar de forma creciente. Tan pronto como los usuarios desarrollen un mejor entendimiento de su problema, esto se puede reflejar en el software. (Cendejas Valdez, 2014)

2.11.1.2. Arquitectura Cliente – Servidor

La estructura cliente - servidor es una arquitectura de computación en la que se consigue un procesamiento cooperativo de la información por medio de un conjunto de procesadores, de tal forma que uno o varios clientes, distribuidos geográficamente o no, solicitan servicios de computación a uno o más servidores. (Gonzales Reyes, Quijije Lucas, Betty Carvajal, Manrique Neira, & Quijije Toro, 2017)

La arquitectura cliente servidor tiene dos partes claramente diferenciadas, por un lado, la parte del servidor y por otro la parte de cliente o grupo de clientes donde lo habitual es que un servidor sea una máquina bastante potente con un hardware y software específico que actúa de depósito de datos y funcione como un sistema gestor de base de datos o aplicaciones.

En esta arquitectura el cliente suele ser estaciones de trabajo que solicitan varios servicios al servidor, mientras que un servidor es una máquina que actúa como depósito de datos y funciona como un sistema gestor de base de datos, este se encarga de dar la respuesta demandada por el cliente. (Schiaffarino, 2019)

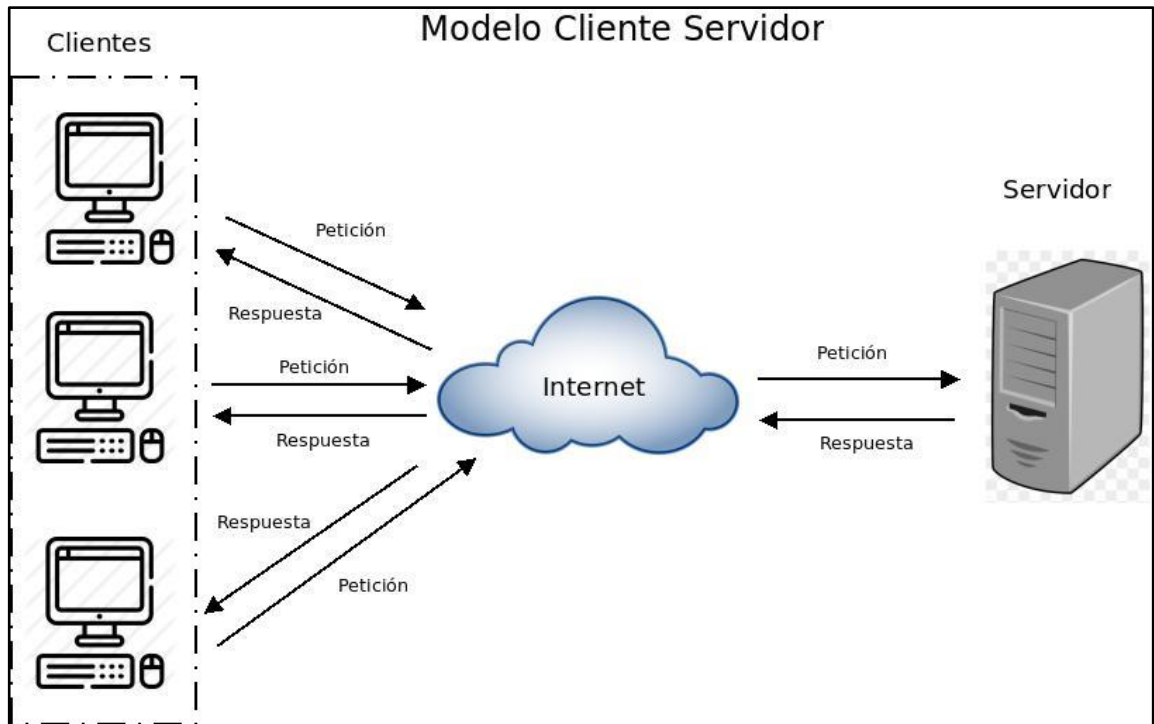


Figura 2.1. Esquema del Modelo Cliente Servidor
Fuente: (Schiaffarino, 2019)

2.12. METODOLOGÍA DE INGENIERÍA WEB

2.12.1. Metodología uwe

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

Es una herramienta que nos permitirá modelar aplicaciones web, utilizada en la ingeniería web, prestando especial atención en sistematización y personalización (sistemas adaptativos). UWE es una propuesta basada en el proceso unificado y UML pero adaptados a la web. En requisitos separa las fases de captura, definición y validación. Hace además una clasificación y un tratamiento especial dependiendo del carácter de cada requisito.

En el marco de UWE es necesario la definición de un perfil UML (extensión) basado en estereotipos con este perfil se logra la asociación de una semántica distinta a los diagramas del UML puro, con el propósito de acoplar el UML a un dominio específico, en este caso, las aplicaciones Web. Entre los principales modelos de UWE podemos citar: el modelo lógico-conceptual, modelo navegacional, modelo de presentación, visualización de Escenarios Web y la interacción temporal, entre los diagramas: diagramas de estado, secuencia, colaboración y actividad.

UWE define vistas especiales representadas gráficamente por diagramas en UML. Además, UWE no limita el número de vistas posibles de una aplicación, UML proporciona mecanismos de extensión basados en estereotipos.

Estos mecanismos de extensión son los que UWE utiliza para definir estereotipos que son lo que finalmente se utilizarán en las vistas especiales para el modelado de aplicaciones Web. De esta manera, se obtiene una notación UML adecuada a un dominio en específico a la cual se le conoce como Perfil UML.

UWE está especializada en la especificación de aplicaciones adaptativas, y por tanto hace especial hincapié en características de personalización, como es la definición de un modelo de usuario o una etapa de definición de características adaptativas de la navegación en función de las preferencias, conocimiento o tareas de usuario.

Además de estar considerado como una extensión del estándar UML, también se basa en otros estándares como, por ejemplo: XMI como modelo de intercambio de formato, MOF para la meta-modelado, los principios de modelado de MDA, el modelo de transformación del lenguaje QVT y XML. (Ruiz Andrade & Rosales Terán, 2016)

2.12.2. Características de UWE

Las principales características de la metodología UWE refieren a que:

- Provee un lenguaje completamente basado en UML.
- Se basa en los sistemas personalizados y en la sistematización.
- Es una metodología que añade modelos utilizados para representar aplicaciones web.
- Proporciona una herramienta para generar aplicaciones web semiautomática.

2.12.3. Actividades de modelado de UWE

Las actividades base de modelado de UWE son el análisis de requerimientos, el modelo conceptual, el modelo de navegación y el modelo de presentación.

A estos modelos se pueden sumar otros modelos como lo son el modelo de interacción y la visualización de Escenarios Web.

2.12.4. Etapas de modelado

El modelo que propone UWE está compuesto por etapas o sub-modelos:

- Modelo de Requerimientos
- Modelo de Contenido
- Modelo de Navegación
- Modelo de Presentación
- Modelo de Procesos

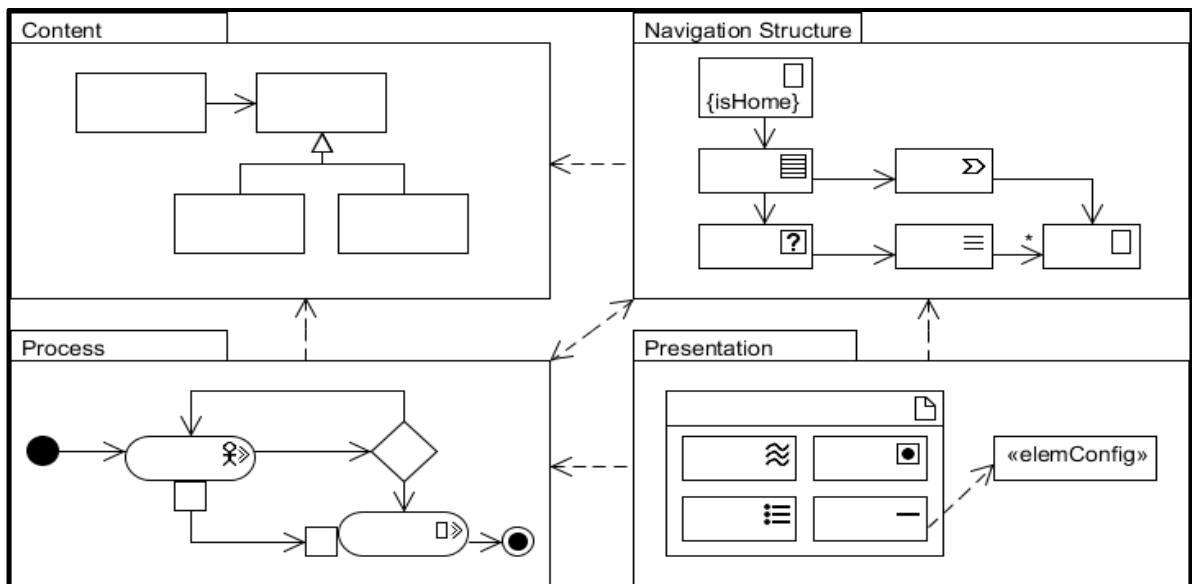


Figura 2.2. Descripción general de los modelos UWE
Fuente: (Ingeniería Web, 2012)

2.12.4.1. Modelo de requerimientos

En UWE el modelado de requerimientos consta de dos partes:

- Casos de uso de la aplicación y sus relaciones.
- Actividades que describen los casos en detalle.

Los requerimientos pueden ser documentados en diferentes niveles de detalle, para este caso, UWE propone dos niveles de granularidad. En primera instancia se deben describir detalladamente las funcionalidades del sistema, las cuales son modeladas con casos de uso UML. Como segundo paso, se debe elaborar una descripción de los casos de uso más detallada, por ejemplo, realizando diagramas de actividad UML donde se limiten las responsabilidades y acciones de los actores involucrados.

Los **casos de uso** fueron propuestos por el Proceso de Desarrollo de Software Unificado (RUP) para capturar los requerimientos del sistema. Es una técnica centrada en el usuario que obliga a definir quiénes son los usuarios (actores) de la aplicación y ofrece una forma intuitiva de representar la funcionalidad que una aplicación tiene que cumplir para cada actor.

En UWE, los casos de uso se distinguen por los estereotipos «navegación» y «procesamiento» para indicar si una aplicación modifica los datos persistentes de la aplicación o no. "SearchContact", por ejemplo, modela la búsqueda de contactos y tiene el estereotipo de "búsqueda" porque durante una búsqueda la base de datos solo se lee y se presenta al usuario. Los otros casos de uso, sin embargo, modelan cambios en los datos y, por lo tanto, se escriben con "procesamiento".

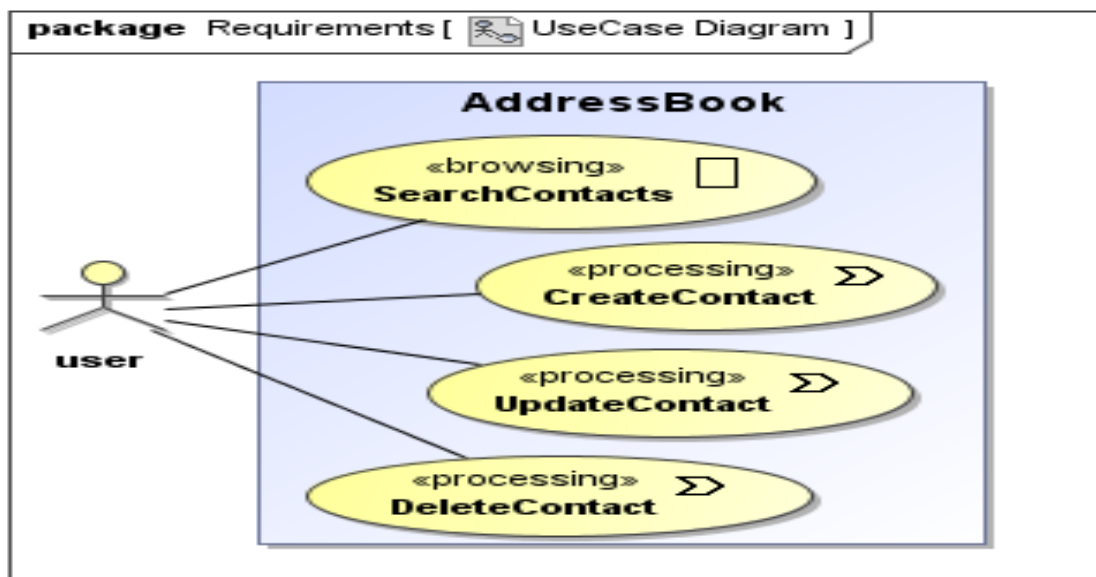


Figura 2.3. Diagrama de Caso de Uso de la Metodología UWE
Fuente: (Ludwing-Maximilians, 2016)

2.12.4.2. Modelo de contenido

El propósito del modelo de contenido es proporcionar una especificación visual de la información relevante para el dominio del sistema web, que comprende principalmente el contenido de la aplicación Web.

En él se representa la información del dominio, sus datos persistentes, mediante un diagrama de clases UML, donde se puede observar las distintas clases que forman parte del sistema con sus respectivos atributos y estereotipos definido por UWE.

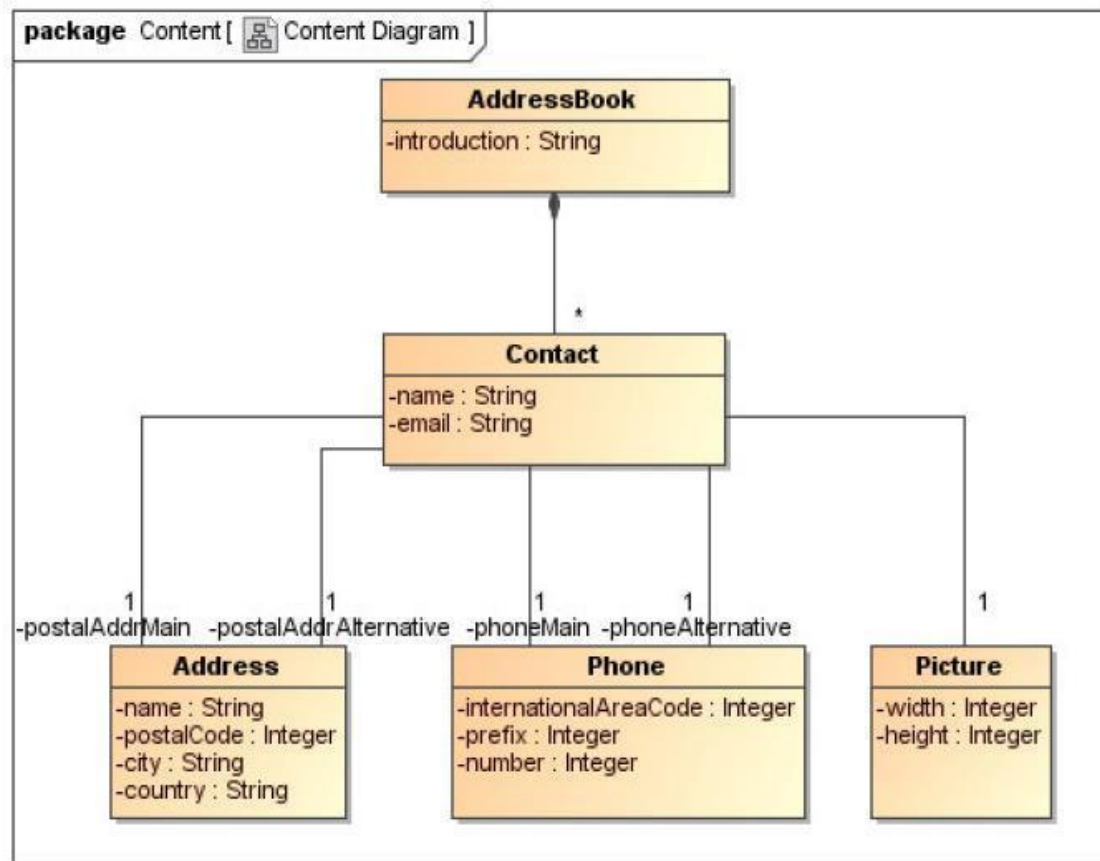




Figura 2.4. Diagrama de Contenido UWE
Fuente: (Ludwing-Maximilians, 2016)

2.12.4.3. Modelo de navegación

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

Los elementos de modelado son:

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

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

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








NOMBRES DE ESTEREOTIPOS Y SUS ICONOS	
 clase de navegación	 menú
 índice	 pregunta
 visita guiada	 clase de proceso
 nodo externo	

Figura 2.5. Estereotipos e Iconos de Diagrama de Navegación

Fuente: (Ludwing-Maximilians, 2016)

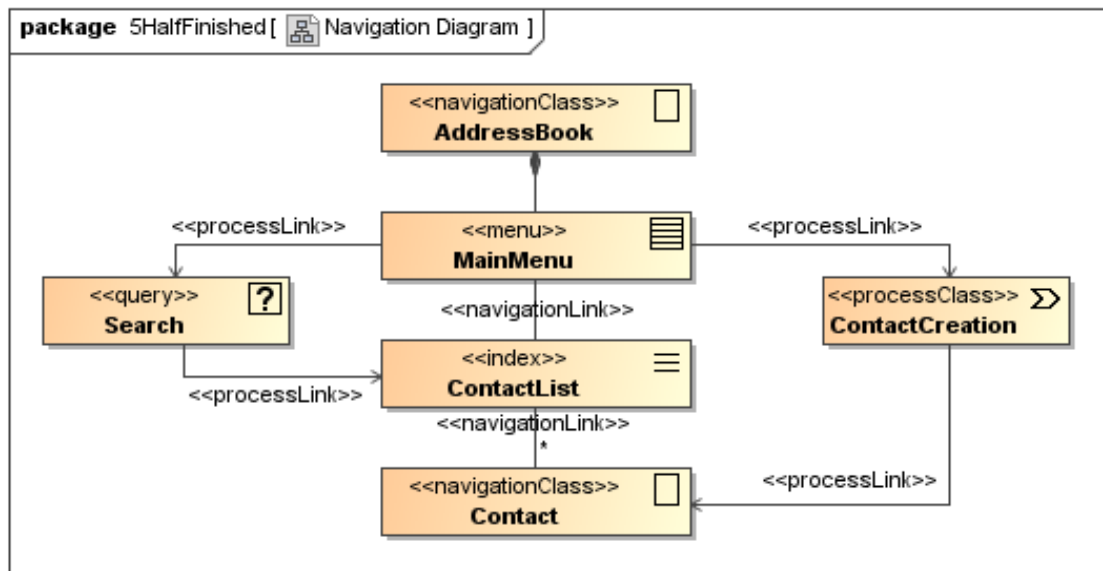


Figura 2.6. Diagrama de Navegación UWE
Fuente: (Ludwing-Maximilians, 2016)

2.12.4.4. Modelo de presentación

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

Los elementos básicos del modelo de presentación son:

- **Clases de presentación**, las cuales se basan directamente en los nodos del modelo de navegación. Una clase de presentación () está compuesta por elementos de UI tales como, texto (<<text>>), vinculo (<<anchor>>), botón (<<button>>), imagen (<<image>>), formulario (<<form>>), y colección de vínculos (<<anchored collection>>)
- **Páginas web** (<<page>>), que se utilizan para modelar la información proveniente de varios nodos de navegación y que se presentan en una misma página web.

- **Grupo de presentación** (<<presentation group>>), el cual es un contenedor de clases de presentación, y a su vez de otros grupos de presentación.

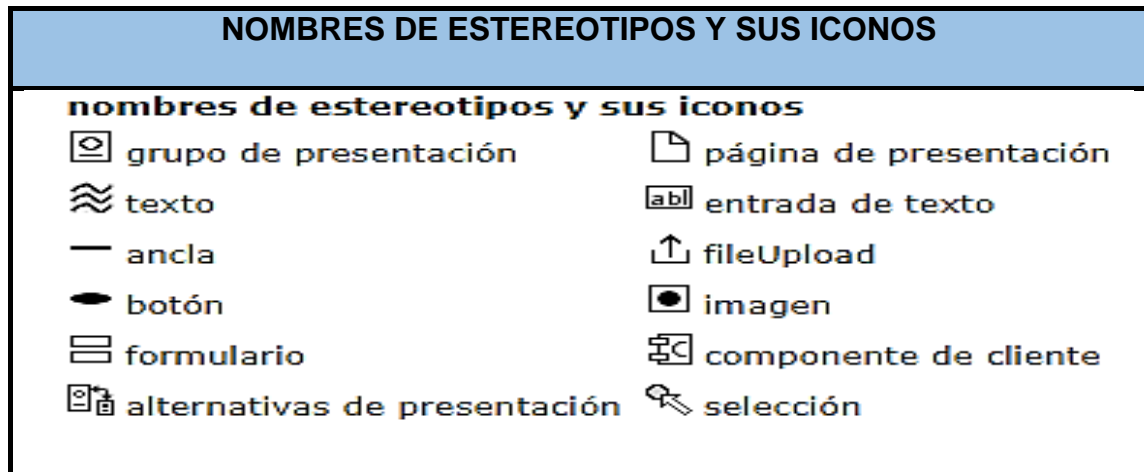


Figura 2.7. Estereotipos e Iconos de Diagrama de Presentación

Fuente: (Ludwing-Maximilians, 2016)

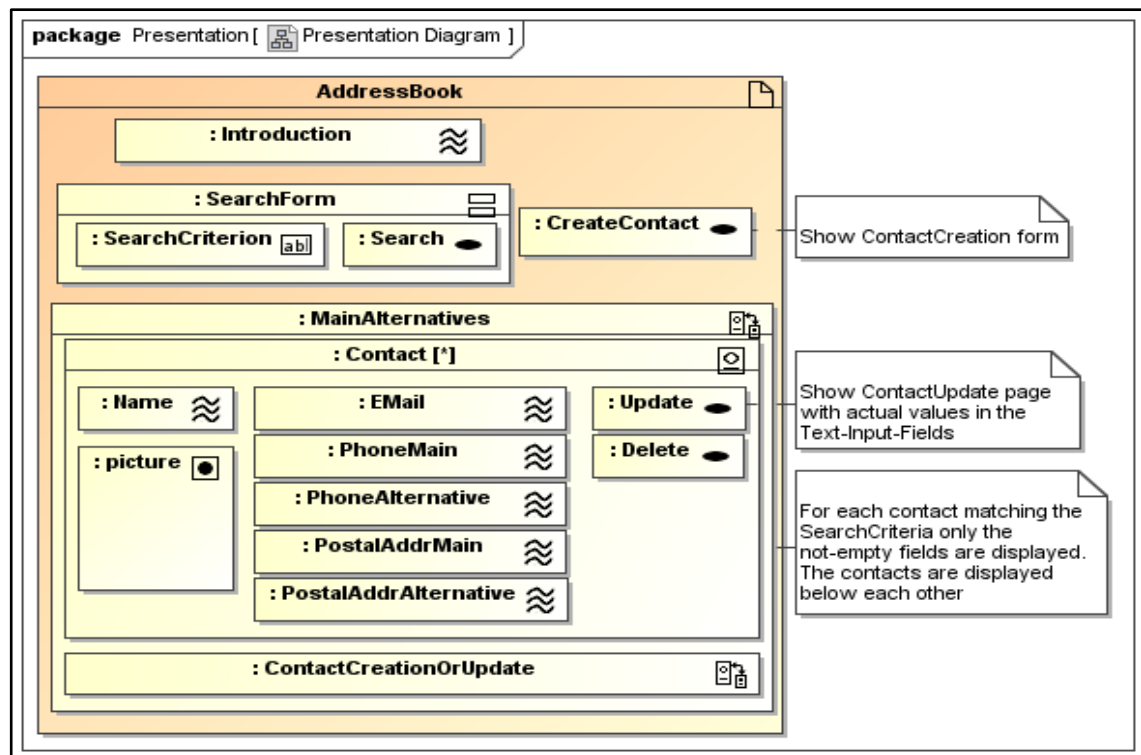


Figura 2.8. Diagrama de Presentación UWE

Fuente: (Ludwing-Maximilians, 2016)

2.12.4.5. Modelo de procesos

Representa los aspectos dinámicos de la aplicación Web y especifica funcionalidad cómo 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.

El Modelo de Proceso comprende:

- el Modelo de Estructura del Proceso que describe las relaciones entre las diferentes clases de proceso y
- el Modelo de Flujo del Proceso que especifica las actividades conectadas con cada «processClass».

2.12.5. Fases de la metodología UWE

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

Las fases o etapas a utilizar son:

2.12.5.1. Captura, análisis y especificación de requisito

En simple palabras y básicamente, durante esta fase, se adquieren, reúnen y especifican las características funcionales y no funcionales que deberá cumplir la aplicación web.

Trata de diferente forma las necesidades de información, las necesidades de navegación, las necesidades de adaptación y las de interfaz de usuario, así como algunos requisitos adicionales. Centra el trabajo en el estudio de los casos de uso, la generación de los glosarios y el prototipado de la interfaz de usuario.

2.12.5.2. Diseño del sistema

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

2.12.5.3. Codificación del software

Durante esta etapa se realizan las tareas que comúnmente se conocen como programación; que consiste, esencialmente, en llevar a código fuente, en el lenguaje de programación elegido, todo lo diseñado en la fase anterior.

2.12.5.4. Pruebas

Las pruebas se utilizan para asegurar el correcto funcionamiento de secciones de código.

2.12.5.5. La Instalación o Fase de Implementación

Es el proceso por el cual los programas desarrollados son transferidos apropiadamente al computador destino, inicializados, y, eventualmente, configurados; todo ello con el propósito de ser ya utilizados por el usuario final. Esto incluye la implementación de la arquitectura, de la estructura del hiperespacio, del modelo de usuario, de la interfaz de usuario, de los mecanismos adaptativos y las tareas referentes a la integración de todas estas implementaciones.

2.12.5.6. El Mantenimiento

Es el proceso de control, mejora y optimización del software ya desarrollado e instalado, que también incluye depuración de errores y defectos que puedan haberse filtrado de la fase de pruebas de control. (Galeano, 2012)

2.13. HERRAMIENTAS DE DESARROLLO DE SOFTWARE

2.13.1. Servidor Apache

El servidor Apache HTTP, también llamado Apache, es un servidor web HTTP de código abierto para la creación de páginas y servicios web. Es un servidor multiplataforma, gratuito, muy robusto y que destaca por su seguridad y rendimiento. (Fumas Cases, 2014)

El servidor HTTP Apache es un servidor web HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual.

El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation.

Apache es altamente configurable, admite bases de datos de autenticación y negociado de contenido, aunque carece de una interfaz gráfica que ayude en su configuración.

Apache es una aplicación que permite montar un servidor web en cualquier equipo y casi cualquier sistema operativo. Al contrario que IIS (Internet Information Server) que sólo funciona en sistemas operativos de Microsoft.

Apache soporta PHP como lenguaje de programación. Con los módulos adecuados, Apache puede soportar también ASP. (Pérez Díaz, s.f.)

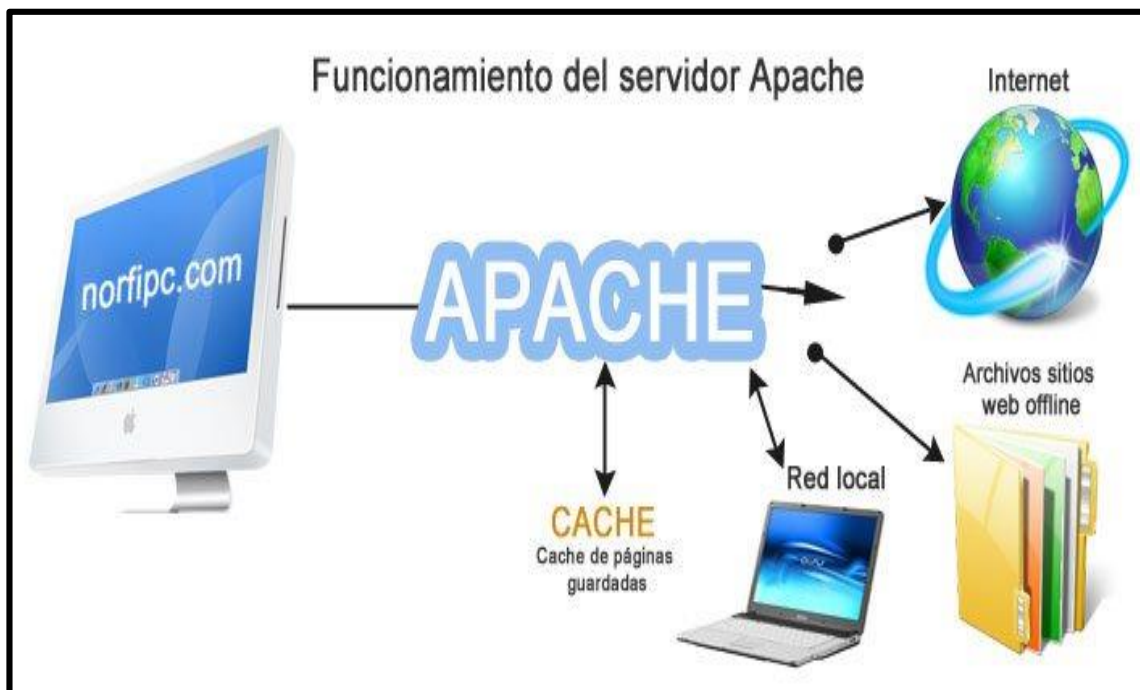


Figura 2.9. Servidor Apache
Fuente: (Carrodeguas, 2019)

2.13.2. Base de Datos

Se define una base de datos como una serie de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por los sistemas de información de una empresa o negocio en particular.

Entre las principales características de los sistemas de base de datos podemos mencionar:

- Independencia lógica y física de los datos.
- Redundancia mínima.
- Acceso concurrente por parte de múltiples usuarios.
- Integridad de los datos.
- Consultas complejas optimizadas.
- Seguridad de acceso y auditoría.
- Respaldo y recuperación.
- Acceso a través de lenguajes de programación estándar. (Perez, 2007)

2.13.3. Gestor de base de datos MySql

MySQL es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual: Licencia pública general/Licencia comercial por Oracle Corporation y está considerada como la base de datos de código abierto más popular del mundo, y una de las más populares en general junto a Oracle y Microsoft SQL Server, sobre todo para entornos de desarrollo web.

MySQL es un sistema de administración relacional de bases de datos. Una base de datos relacional archiva datos en tablas separadas en vez de colocar todos los datos en un gran archivo. Esto permite velocidad y flexibilidad. Las tablas están conectadas por relaciones definidas que hacen posible combinar datos de diferentes tablas sobre pedido.

MySQL es software de fuente abierta. Fuente abierta significa que es posible para cualquier persona usarlo y modificarlo. Cualquier persona puede bajar el código fuente de MySQL y usarlo sin pagar. Cualquier interesado puede estudiar el código fuente y ajustarlo a sus necesidades. MySQL usa el GPL (GNU General Public License) para definir qué puede hacer y qué no puede hacer con el software en diferentes situaciones. Si usted no se ajusta al GPL o requiere introducir código MySQL en aplicaciones comerciales, usted puede comprar una versión comercial licenciada. (Wikipedia, 2019)

2.13.4. MySQL Workbench

MySQL Workbench es un software creado por la empresa Sun Microsystems, esta herramienta permite modelar diagramas de Entidad-Relación para bases de datos MySQL.

Con esta herramienta se puede elaborar una representación visual de las tablas, vistas, procedimientos almacenados y claves foráneas de la base de datos. Además, es capaz de sincronizar el modelo en desarrollo con la base de datos real. Se puede realizar una ingeniería directa e ingeniería inversa para exportare e importar el esquema de una base de datos ya existente el cual haya sido guardado o hecho copia de seguridad con MySQL Administrador. (Aranibar, Tinoco, & Ibarra, 2011)

2.13.5. Lenguaje de programación “PHP”

PHP es un lenguaje de programación de uso general de código del lado del servidor. Originalmente fue diseñado para el desarrollo web de contenido dinámico. Está actualmente entre los proyectos de código abierto más populares (gracias en parte a la similitud de su sintaxis con el lenguaje C). El código es interpretado por un servidor web con un módulo procesador de PHP que genera la página Web resultante.

Básicamente, PHP permite a páginas estáticas convertirse en dinámicas (aunque es mucho más que eso). El nombre "PHP" es un acrónimo que significa "PHP: Hypertext Preprocessor", en español "PHP: Preprocesador de hipertexto". La palabra "hipertexto" significa que PHP realiza cambios antes de que el contenido (como la página HTML) sea creado. (WIKILIBROS, 2017)

PHP es un lenguaje de código abierto muy popular, adecuado para desarrollo web y que puede ser incrustado en HTML. Es popular porque un gran número de páginas y portales web están creadas con PHP. Código abierto significa que es de uso libre y gratuito para todos los programadores que quieran usarlo. Incrustado en HTML significa que en un mismo archivo vamos a poder combinar código PHP con código HTML, siguiendo unas reglas. (Gonzales Gutierrez)

2.13.6. HTML

HTML (HyperText Markup Language o lenguaje de marcado de hipertexto) es el lenguaje de etiquetas que funciona como una de las piedras angulares de la World Wide Web. Aunque la evolución de Internet nos ha traído muchos avances en lo que se refiere a tecnología (Web 2.0 y Web 3.0), el lenguaje de etiquetas que se popularizó en la década del noventa sigue siendo fundamental para el desarrollo web, ya que es el que comprenden e interpretan los navegadores. Claro está que por sí solo ya no es tan potente como lo fue en aquellos tiempos y, hoy por hoy, necesita combinarse con otras tecnologías y lenguajes para lograr resultados que estén a la altura de las necesidades del desarrollo web actual. (De Luca D. , 2011)

2.13.7. CSS3

Mientras que HTML nos permite definir la estructura una página web, las hojas de estilo en cascada (Cascading Style Sheets o CSS) son las que nos ofrecen la posibilidad de definir las reglas y estilos de representación en diferentes dispositivos, ya sean pantallas de equipos de escritorio, portátiles, móviles, impresoras u otros dispositivos capaces de mostrar contenidos web.

A partir del año 2005 se comenzó a definir el sucesor de esta versión, al cual se lo conoce como CSS3 o Cascading Style Sheets Level 3. Actualmente en definición, esta versión nos ofrece una gran variedad de opciones muy importantes para las necesidades del diseño web actual. Desde opciones de sombreado y redondeado, hasta funciones avanzadas de movimiento y transformación, CSS3 es el estándar que dominará la web por los siguientes años. (De Luca D. , 2012)

2.13.8. Javascript

Javascript es un lenguaje de programación que surgió con el objetivo inicial de programar ciertos comportamientos sobre las páginas web, respondiendo a la interacción del usuario y la realización de automatismos sencillos. En ese contexto podríamos decir que nació como un "lenguaje de scripting" del lado del cliente, sin embargo, hoy Javascript es mucho más. Las necesidades de las aplicaciones web modernas y el HTML5 ha provocado que el uso de Javascript que encontramos hoy haya llegado a unos niveles de complejidad y prestaciones tan grandes como otros lenguajes de primer nivel. (Desarrolloweb.com, s.f.)

Se trata de un lenguaje de programación que permite realizar acciones de diversos grados de complejidad en sitios web sin necesidad de compilación. El hecho de que los mismos navegadores lean y asimilen el código para efectuar las instrucciones indicadas por éste, ha convertido a JavaScript en un lenguaje de programación muy utilizado y apreciado por los desarrolladores. (HACK A BOSS, 2019)

2.13.9. Framework Bootstrap

Bootstrap, es un framework originalmente creado por Twitter, que permite crear interfaces web con CSS y JavaScript, cuya particularidad es la de adaptar la interfaz del sitio web al tamaño del dispositivo en que se visualice. Es decir, el sitio web se adapta automáticamente al tamaño de una PC, una Tablet u otro dispositivo. Esta técnica de diseño y desarrollo se conoce como “**responsive design**” o diseño adaptativo.

Aun ofreciendo todas las posibilidades que ofrece Bootstrap a la hora de crear interfaces web, los diseños creados con Bootstrap son simples, limpios e intuitivos, esto le da agilidad a la hora de cargar y al adaptarse a otros dispositivos. El Framework trae varios elementos con estilos predefinidos fáciles de configurar: Botones, Menús desplegables, Formularios incluyendo todos sus elementos e integración jQuery para ofrecer ventanas y tooltips dinámicos. (Solis, 2014)

2.13.10. Framework Codeigniter

Codeigniter es un framework para el desarrollo de aplicaciones en php que utiliza el **MVC**. Permite a los programadores Web mejorar la forma de trabajar y hacerlo a mayor velocidad.

Codeigniter y otros frameworks PHP pueden ayudarte a dar el salto definitivo como desarrollador PHP, creando aplicaciones web más profesionales y con código más reutilizable, con la diferencia que Codeigniter está creado para que sea fácil de instalar en cualquier servidor y de empezar a usar que cualquier otro framework. Además, muchas de sus utilidades y modos de funcionamiento son opcionales, lo que hace que goces de mayor libertad a la hora de desarrollar sitios web.

Algunas otras son las siguientes:

- Es realmente ligero. El núcleo principal utiliza unas pocas librerías, mientras que el resto de librerías se carga solo cuando las necesitamos. Así, la base de este sistema es ligera y rápida.
- Es rápido. De hecho, los desarrolladores de este framework te retan a que encuentres otro framework con mejor desempeño que Codeigniter.
- Utiliza el patrón de diseño **Modelo–Vista–Controlador**.
- El código de Codeigniter es muy limpio y está bien documentado, lo que facilita el poder *extender* este framework de forma sencilla, según nuestras necesidades.

El Modelo Vista Controlador es un estilo de programación en el que la aplicación está dividida en 3 capas:

- **Modelo:** es dónde se procesa y obtienen los datos, la conexión con la bd.
- **Vista:** presenta los datos en pantalla, es donde va el código HTML.
- **Controlador:** controla los datos, dicho de forma rápida obtiene datos de un modelo, los procesa, y se los pasa a la vista.

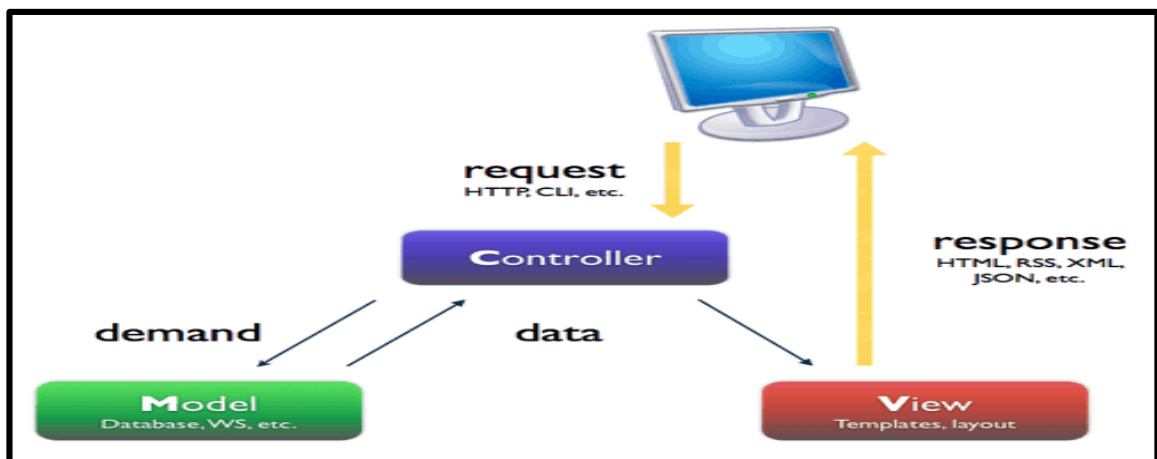


Figura 2.10. MVC de Codeigniter
Fuente: (Fontan, 2012)

2.14. INGENIERÍA DE REDES

Es una disciplina constituyente de la ingeniería que se encarga de administrar, crear, diseñar y gestionar sistemas en tecnología de la información y las telecomunicaciones. Brinda soluciones técnicas a problemas en la transmisión y recepción de señales y la interconexión de redes en cualquier empresa u organización.

Una red de comunicaciones de datos o una red de computadoras es la interconexión de distinto número de sistemas informáticos a través de una serie de dispositivos de telecomunicaciones y un medio físico (alámbrico o inalámbrico). (Raffino M. , Concepto.de, 2019)

Las redes están presentes hoy en casi todos los ámbitos cotidianos, especialmente en los vinculados con la burocracia o con la administración de recursos. De hecho, la conexión a Internet a la que accedemos desde nuestra computadora, teléfono celular u otros dispositivos, no es otra cosa que una inmensa red de computadoras.

Usualmente en las redes informáticas se presentan los siguientes elementos:

- **Servidores.** En una red no siempre los computadores poseen la misma jerarquía o funciones. Los servidores son los que procesan el flujo de datos, atendiendo a todos los demás computadores de la red (“sirviéndolos”, de allí su nombre) y centralizando el control de la red.
- **Clientes o estaciones de trabajo.** Se llama así a los computadores que no son servidores, sino que forman parte de la red y permiten a los usuarios el acceso a la misma, empleando los recursos administrados por el servidor.
- **Medios de transmisión.** Se llama así al cableado o a las ondas electromagnéticas, según sea el caso, que permiten la transmisión de la información.
- **Elementos de hardware.** Aquellas piezas que permiten el establecimiento físico de la red, como son las tarjetas de red en cada computador, los módems y enrutadores que sostienen la transmisión de datos, o las antenas repetidoras que extienden la conexión (en caso de ser inalámbricas).
- **Elementos de software.** Por último, están los programas requeridos para administrar y poner en funcionamiento el hardware de comunicaciones, y que incluye el Sistema Operativo de Redes (NOS, del inglés Network Operating System), el cual además de sostener el funcionamiento de la red le brinda soporte de antivirus y firewall; y los protocolos comunicativos (como los TCP e IP) que permiten a las máquinas “hablar” el mismo idioma. (Raffino M. , Concepto.de, 2020)

2.15. MÉTRICAS DE CALIDAD DE SOFTWARE

En la evaluación del Software es muy importante acatar modelos y estándares actuales de calidad, que detallan las características mediante las cuales se hace la evaluación, las métricas de calidad y las métricas técnicas de calidad del software que son los indicadores. Es necesario tener en cuenta las escalas de medición cualitativa o cuantitativa y las pruebas del software que se ejecutan para verificación de la calidad del producto o proceso.

2.15.1. Calidad de Software

Dentro del contexto de Ingeniería de Software, se tomará la definición de calidad en el software propuesta por la organización internacional de estándares (ISO/IEC DEC 9126): La totalidad de características de un producto de software que tienen como habilidad, satisfacer necesidades explícitas o implícitas.

Según Roger S. Pressman, ingeniero de software, profesor, consultor y autor de productos centrados en la Ingeniería del Software, la calidad de software es la concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados, y con las características implícitas que se espera de todo software desarrollado profesionalmente.

Resumiendo, podemos decir, que la calidad de software se refiere a: “Los factores de un producto de software que contribuyen a la satisfacción completa y total de las necesidades de un usuario u organización”.

2.15.2. La Norma de Evaluación ISO/IEC 9126

Esta norma Internacional fue publicada en 1992, la cual es usada para la evaluación de la calidad de software, llamado “Information technology-Software product evaluation-Quality characteristics and guidelines for their use”; o también conocido como ISO 9126 (o ISO/IEC 9126). Este estándar describe 6 características generales: Funcionalidad, Confiabilidad, Usabilidad, Eficiencia, Mantenibilidad, y Portabilidad.

La norma ISO/IEC 9126 permite especificar y evaluar la calidad del software desde diferentes criterios asociados con adquisición, requerimientos, desarrollo, uso, evaluación, soporte, mantenimiento, aseguramiento de la calidad y auditoría de software. Los modelos de calidad para el software se describen así:

Calidad interna y externa: Especifica 6 características para calidad interna y externa, las cuales, están subdivididas. Estas divisiones se manifiestan externamente cuando el software es usado como parte de un sistema Informático, y son el resultado de atributos internos de software.

Calidad en uso: Calidad en uso es el efecto combinado para el usuario final de las 6 características de la calidad interna y externa del software. Especifica 4 características para la calidad en uso.

Al unir la calidad interna y externa con la calidad en uso se define un modelo de evaluación más completo, se puede pensar que la usabilidad del modelo de calidad externa e interna pueda ser igual al modelo de calidad en uso, pero no, la usabilidad es la forma como los profesionales interpretan o asimilan la funcionabilidad del software y la calidad en uso se puede asumir como la forma que lo asimila o maneja el usuario final. Si se unen los dos modelos, se puede definir que los seis indicadores del primer modelo tienen sus atributos y el modelo de calidad en uso sus 4 indicadores pasarían hacer sus atributos, mirándolo gráficamente quedaría así:

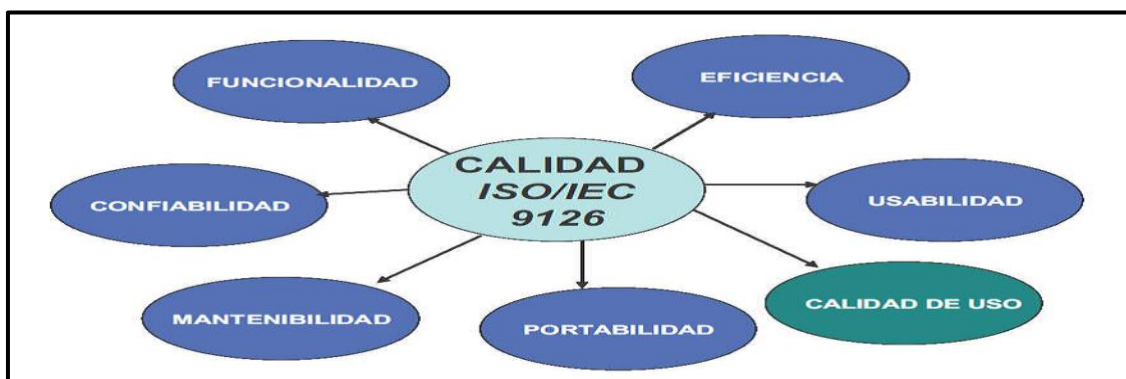


Figura 2.11. Norma de Evaluación ISO/IEC 9126
Fuente: (Borbón, 2013)

Se establecen categorías para las cualidades de la calidad externa e interna y calidad en uso del software, teniendo en cuenta estos 7 indicadores (funcionalidad, confiabilidad, utilidad, eficiencia, capacidad de mantenimiento, portabilidad y calidad en uso), que se subdividen a su vez en varios indicadores; estas se pueden medir por métrica interna o externa.



Figura 2.12. Evaluación Interna, Externa y Calidad de Uso ISO/IEC 9126

Fuente: (Borbón, 2013)

Las definiciones se dan para cada característica y subcaracterística de calidad del software que influye en la calidad. Para cada característica y subcaracterística, la capacidad del software es determinada por un conjunto de atributos internos que pueden ser medidos. Las características y subcaracterísticas se pueden medir externamente por la capacidad del sistema que contiene el software.

2.15.2.1. Funcionalidad

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. A continuación, se muestra la característica de Funcionalidad y las subcaracterísticas que cubre:



Figura 2.13. Característica de Funcionalidad ISO/IEC 9126

Fuente: (Borbón, 2013)

La funcionalidad se divide en 5 criterios:

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.

2.15.2.2. Confiabilidad

La confiabilidad es la capacidad del software para asegurar un nivel de funcionamiento adecuado cuando es utilizando en condiciones específicas. En este caso a la confiabilidad se amplía sostener un nivel especificado de funcionamiento y no una función requerida.

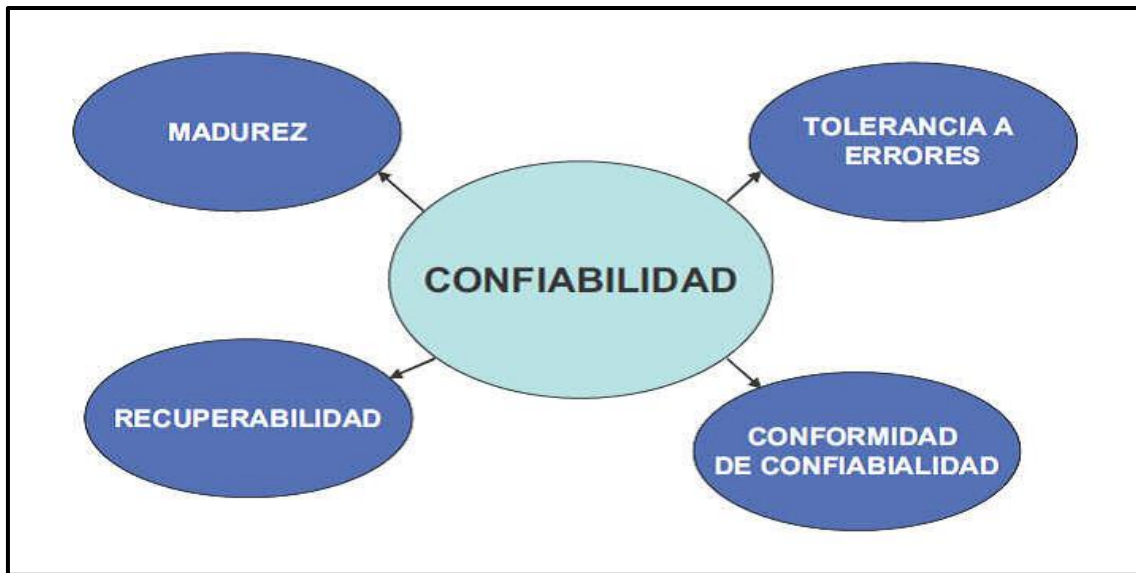


Figura 2.14. Característica de Confiabilidad ISO/IEC 9126

Fuente: (Borbón, 2013)

La confiabilidad se divide en 4 criterios:

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.

2.15.2.3. Usabilidad

La usabilidad es la capacidad del software de ser entendido, aprendido, y usado en forma fácil y atractiva. Algunos criterios de funcionalidad, fiabilidad y eficiencia afectan la usabilidad, pero para los propósitos de la ISO/IEC 9126 ellos no clasifican como usabilidad.

La usabilidad 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.

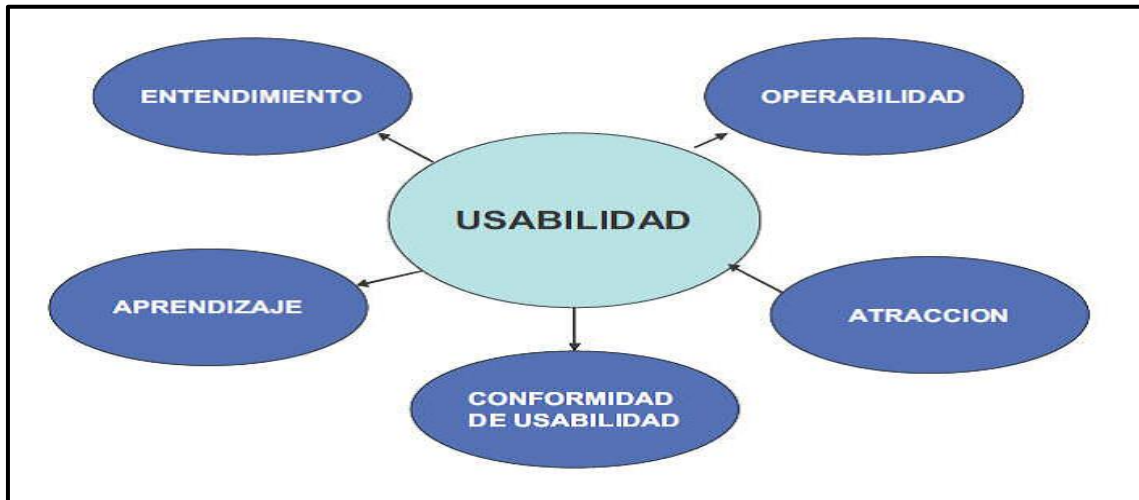


Figura 2.15. Característica de Usabilidad ISO/IEC 9126

Fuente: (Borbón, 2013)

La usabilidad se divide en 5 criterios:

Entendimiento: La capacidad que tiene el software para permitir al usuario entender si es adecuado, y de una manera fácil como ser utilizado para las tareas y las condiciones particulares de la aplicación. En este criterio se debe tener en cuenta la documentación y de las ayudas que el software entrega.

Aprendizaje: La forma como el software permite al usuario aprender su uso. También es importante considerar la documentación.

Operabilidad: La manera como el software permite al usuario operarlo y controlarlo.

Atracción: La presentación del software debe ser atractiva al usuario. Esto se refiere a las cualidades del software para hacer más agradable al usuario, ejemplo, el diseño gráfico.

Conformidad de uso: La capacidad del software de cumplir los estándares o normas relacionadas a su usabilidad.

2.15.2.4. Eficiencia

La eficiencia del software es la forma del desempeño adecuado, de acuerdo a al número recursos utilizados según las condiciones planteadas. Se debe tener en cuenta otros aspectos como la configuración de hardware, el sistema operativo, entre otros.

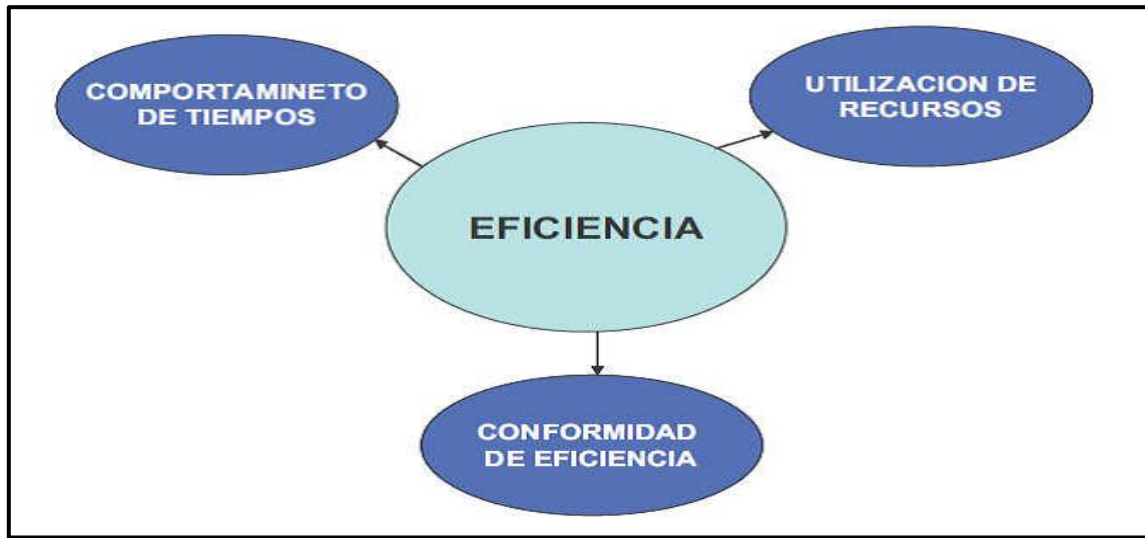


Figura 2.16. Característica de Eficiencia ISO/IEC 9126

Fuente: (Borbón, 2013)

La eficiencia se divide en 3 criterios:

Comportamiento de tiempos: Los tiempos adecuados de respuesta y procesamiento, el rendimiento cuando realiza su función en condiciones específicas. Ejemplo, ejecutar el procedimiento más complejo del software y esperar su tiempo de respuesta, realizar la misma función, pero con más cantidad de registros.

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.

2.15.2.5. Capacidad de mantenimiento

La capacidad de mantenimiento 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.



Figura 2.17. Característica de Mantenimiento ISO/IEC 9126

Fuente: (Borbón, 2013)

El mantenimiento se divide en 5 criterios:

Capacidad de ser analizado: La forma como el software permite diagnósticos de deficiencias o causas de fallas, o la identificación de partes modificadas.

Cambiabilidad: 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.

2.15.2.6. Portabilidad

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



Figura 2.18. Característica de Portabilidad ISO/IEC 9126

Fuente: (Borbón, 2013)

La portabilidad se divide en 5 criterios:

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. Ejemplo, la reemplazabilidad de una nueva versión 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.

2.15.2.7. Calidad de uso

Calidad en uso es la calidad del software que el usuario final refleja, la forma como el usuario final logra realizar los procesos con satisfacción, eficiencia y exactitud. La calidad en uso debe asegurar la prueba o revisión de todas las opciones que el usuario trabaja diariamente y los procesos que realiza esporádicamente relacionados con el mismo software.



Figura 2.19. Característica de Calidad de Uso ISO/IEC 9126

Fuente: (Borbón, 2013)

La calidad de uso se divide en 4 criterios:

Eficacia: La capacidad del software para permitir a los usuarios finales realizar los procesos con exactitud e integridad.

Productividad: La forma como el software permite a los usuarios emplear cantidades apropiadas de recursos, en relación a la eficacia lograda en un contexto específico de uso. Para una empresa es muy importante que el software no afecte a la productividad del empleado.

Seguridad: Se refiere al que el Software no tenga niveles de riesgo para causar daño a las personas, instituciones, software, propiedad intelectual o entorno. Los riesgos son normalmente el resultado de deficiencias en la funcionalidad (Incluyendo seguridad), fiabilidad, usabilidad o facilidad de mantenimiento.

Satisfacción: La satisfacción es la respuesta del usuario a la interacción con el software, e incluye las actitudes hacia el uso del mismo.

2.16. COSTOS

La estimación de los costos de desarrollo de software es un factor muy importante en el análisis de los proyectos informáticos, constituye un tema estratégico contar con indicadores para medir el costo de los mismos, garantizando la eficiencia, excelencia, calidad y la competitividad. El análisis de costo es el proceso de identificación de los recursos necesarios para llevar a cabo el trabajo o proyecto eficientemente.

La evaluación del costo determina la calidad y cantidad de los recursos necesarios en términos de dinero, esfuerzo, capacidad, conocimientos y tiempo incidiendo en la gestión empresarial. En la actualidad existen un conjunto de métricas que no se utilizan, y que pueden ser aplicables a cualquier tipo de proyecto de software para calcular el costo de los mismos.

2.16.1. Cocomo

El Modelo COCOMO Creado por Barry Boehm en 1981. Su nombre significa COConstructiveCOst MOdel (Modelo constructivo de costo) constituye una jerarquía de modelos de estimación para el software. La jerarquía está constituida por los siguientes modelos.

Cocomo básico. Calcula el esfuerzo y el costo del desarrollo en función del tamaño del programa estimado en LOC.

Cocomo intermedio. Calcula el esfuerzo del desarrollo en función del tamaño del programa y un conjunto de conductores de costo que incluyen la evaluación subjetiva del producto, del hardware, del personal y de los atributos del proyecto.

Cocomo detallado. Incorpora las características de la versión intermedia y lleva a cabo una evaluación del impacto de los conductores de costo en cada fase (análisis, desarrollo, etc.) del proceso.

COCOMO es una herramienta basada en las líneas de código la cual le hace muy poderoso para la estimación de costos y no como otros que solamente miden el esfuerzo en base al tamaño. Es necesario para un administrador de proyectos una herramienta de estimación de costos; y esta herramienta puede estar relacionada con COCOMO ya que esta técnica representa uno de los más completos modelos empíricos para la estimación de software.

2.16.2. Cocomo II

COCOMO II es un modelo que permite estimar el costo, el esfuerzo y el tiempo cuando se planifica una nueva actividad de desarrollo software, y está asociado a los ciclos de vida modernos. Fue desarrollado a partir de COCOMO, incluyendo actualizaciones y nuevas extensiones más adecuadas a los requerimientos de los ingenieros software.

Está construido para satisfacer aquellas necesidades expresadas por los estimadores de software, como por ejemplo el apoyo a la planificación de proyectos, la previsión de personal del proyecto, re planificación, seguimiento, negociaciones de contrato y la evaluación del diseño.

Los factores de costo describen aspectos relacionados con la naturaleza del producto, hardware utilizado, personal involucrado, y características propias del proyecto. El conjunto de factores de escala explica los ahorros y pérdidas producidos a medida que un proyecto de software incrementa su tamaño. El usar un modelo u otro depende del nivel de detalle del proyecto, de la fidelidad requerida de las estimaciones, de la definición de los requerimientos y de los detalles de la arquitectura . (Del Valle Roque, 2014)

2.16.2.1. Características

- a) Es una herramienta basada en las líneas de código la cual la hace muy poderosa para la estimación de costos y no como otros que solamente miden el esfuerzo en base al tamaño.

- b) Representa el más extenso modelo empírico para la estimación de software.
- c) Existen herramientas automáticas que estiman costos basados en COCOMO como ser: Costar, COCOMO 81.

2.16.2.2. Modelos de COCOMO II

Los tres modelos de COCOMO II se adaptan tanto a las necesidades de los diferentes sectores, como al tipo y cantidad de información disponible en cada etapa del ciclo de vida de desarrollo, lo que se conoce por granularidad de la información. Estos tres modelos son:

- **Modelo de composición de aplicación.** Utilizado durante las primeras etapas de la Ingeniería del software, donde el prototipado de las interfaces de usuario, la interacción del sistema y del software, la evaluación del rendimiento, y la evaluación de la madurez de la tecnología son de suma importancia.
- **Modelo de fase de diseño previo.** Utilizado una vez que se han estabilizado los requisitos y que se ha establecido la arquitectura básica del software.
- **Modelo de fase posterior a la arquitectura.** Utilizado durante la construcción del software. (ECURED, 2019)

2.16.2.3. Modelos de Estimación

En la estimación del tamaño de software COCOMO II utiliza tres técnicas:

- Puntos Objeto,
- Puntos Función No Ajustados y
- Líneas de Código Fuente.

Además, se emplean otros parámetros relativos al tamaño que contemplan aspectos tales como: reuso, reingeniería, conversión y mantenimiento.

a) Puntos Objeto

A pesar de que la estimación a través de Puntos Objeto es un enfoque de medición de tamaño de software relativamente nuevo, es apropiado para las aplicaciones con componentes y para estimar esfuerzos en las etapas de prototipación.

El procedimiento para determinar Puntos Objeto en un proyecto software se resume en:

1. **Determinar Cantidad de Objetos:** Estimar la cantidad de pantallas, reportes, componentes que contendrá la aplicación.
2. **Clasificar cada instancia de un objeto según sus niveles de complejidad** (simple, media o difícil).
3. **Dar el peso a cada objeto según el nivel de complejidad.** Los pesos reflejan el esfuerzo relativo requerido para implementar una instancia de ese nivel de complejidad.
4. **Determinar la cantidad de Puntos Objeto,** sumando todos los pesos de las instancias de los tipos de objetos especificados.

Para Pantallas			
Cantidad de Vistas Contenidas	Cantidad y fuente de las tablas de datos		
	Total < 4 (< 2 servidor < 3 cliente)	Total < 8 (< 2 - 3 servidor < 3 - 5 cliente)	Total 8 + (> 3 servidor < 5 cliente)
< 3	Simple	Simple	Media
3 - 7	Simple	Media	Difícil
> 8	Media	Difícil	Difícil
Para Reportes			
Cantidad de Vistas Contenidas	Cantidad y fuente de las tablas de datos		
	Total < 4 (< 2 servidor < 3 cliente)	Total < 8 (< 2- 3 servidor < 3-5 cliente)	Total 8 + (> 3 servidor < 5 cliente)
0 o 1	Simple	Simple	Media
2 o 3	Simple	Media	Difícil
4 +	Media	Difícil	Difícil

Figura 2.20. Clasificación de Puntos Objetos
FUENTE: (Gomez, López, Migani, & Otazú, 2010)

Tabla 2.1.
Peso de un Punto Objeto

Tipo de Objeto	Complejidad - Peso		
	Simple	Media	Difícil
Pantalla	1	2	3
Reporte	2	5	8
Componente 3GL			10

FUENTE: (Gomez, López, Migani, & Otazú, 2010)

La cuenta de los puntos de objetos se determina multiplicando el numero original de instancias del objeto por el factor del peso y se suman para obtener la cuenta total de los puntos objetos.

Se debe calcular el porcentaje de reutilización de software de software para así realizar el ajuste de los puntos objetos ya calculados. Entonces los puntos objetos nuevos se calculan de la siguiente manera:

$$NOP = (PUNTOS\ OBJETO) \times [(100 - \%reuso)/100]$$

Para realizar la estimación del esfuerzo todavía falta una variable llamada productividad, la cual se halla por medio de la siguiente tabla:

Tabla 2.2.
Productividad para el modelo Composición de Aplicación

Modelo Composición de Aplicación					
Experiencia/capacidad del desarrollador	Muy Baja	Baja	Normal	Alta	Muy Alta
Madurez/capacidad del entorno	Muy Baja	Baja	Normal	Alta	Muy Alta
Productividad (PROD)	4	7	13	25	50

FUENTE: (Gomez, López, Migani, & Otazú, 2010)

Con los valores necesarios ya encontrados se calcula el esfuerzo con la siguiente ecuación:

$$\text{Esfuerzo estimado} = \text{NOP}/\text{PROD}$$

b) Puntos Función

El modelo COCOMO II usa Puntos Función y/o Líneas de Código Fuente (SLOC) como base para medir tamaño en los modelos de estimación de Diseño Temprano y Post-Arquitectura.

Los puntos función están basados en información disponible en las etapas tempranas del ciclo de vida del desarrollo de software.

COCOMO II considera solamente UFP (Puntos Función no ajustados).

$$FP = UFP \times TCF$$

Donde **UFP**: Puntos Función no Ajustados

TCF: Factor de Complejidad Técnica

Para calcular los **UFP**, se deben identificar los siguientes elementos:

- **Entradas Externas (Inputs)**: Entrada de datos del usuario o de control que ingresan desde el exterior del sistema para agregar y/o cambiar datos a un archivo lógico interno.
- **Salidas Externas (Outputs)**: Salida de datos de usuario o de control que deja el límite del sistema de software.
- **Archivo Lógicos Internos (Archivos)**: Incluye cada archivo lógico, es decir cada grupo lógico de datos que es generado, usado, o mantenido por el sistema de software.
- **Archivos Externos de Interface (Interfaces)**: Archivos transferidos o compartidos entre sistemas de software.

- **Solicitudes Externas (Queries):** Combinación única de entrada-salida, donde una entrada causa y genera una salida inmediata, como un tipo de solicitud externa.

Una vez identificados los elementos se clasifican de acuerdo al grado de complejidad en: bajo, promedio o alto. Se asigna un peso a cada ítem según el tipo y el grado de complejidad correspondiente.

Finalmente, los **UFP** son calculados sumando los pesos de todos los ítems identificados.

$$UFP = \sum_{i=1}^{15} (Cantidad_Items_Tipo_i) \times (Peso_i)$$

Para archivos lógicos internos y archivos externos de interfase				Para salidas y consultas externas				Para entradas externas			
Elementos de Registro	Elementos de datos			Tipos de archivos	Elementos de datos			Tipos de archivos	Elementos de datos		
	1-19	20-50	51+		1-5	6-19	20+		1-4	5-15	16+
1	Bajo	Bajo	Prom.	0 ó 1	Bajo	Bajo	Prom.	0 ó 1	Bajo	Bajo	Prom.
2-5	Bajo	Prom.	Alto	2-3	Bajo	Prom.	Alto	2-3	Bajo	Prom.	Alto
6+	Prom.	Alto	Alto	4+	Prom.	Alto	Alto	3+	Prom.	Alto	Alto

Figura 2.21. Puntos Función Determinación del Peso
Fuente: (Gomez, López, Migani, & Otazú, 2010)

Tipo de función	Peso del Factor de Complejidad		
	Bajo	Promedio	Alto
Entradas Externas (Inputs)	3	4	6
Salidas Externas (Outputs)	4	5	7
Archivo Lógicos Internos (Archivos)	7	10	15
Archivos Externos de Interfase (Interfases)	5	7	10
Consultas Externas (Queries)	3	4	6

Figura 2.22. Peso del Factor de Complejidad
Fuente: (Gomez, López, Migani, & Otazú, 2010)

Para el cálculo del Factor de Complejidad Técnica, **TCF**, se considera la siguiente fórmula:

$$TCF = 0.65 + 0.01 \times \sum_{i=1}^{14} F_i$$

c) Líneas de Código Fuente (SLOC)

El objetivo es medir la cantidad de trabajo intelectual puesto en el desarrollo de un programa.

Definir una línea de código es difícil debido a que existen diferencias conceptuales cuando se cuentan sentencias ejecutables y de declaraciones de datos en lenguajes diferentes.

A los efectos de COCOMO II, se eliminan las categorías de software que consumen poco esfuerzo. Así no están incluidas librerías de soporte, sistemas operativos, librerías comerciales, etc., ni tampoco el código generado con generadores de código fuente.

d) Conversión de Puntos Función a Líneas de Código Fuente (SLOC)

Para determinar el esfuerzo nominal en el modelo COCOMO II los puntos función no ajustados tienen que ser convertidos a líneas de código fuente considerando el lenguaje de implementación. Esto se realiza para los modelos Diseño Temprano y Post Arquitectura. (Gomez, López, Migani, & Otazú, 2010)

2.17. ISO 27001 SEGURIDAD DE LA INFORMACIÓN

La ISO 27001 es la norma internacional para los sistemas de gestión de la seguridad de la información (SGSI). Proporciona un marco robusto para proteger la información que se puede adaptar a organizaciones de todo tipo y tamaño. Las organizaciones más expuestas a los riesgos relacionados con la seguridad de la información eligen cada vez más implementar un SGSI que cumpla con la norma ISO 27001.

El propósito central de un SGSI es proporcionar protección a la información sensible o de valor.

La información sensible incluye información sobre los empleados, clientes y proveedores. La información de valor incluye propiedad intelectual, datos financieros, registros legales datos comerciales y datos operativos.

Los tipos de riesgos que la información sensible y de valor sufren pueden agruparse en 3 categorías:

- **Confidencialidad** cuando una o más personas ganan acceso no autorizado a la información.
- **Integridad** cuando el contenido de la información se cambia de manera que ya no es precisa o completa
- **Disponibilidad** cuando se pierde o daña el acceso a la información.

Estos tipos de riesgo de seguridad de la información se conocen comúnmente como "CID".

Los riesgos en la seguridad de la información generalmente surgen debido a la presencia de amenazas para los activos que procesan, almacenan, mantienen, protegen o controlan el acceso a la información, lo que da lugar a incidentes.

Los activos en este contexto suelen ser personas, equipos, sistemas o infraestructura.

La información es el conjunto de datos que una organización desea proteger, como registros de empleados, de clientes, datos financieros, de diseño, de prueba, etc.

Los incidentes son eventos no deseados que resultan en una pérdida de confidencialidad (violación de datos), integridad (corrupción de datos) o disponibilidad (fallo del sistema).

Las amenazas son las que causan incidentes y pueden ser maliciosas (por ejemplo, un robo), accidentales (por ejemplo, un error tipográfico) o un acto de divino (por ejemplo, una inundación).

Las vulnerabilidades, como las ventanas abiertas de la oficina, los errores del código fuente o la ubicación junto a los ríos, aumentan la probabilidad de que la amenaza provoque un incidente no deseado y costoso.

En seguridad de la información, el riesgo se gestiona mediante el diseño, implementación y mantenimiento de controles como ventanas bloqueadas, pruebas de software o la ubicación de equipos vulnerables por encima de la planta baja.

Un SGSI que cumple con la ISO 27001 tiene un conjunto interrelacionado de procesos de mejores prácticas que facilitan y respaldan el diseño, implementación y mantenimiento de los controles. Los procesos que forman parte del SGSI suelen ser una combinación de procesos comerciales centrales existentes (por ejemplo, reclutamiento, inducción, capacitación, compras, diseño de productos, mantenimiento de equipos, prestación de servicios) y aquellos específicos para mantener y mejorar la seguridad de la información (por ejemplo, gestión de cambios, respaldo de información, control de acceso, gestión de incidentes, clasificación de la información). (ISO 27001: Sistemas de gestión de seguridad de la información, s.f.)

A thick, dark blue vertical bar is positioned on the left side of the page. From the bottom of this bar, several thin, curved lines in shades of blue and grey extend outwards and upwards, creating an abstract, organic shape.

CAPITULO III

MARCO APLICATIVO

3.1. INTRODUCCIÓN

En el presente capítulo se desarrollará las fases correspondientes a las conceptualizaciones, análisis y diseño del sistema siguiendo el proceso de desarrollo de la metodología UWE detallado en el capítulo II.

3.2. DESCRIPCIÓN GENERAL DE COMITÉ DE AGUA POTABLE LAKA PUCARA

El comité de agua potable Laka Pucara es una organización comunitaria sin fines de lucro cuya finalidad es prestar el servicio público de agua potable a la comunidad en sus 3 zonas. Por tal motivo se encuentra conformada por un presidente, secretario, tesorero y operador responsables del cumplimiento de las siguientes funciones.

- Gestión técnica (operación y mantenimiento de los servicios).
- Gestión administrativa (manejo administrativo y contable del comité y cobranzas por los servicios).

Como anteriormente se describió el comité de agua potable es una organización sin fines de lucro, esto significa que los fondos recaudados son reinvertidos en el mantenimiento de la infraestructura de bombeo del agua, lo cual permite brindar un servicio de calidad a los clientes.

Un Comité de Agua Potable y Saneamiento debe ser sostenible, por tanto, debe llevar un cobro adecuado, realizar de manera adecuada la operación y mantenimiento del sistema y, sobretodo, rendir cuentas a la población.

3.3. ESTRUCTURA DE COMITÉ DE AGUA POTABLE LAKA PUCARA

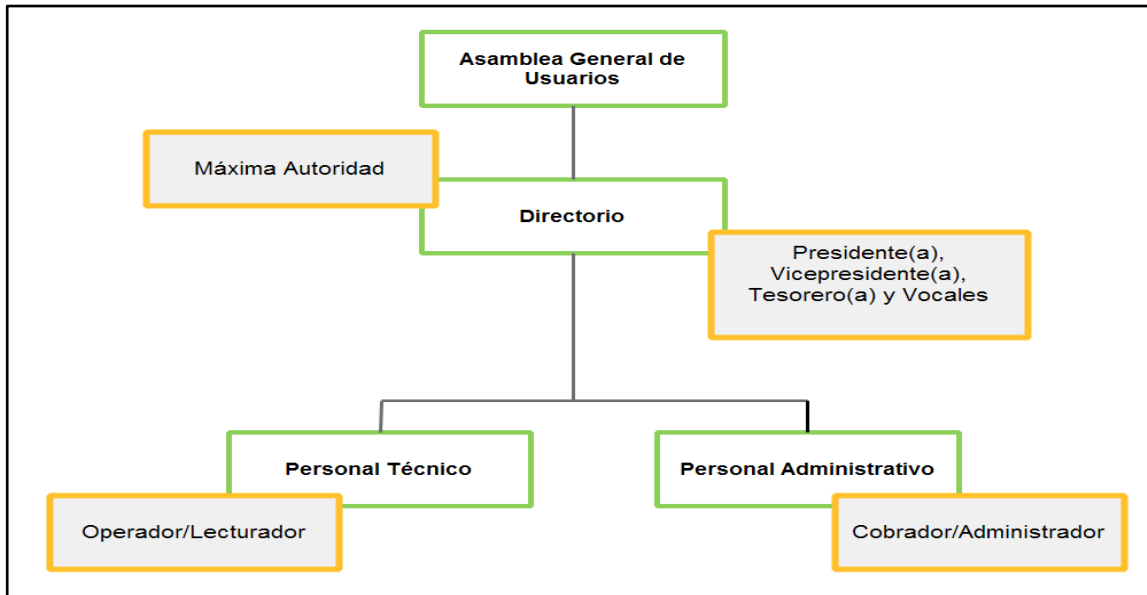


Figura 3.23. Estructura Organizacional de CAPyS Laka Pucara
Fuente: (Elaboración Propia)

3.4. ESQUEMA DEL SISTEMA

El Sistema de Información para el Control de Pagos por el Suministro de Agua Potable, es una aplicación web que cuenta con un administrador, una tesorera y un operador, donde las personas denominadas serán las que interactuarán con el sistema de la siguiente manera:

El administrador será la persona encargada de tener el control total del sistema, como permisos de acceso a los módulos, registro de usuarios, registro de clientes, registro de tarifas, registro de multas, consulta de planilla de consumo y reportes e informes.

La tesorera es la persona encargada de realizar el cobro correspondiente previo la verificación de planilla de consumo con solo ingresando el número de medidor, se desplegará la panilla y el costo a cancelar, una vez realizado el cobro por el consumo de agua, procederá a imprimir la boleta de comprobante y luego entregará al cliente.

También esta persona podrá consultar las deudas de los clientes que hayan sido multadas por alguna razón de incumplimiento de las normas la misma podrá cobrar las multas.

El operador es la persona encargada de registro de lecturas de los medidores desde los domicilios de los clientes, donde ingresará la información de consumo desde un dispositivo móvil que tendrá una conexión a internet.

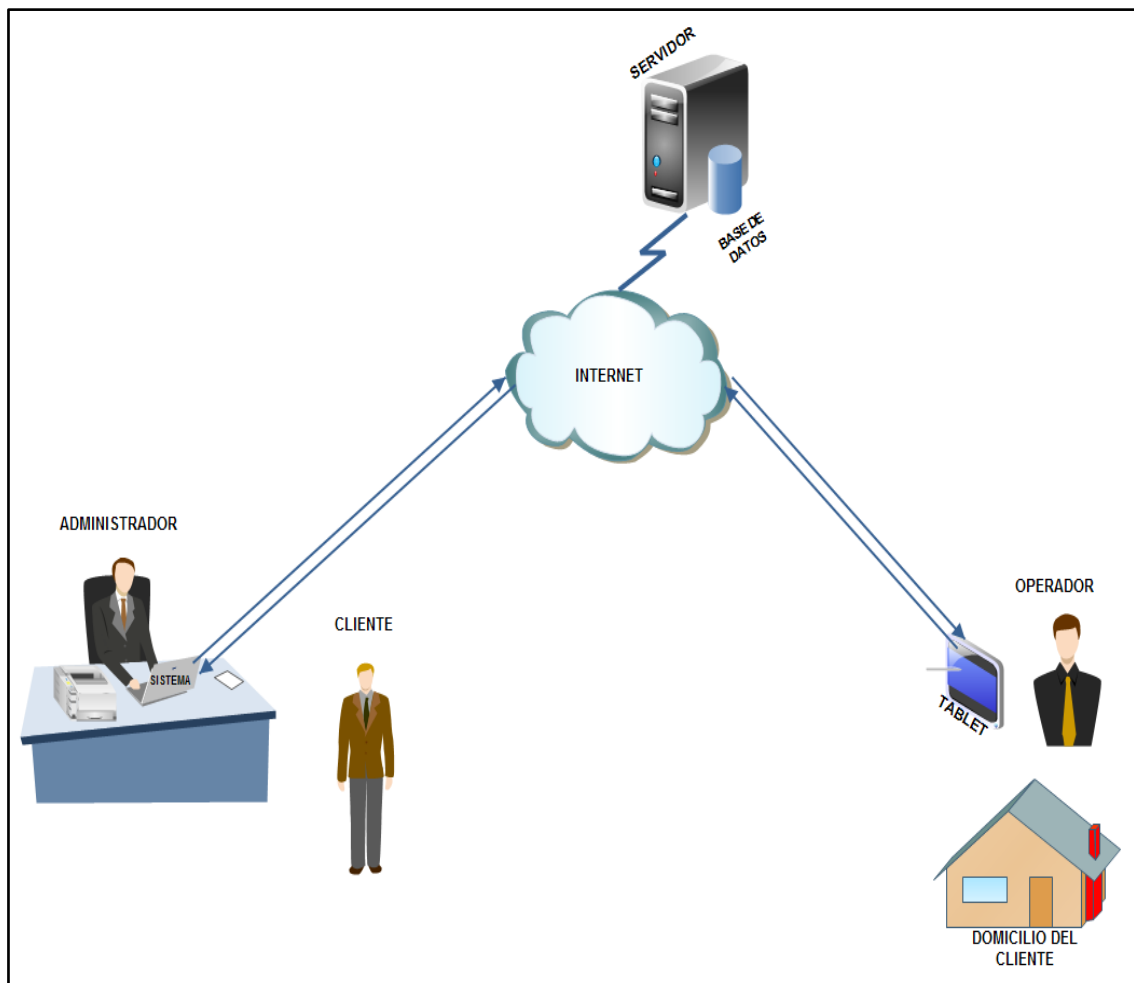


Figura 3.24. Esquema del Sistema
Fuente: (Elaboración Propia)

3.5. OBTENCIÓN DE REQUERIMIENTOS

Como sabemos, un área de conocimiento de gran importancia en el desarrollo de software, es la ingeniería de requerimientos. Esta comprende las actividades de obtención (captura, descubrimiento y adquisición), especificación, y validación de requisitos. Además, establece una actividad de gestión de requerimientos para manejar los cambios, mantenimiento y rastreabilidad de los requerimientos.

El proceso de obtención de requisitos, cuya finalidad es llevar a la luz los requisitos, no solo es un proceso técnico, sino también un proceso social que envuelve a diferentes personas, lo que conlleva dificultades añadidas a su realización.

Existe un gran número de técnicas para obtener requerimientos. A continuación, describo las más utilizadas. Hay que aclarar que ninguna de estas técnicas es suficiente por sí sola y que es recomendable combinarlas para obtener requerimientos completos.

Tabla 3.3.
Tareas de Obtención de Requisitos

TÉCNICAS REALIZADAS	DESCRIPCIÓN
Entrevistas	Se ha realizado las entrevistas con las siguientes personas de comité de agua potable: <ul style="list-style-type: none">➤ Presidente➤ Tesorera➤ Operador
Observación	En la comunidad Laka Pucara se pudo evidenciar de manera directa los procesos en cuanto a la lectura de medidores, el cobro de planillas y los documentos de la información en el comité de agua potable.

	<p>Se pudo observar lo siguiente:</p> <ul style="list-style-type: none"> ➤ La lectura de medidores se lo realiza de manera manual y se registra en hojas. ➤ La información que se recaba de la lectura de medidores se entrega a la tesorera. ➤ Los documentos se conservan de forma física. ➤ El tiempo de respuesta cuando un usuario requiere información no es oportuna. ➤ Los cálculos para determinar los cobros no son precisos. ➤ No tienen un sistema informático automatizado. ➤ Los clientes para conocer las planillas de consumo y el monto a cancelar necesariamente tienen que ir hasta el comité de agua potable Laka Pucara.
Encuesta	<p>A través de la misma se recopiló información, lo que nos permitió evaluar las necesidades existentes de los clientes en relación a la recaudación de tarifas por el consumo de agua y los servicios brindados por el comité de agua potable Laka Pucara.</p>

Fuente: (Elaboración Propia)

3.6. INGENIERÍA DE REQUERIMIENTOS

La Ingeniería de Requerimientos contempla todas las tareas específicas para satisfacer las necesidades durante el proceso de creación o modificación de un software.

Es importante la Ingeniería de Requerimientos en el proceso de desarrollo de un software porque prácticamente esto da comienzo para actividades de planeación donde podremos hacer estimaciones sobre el tiempo, costos y recursos que necesitaremos al momento de desarrollar el software, además nos permite hacer un cronograma que es el principal mecanismo de control con el que podemos contar durante la etapa de desarrollo.

Las especificaciones de requerimientos, nos permiten verificar si se están cumpliendo o no los objetivos establecidos ya que estos son el reflejo de los requerimientos del cliente, usuarios que nos permite verificar el cumplimiento de metas.

3.6.1. Requerimientos funcionales

Los requerimientos funcionales contemplan todo lo que el usuario desea que realice el sistema, se detallan a continuación en la tabla 3.4.

Tabla 3.4.
Requerimientos Funcionales

REF.	FUNCIÓN	CATEGORÍA
R1	Control de acceso al Sistema	Evidente
R2	Gestión de privilegios de acceso	Evidente
R3	Desplegar vista y menús de acuerdo de cada usuario	Oculto
R4	Gestión de Usuarios	Evidente
R5	Gestión de Clientes	Evidente
R6	Gestión de Transferencia de Clientes	Evidente
R7	Gestión de Multas	Evidente
R8	Gestión de Tarifas	Evidente
R9	Historial de Pagos	Evidente
R10	Registro de lectura de medidor	Evidente
R11	Determinar el estado de cada usuario	Evidente
R12	Búsqueda al instante en los datos requeridos	Evidente
R13	Consultas de planillas	Evidente
R14	Emisión de Reportes	Evidente

Fuente: (Elaboración Propia)

3.6.2. Requerimientos no funcionales

Los requerimientos no funcionales contemplan todo lo que se necesita para que el sistema funcione correctamente.

Tabla 3.5.

Requerimientos No Funcionales

REF.	FUNCIÓN	CATEGORÍA
R1	Validación de campos obligatorios	Evidente
R2	Validación de cédula de identidad	Evidente
R3	El sistema debe tener seguridad en el acceso a la información.	Evidente
R4	La interfaz presentada debe ser fácil de usar con interfaces intuitivas.	Evidente
R5	El sistema debe visualizarse y compatible en cualquier navegador como: Microsoft Edge, Opera, Mozilla Firefox, Chrome.	Evidente

Fuente: (Elaboración Propia)

3.7. MODELADO DEL SISTEMA POR MEDIO DE LA METODOLOGÍA UWE

3.7.1. Modelo Caso de Uso

Un caso de uso es una descripción de las acciones de un sistema desde el punto de vista del usuario. Los diagramas de caso de uso modelan la funcionalidad del sistema usando actores y casos de uso. Los casos de uso son servicios o funciones provistas por el sistema para sus usuarios.

3.7.1.1. Identificación de actores

Los actores representan a los usuarios que presenta el sistema. Se comprende como usuario cualquier persona que llegue a interactuar con el sistema.

A continuación, se describe a los actores que interactúan con el sistema que se desarrolla.

Tabla 3.6.
Descripción de Actores del Sistema

ACTOR	DESCRIPCIÓN
Administrador	Es la entidad encargada de interactuar directamente con el sistema para la creación de nuevos usuarios, administración de clientes, administración de tarifas, administración de multas, consulta de planillas, y reportes por el consumo de agua.
Tesorera	Entidad que representa a la persona encargada de realizar cobros correspondientes de planillas de consumo de agua mensual y emisión de comprobantes de pago.
Operador	Entidad que representa a la persona encargada de realizar el ingreso de la lectura del medidor.
Cliente	Entidad que representa a la persona beneficiaria del servicio de agua potable.

Fuente: (Elaboración propia)

3.7.1.2. Caso de uso general

El siguiente diagrama de caso de uso general, describe las actividades que realiza cada uno de los actores, este diagrama nos sirve para especificar la comunicación y el comportamiento del sistema mediante su interacción con los usuarios de forma general. Continuación presentamos diagrama de caso de uso general en la figura 3.25.

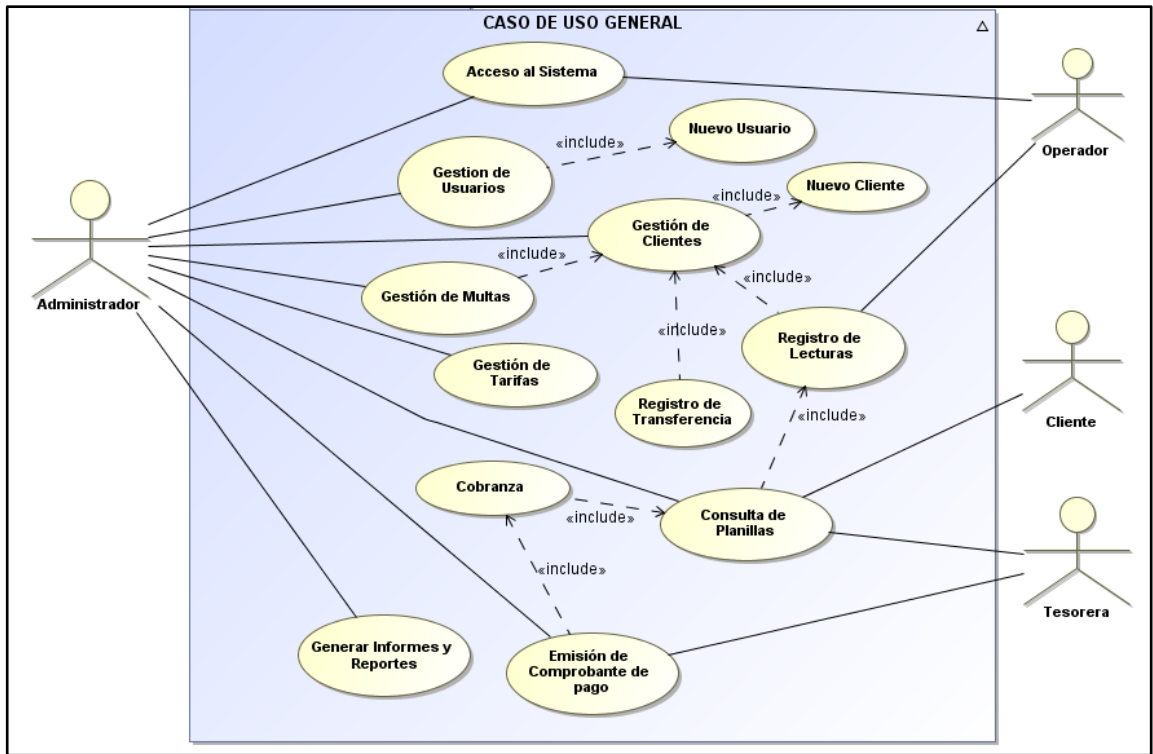


Figura 3.25. Diagrama de Caso de Uso General
Fuente: (Elaboración Propia)

3.7.1.3. Caso de Uso: Acceso al Sistema

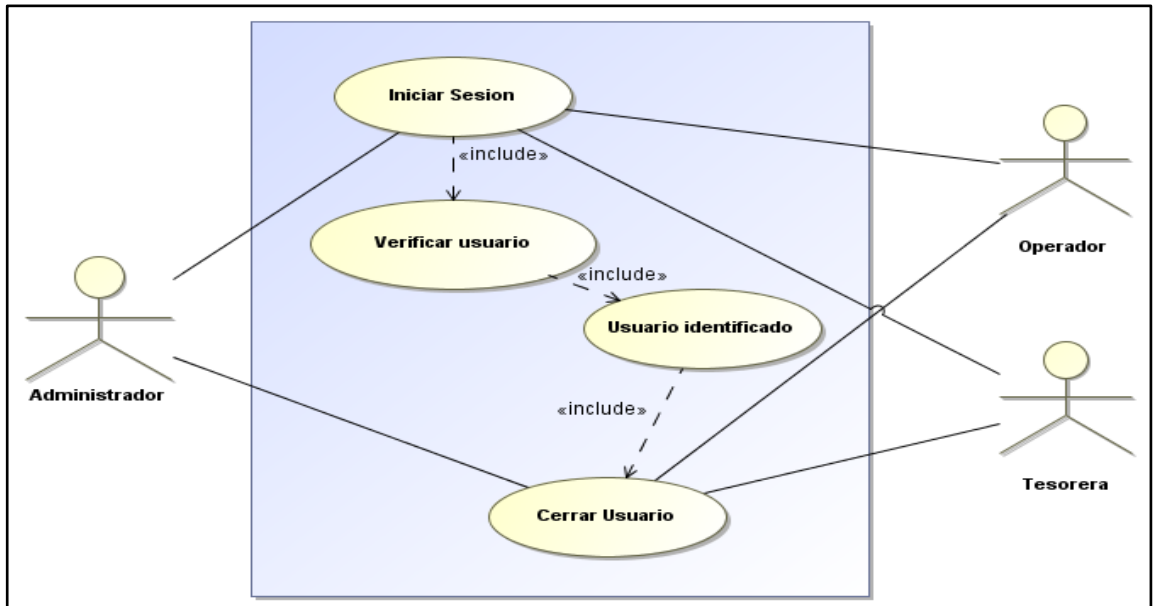


Figura 3.26. Diagrama de Caso de Uso Acceso al Sistema
Fuente: (Elaboración Propia)

Tabla 3.7.
Descripción de Acceso al Sistema

Caso de Uso: Acceso al Sistema		
Actor:	Administrador/Tesorerera/Operador/Sistema	
Descripción:	Permite la validación del administrador que accede al Sistema.	
Flujo Principal:	Eventos ACTOR	Eventos SISTEMA
	Ingresar al Sistema	Presenta el formulario de Login.
	Proporciona las credenciales de acceso	El sistema valida la información ingresada.
	El Administrador/Tesorerera/Operador tiene acceso al sistema.	Presenta los módulos disponibles de acuerdo a las credenciales que tiene el usuario logueado.
Alternativa:	El usuario no puede ingresar al sistema porque no se encuentra registrado o error en la contraseña.	
Precondición:	El usuario debe estar registrado en la base de datos.	
Pos condición:	El usuario accede al sistema.	
Presunción:	El sistema esté conectado con la base de datos.	

Fuente: (Elaboración Propia)

3.7.1.4. Caso de Uso: Gestión de Usuarios

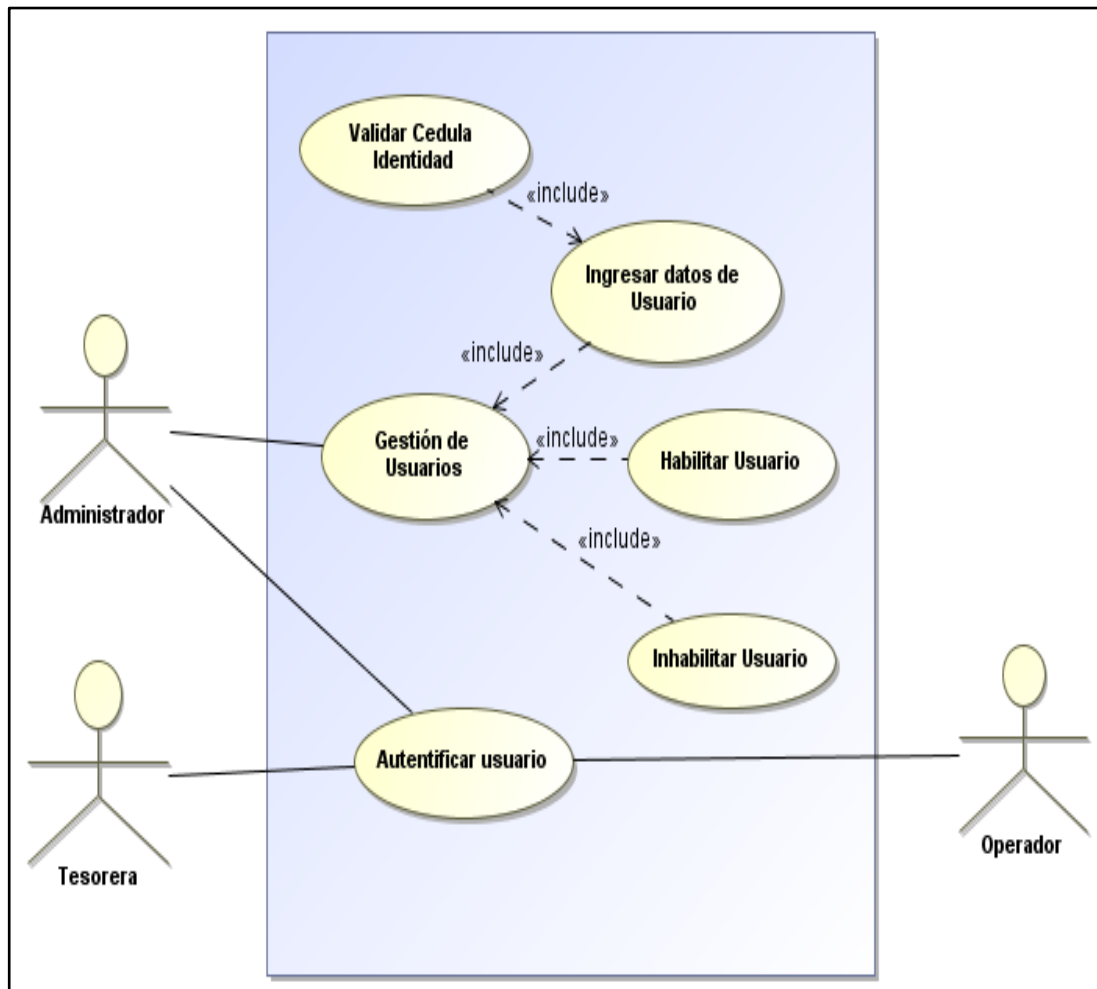


Figura 3.27. Diagrama de Caso de Uso Gestión de Usuarios
Fuente: (Elaboración Propia)

Tabla 3.8.
Descripción de Gestión de Usuarios

Caso de Uso: Gestión de Usuarios	
Actor:	Administrador
Descripción:	Describe el proceso de control de las cuentas de usuario, es decir, el registro de nuevo usuario, la modificación, y la desactivación de una cuenta de usuario.

	Eventos ACTOR	Eventos SISTEMA
Flujo Principal:	Recopilación de la información correspondiente .	
	Ingreso a módulo de registro de usuario.	Presenta el formulario para el ingreso de los datos del usuario.
	Llena los datos del usuario, asignando también los privilegios que tendrá en el sistema.	El sistema muestra el resumen de los datos ingresado.
	Asigna contraseña y un nombre al nuevo registro.	El sistema valida los datos asignados como la contraseña y su nombre.
	Envía los datos de usuario.	El sistema confirma el registro en la base de datos la información con una alerta.
Alternativa:	Si el sistema encuentra datos en la base de datos, el sistema despliega con una alerta indicando que el dato ingresado ya existe registrado.	
Precondición:	El administrador tiene los privilegios para realizar el ingreso de la información.	
Pos condición:	El usuario es dado de alta y adquiere una cuenta en el sistema.	
Presunción:	La base de datos del sistema se encuentra disponible para su guardar los datos.	

Fuente: (Elaboración Propia)

3.7.1.5. Caso de Uso: Gestión de Clientes

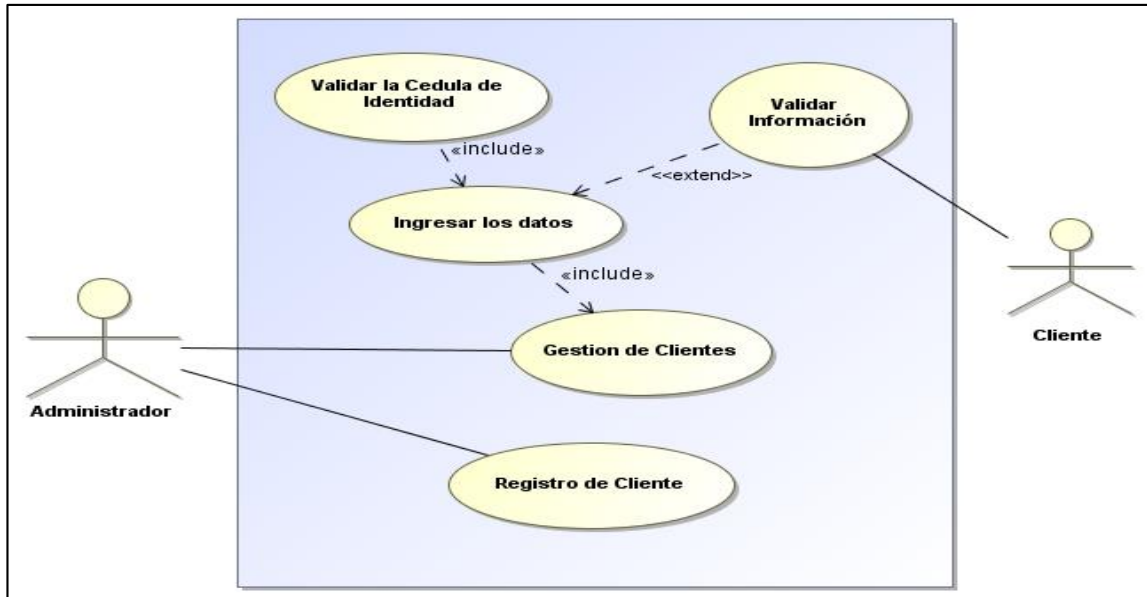


Figura 3.28. Diagrama de Caso de Uso Gestión de Clientes

Fuente: (Elaboración Propia)

Tabla 3.9.

Descripción de Gestión de Clientes

Caso de Uso: Gestión de Clientes		
Actor:	Administrador/Cliente	
Descripción:	Describe el proceso de alta de un nuevo cliente en el comité de agua potable.	
Flujo Principal:	Eventos ACTOR	Eventos SISTEMA
	Recopilación de la información correspondiente del cliente.	
	Ingreso a módulo de registro de cliente.	Presenta el formulario para el ingreso de los datos del cliente.
	Ingreso de datos del cliente.	El sistema muestra el resumen de los datos ingresado.

	El almacenamiento de la información.	El sistema confirma el registro de la información con una alerta.
Alternativa:	Se pide al cliente completar la información para registrarla en el sistema.	
Precondición:	El administrador tiene los privilegios para realizar el ingreso de la información. El cliente obtiene información sobre los documentos necesarios para el registro en el sistema.	
Pos condición:	El cliente es dado de alta y adquiere una cuenta en el sistema.	
Presunción:	La base de datos del sistema se encuentra disponible para su guardar los datos.	

Fuente: (Elaboración Propia)

3.7.1.6. Caso de Uso: Registro de Lectura de Medidor de Agua

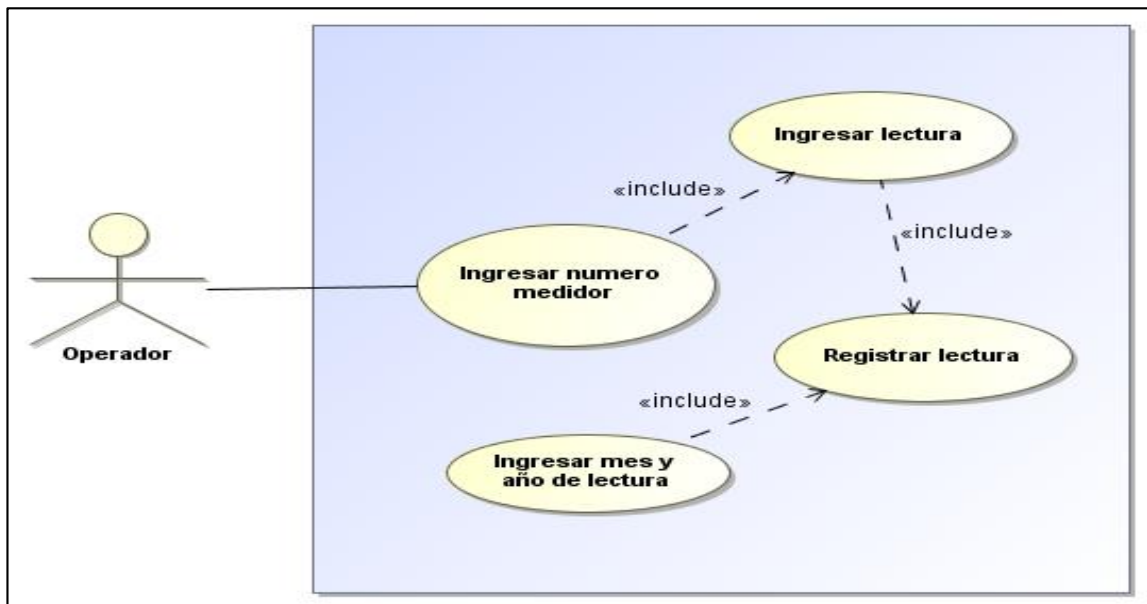


Figura 3.29. Diagrama de Caso de Uso Registro de Medidor

Fuente: (Elaboración Propia)

Tabla 3.10.
Descripción de Lectura de Medidor

Caso de Uso: Acceso al Sistema		
Actor:	Técnico u Operador	
Descripción:	Realiza el proceso de ingresar las lecturas de los medidores de cada cliente.	
Flujo Principal:	Eventos ACTOR	Eventos SISTEMA
	Ingresar al sistema	Muestra el menú de opciones disponible en el sistema.
	Consulta el medidor	Presenta la información del cliente que la pertenece el número del medidor ingresado
	Ingresar el consumo del medidor	El sistema verifica los datos introducidos.
	Almacena la información.	El sistema procesa y confirma el registro de la información con una alerta.
Alternativa:	El número de medidor ingresada no puede estar activo.	
Precondición:	El técnico deberá ingresar desde un dispositivo móvil y que tenga conexión a internet.	
Pos condición:	El sistema envía los datos al servidor.	
Presunción:	La base de datos del sistema se encuentra disponible para su guardar los datos.	

Fuente: (Elaboración Propia)

3.7.1.7. Caso de Uso: Consulta y Pago de Planillas

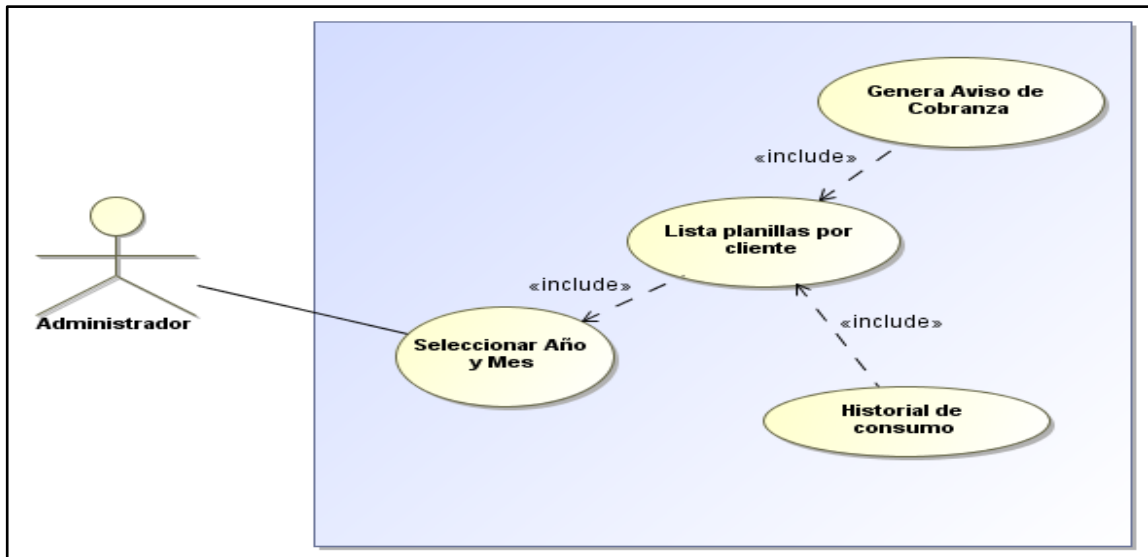


Figura 3.30. Diagrama de Caso de Consulta de Planilla
Fuente: (Elaboración Propia)

Tabla 3.11.
Descripción de Consulta y Pago de Planillas

Caso de Uso: Consulta de Planillas		
Actor:	Administrador	
Descripción:	Permite realizar el listado por cliente de acuerdo al mes y año seleccionado para luego generar aviso de cobranza por concepto de consumo de agua.	
Flujo Principal:	Eventos ACTOR	Eventos SISTEMA
	Ingresar al sistema	Muestra el menú de opciones disponible en el sistema.
	Seleccionar el mes y el año.	Genera una lista de clientes con respectiva información de pagado o pendiente, ordenada por fecha, mes y año

	Ingresar a generar aviso de cobranza.	Presenta la planilla con todo los datos del cliente y el total de consumo en Bs.
	Una vez visualizada el aviso de cobranza, solicita impresión.	Imprime aviso de cobranza
Alternativa:	El mes y año seleccionado no existe la información.	
Precondición:	El número de medidor debe estar registrado en el sistema.	
Pos condición:	El administrador obtiene la planilla, aviso de cobranza correspondiente.	
Presunción:	La base de datos del sistema se encuentra disponible .	

Fuente: (Elaboración Propia)

3.7.1.8. Caso de Uso: Emisión de Comprobante de pago

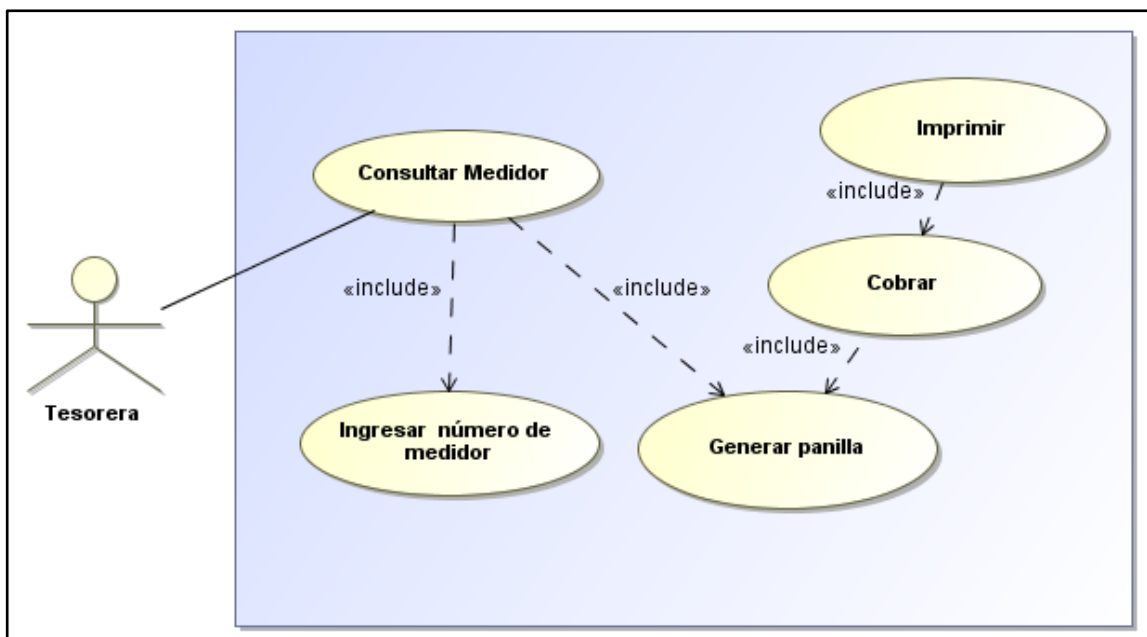


Figura 3.31. Diagrama de Caso de Comprobante de pago

Fuente: (Elaboración Propia)

Tabla 3.12.*Descripción de Comprobante de pago*

Caso de Uso: Emisión de comprobante de pago		
Actor:	Tesorera	
Descripción:	Permite generar un comprobante por concepto de consumo de agua.	
Flujo Principal:	Eventos ACTOR	Eventos SISTEMA
	Ingresar al sistema	Muestra el menú de opciones.
	Ingresar el número de medidor	Presenta el listado de consumo de agua del medidor seleccionado.
	Realiza el pago de la planilla.	Imprime el comprobante
Alternativa:	El número de medidor ingresada no puede estar activo o no correcto.	
Precondición:	El número de medidor debe estar registrado en el sistema.	
Pos condición:	Comprobante de pago generado exitosamente	
Presunción:	La base de datos del sistema debe estar disponible .	

Fuente: (Elaboración Propia)

3.7.1.9. Caso de Uso: Gestión de Tarifas

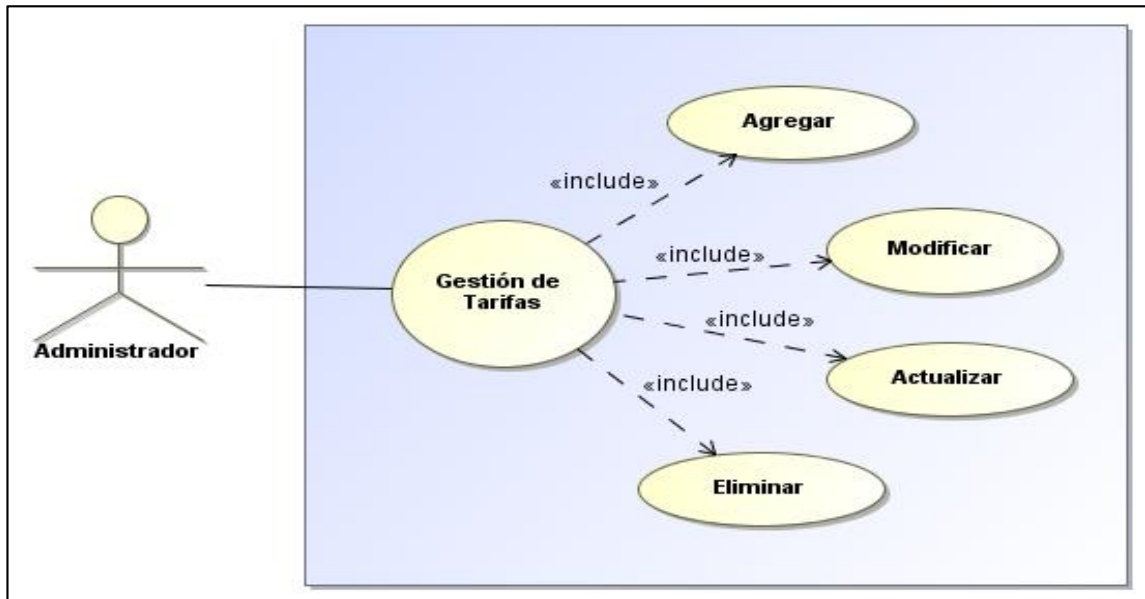


Figura 3.32. Diagrama de Caso de Gestión de Tarifas
Fuente: (Elaboración Propia)

Tabla 3.13.
Descripción de Gestión de Tarifas

Caso de Uso: Gestión de Tarifas		
Actor:	Administrador	
Descripción:	Permite al administrador el mantenimiento de tarifas establecidas según la categoría en el sistema.	
Flujo Principal:	Eventos ACTOR	Eventos SISTEMA
	Ingresa al sistema	Muestra el menú de opciones.
	Selecciona la opción de tarifas.	Presenta las opciones para realizar el CRUD de la información.

	Guarda los cambios	Almacena en la base de datos los cambios efectuados.
Alternativa:	El administrador no puede ingresar al sistema.	
Precondición:	El administrador debe iniciar sesión en el sistema.	
Pos condición:	Realiza el mantenimiento respectivo.	
Presunción:	El sistema y la base de datos están en condiciones de procesar la información.	

Fuente: (Elaboración Propia)

3.7.1.10. Caso de Uso: Informes e Reportes

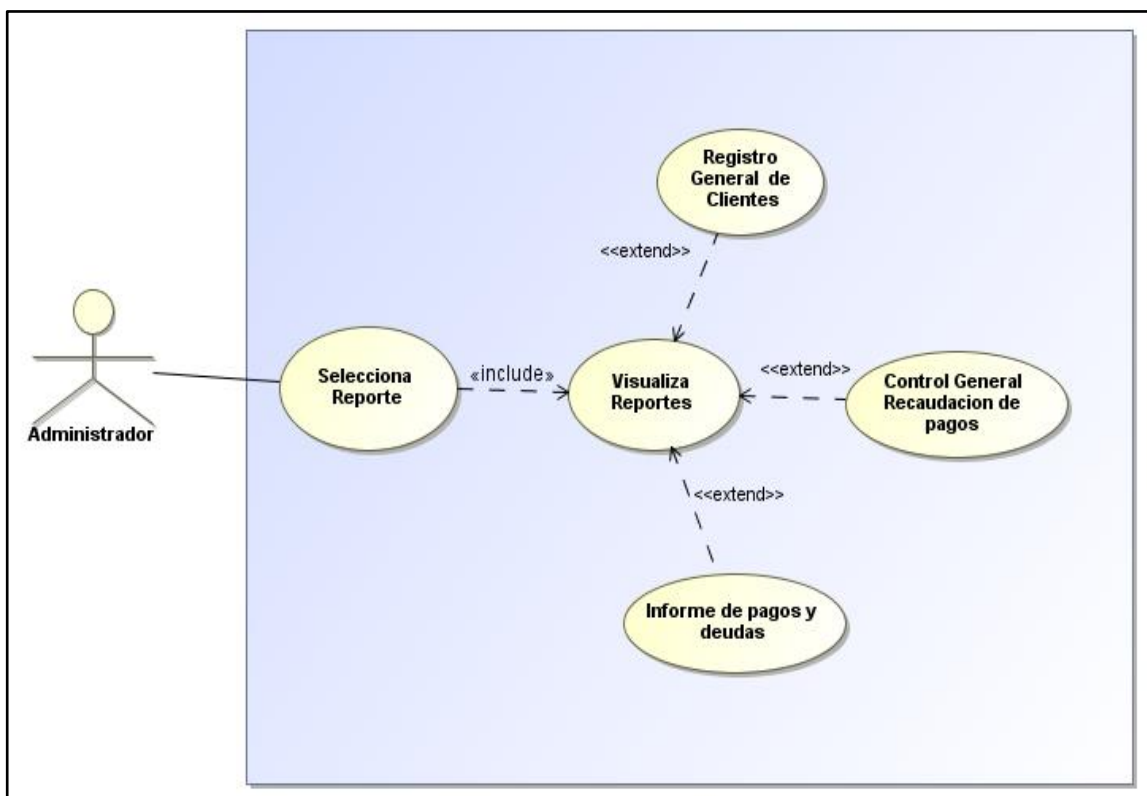


Figura 3.33. Diagrama de Caso de Informe e Reportes

Fuente: (Elaboración Propia)

Tabla 3.14.
Descripción de Reporte e Informe

Caso de Uso: Generar Informes y Reportes		
Actor:	Administrador	
Descripción:	El sistema realiza informes e reportes de cada actividad realizada.	
Flujo Principal:	Eventos ACTOR	Eventos SISTEMA
	Ingresar al sistema	Muestra el menú de opciones.
	El administrador elige la opción de generar informes y reportes en formato pdf y xls.	El sistema muestra la interfaz de generar informes y reportes.
	Una vez visualizada el reporte u informe , envía para ser imprimido.	Imprime el informe u reporte.
Alternativa:	El administrador no pueda ingresar al sistema.	
Precondición:	Los informes deben estar en disposición en cualquier momento que los necesite, ordenados por fecha.	
Pos condición:	Informe y reportes entregados al comité de agua potable.	
Presunción:	La base de datos del sistema debe estar disponible .	

Fuente: (Elaboración Propia)

3.8. MODELO CONCEPTUAL

Construir un modelo conceptual del dominio de la aplicación considerando los requisitos reflejados en los casos de uso.

Especifica cómo se encuentra relacionados los contenidos del sistema, como se muestra en la siguiente figura 3.34.

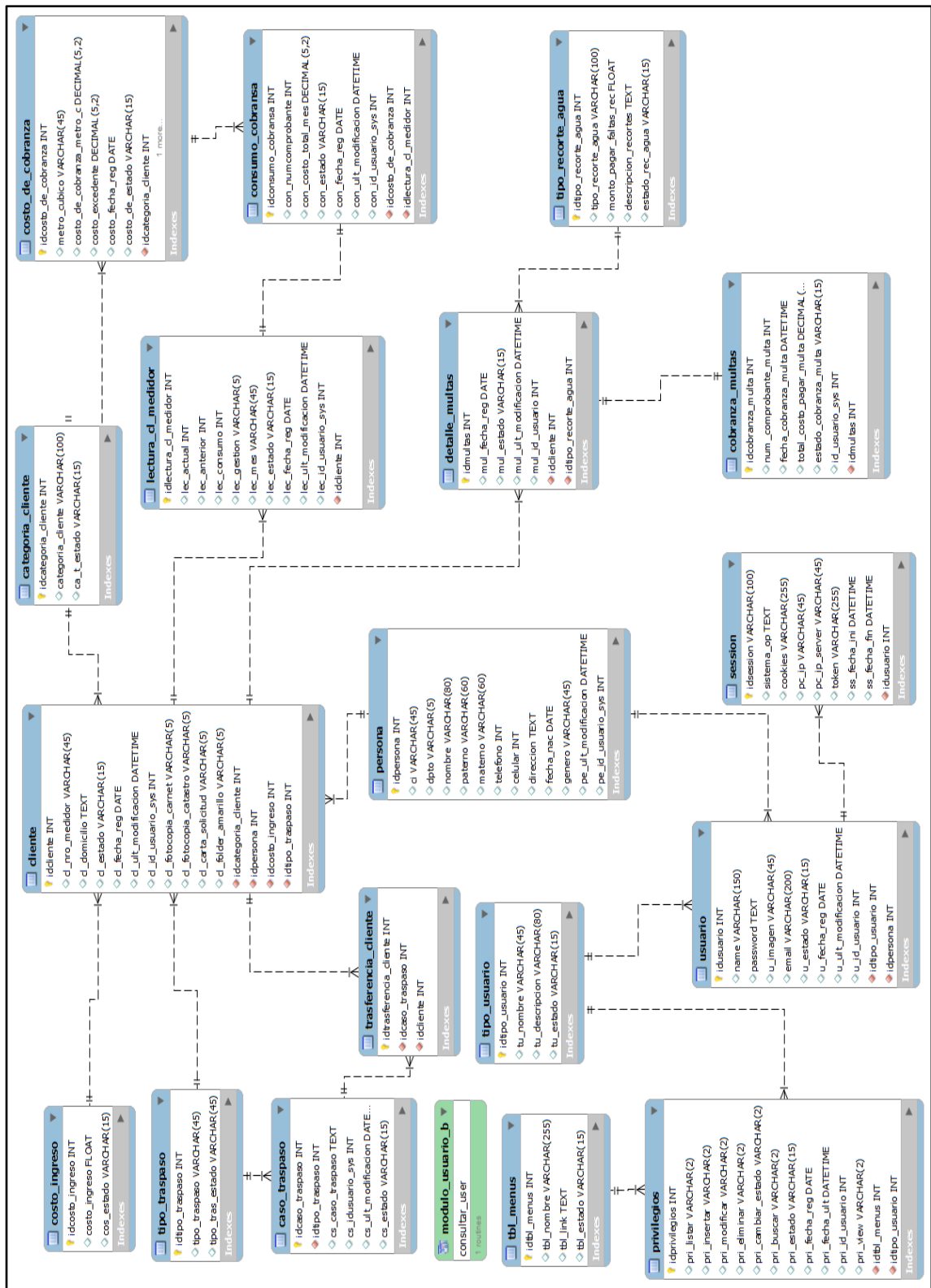


Figura 3.34. Modelo de Contenido

Fuente: (Elaboración propia)

3.9. MODELO DE NAVEGACIÓN

Este modelo de navegación permite ilustrar los vínculos lógicos y navegación entre clases, menús, índices y clases de proceso para tener una mejor comprensión del sistema se muestra las siguientes figuras:

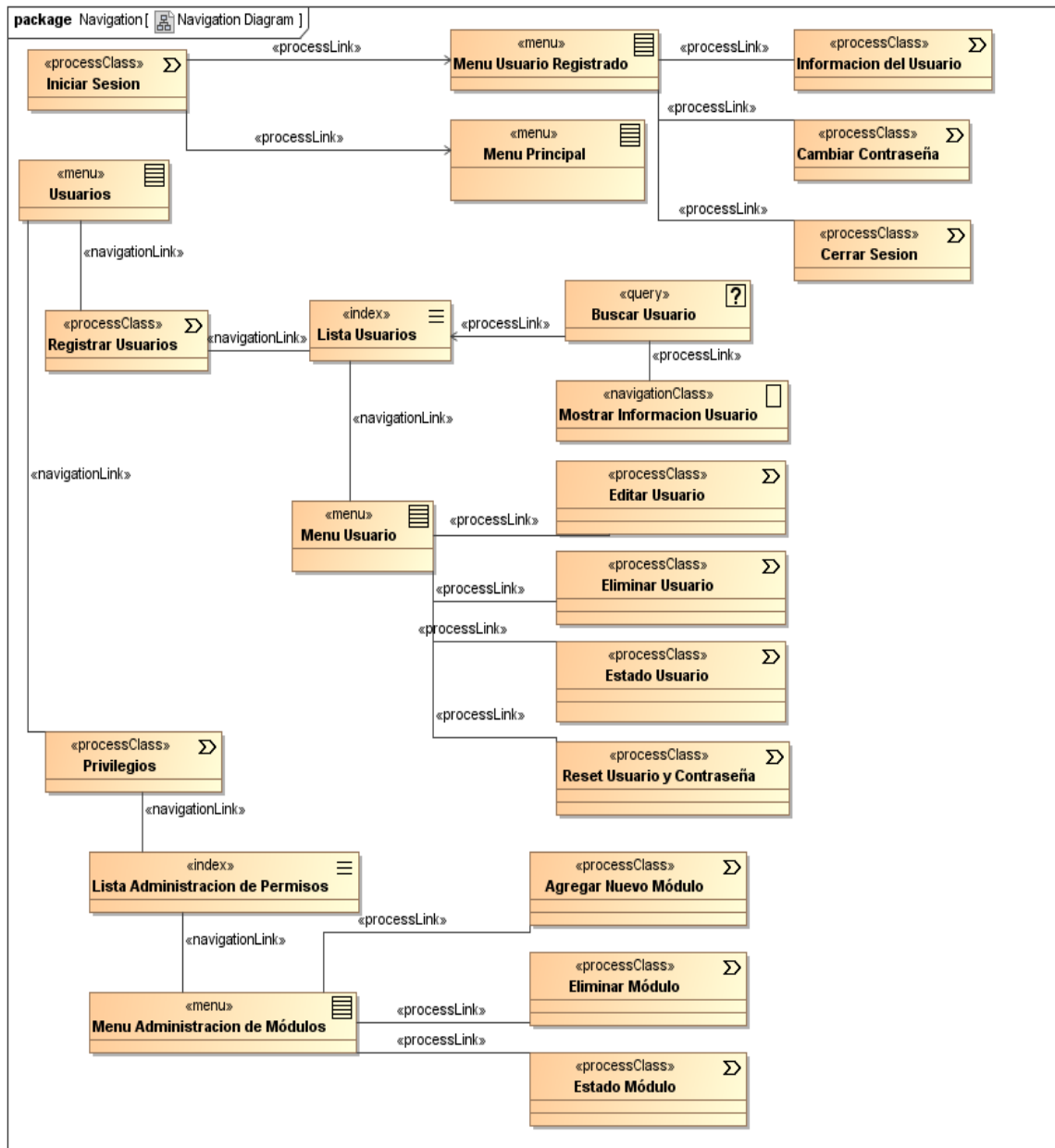


Figura 3.35. Modelo Navegación de Administración de Usuarios
Fuente: (Elaboración propia)

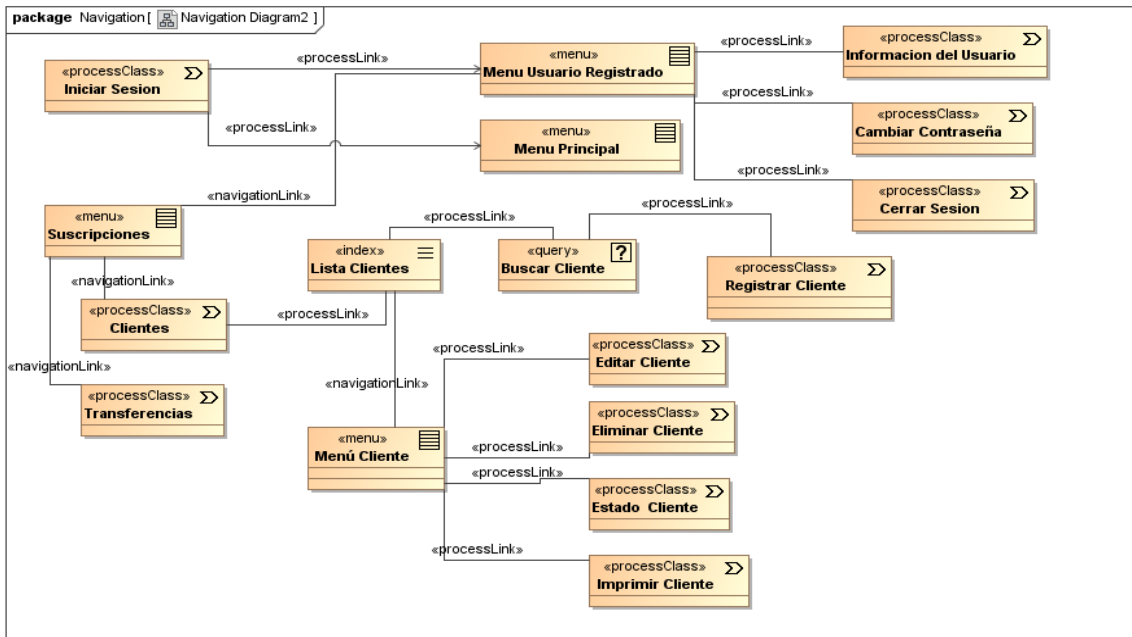


Figura 3.36. Modelo Navegación de Administración de Clientes
Fuente: (Elaboración propia)

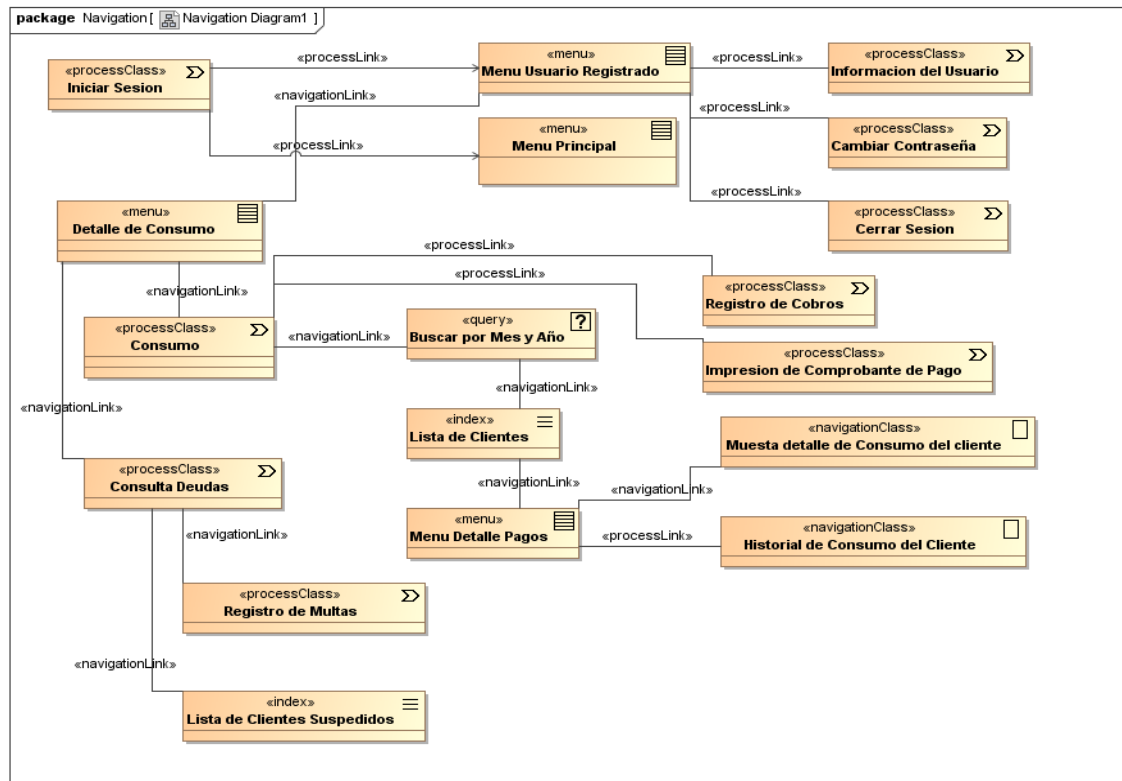


Figura 3.37. Modelo Navegación de Administración de Detalle Consumo
Fuente: (Elaboración propia)

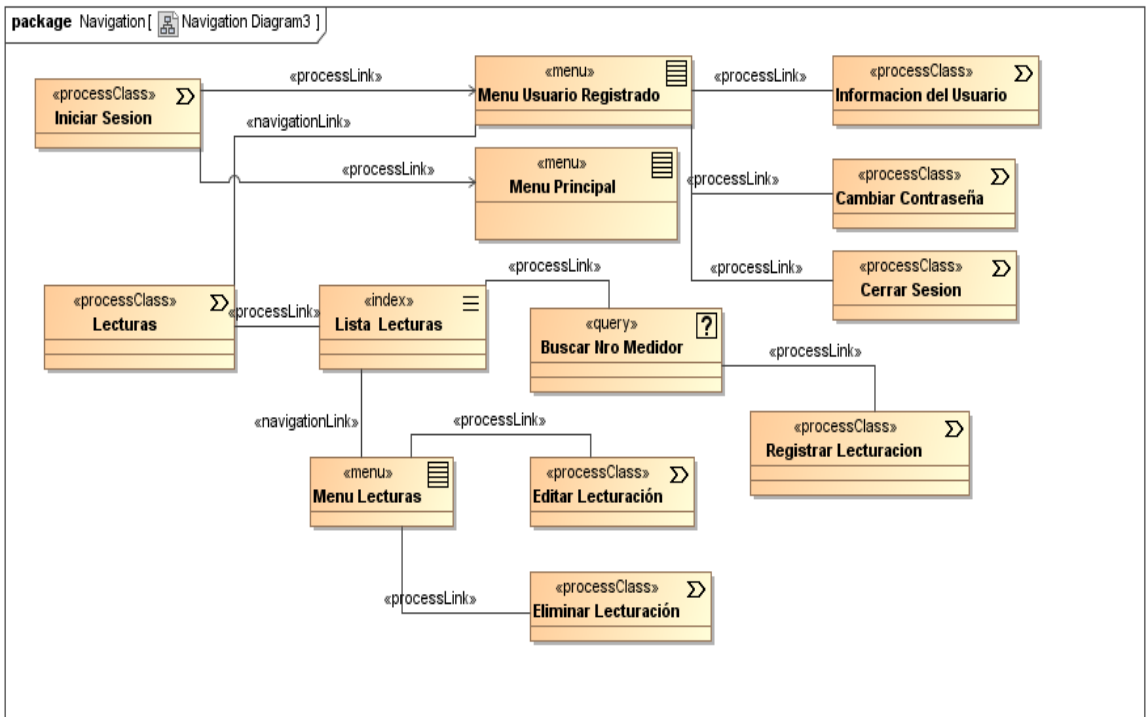


Figura 3.38. Modelo Navegación de Administración de Lecturas
Fuente: (Elaboración propia)

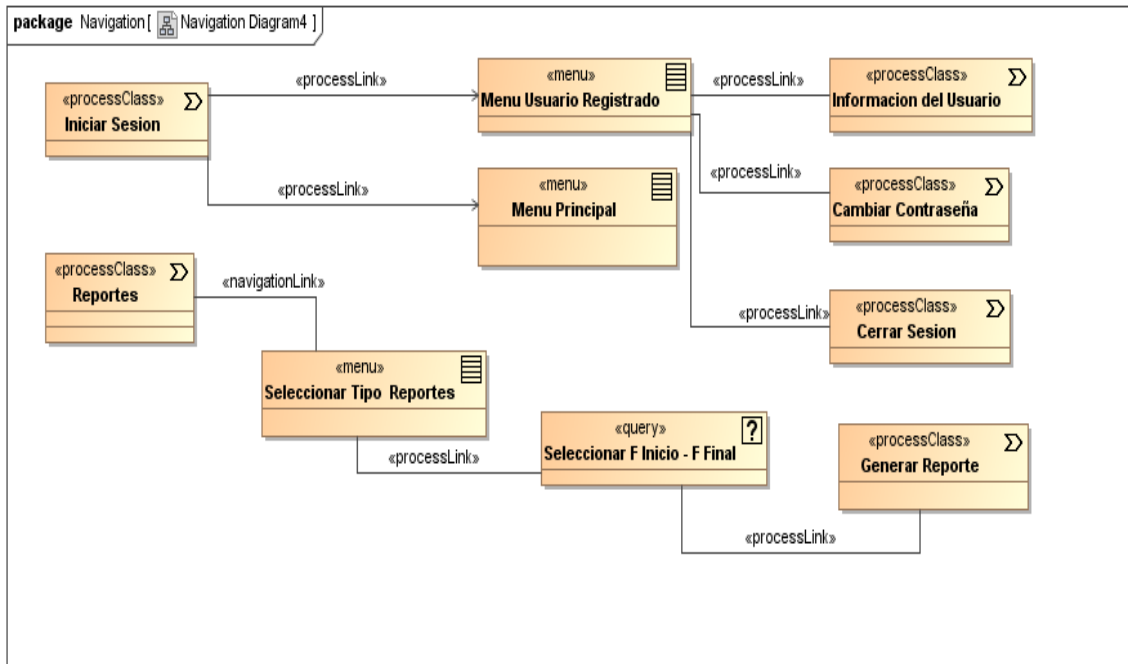


Figura 3.39. Modelo Navegación de Administración de Reportes
Fuente: (Elaboración propia)

3.10. MODELO DE PRESENTACIÓN

El modelo de presentación pretende proporcionar una representación abstracta de la interfaz de usuario final y definir la interacción de las clases navegables con el usuario.

En la figura 3.40 se muestra la presentación de autenticación del usuario, donde en el formulario se deberá ingresar los datos como usuario y la contraseña para luego realizar la verificación con botón ingresar.

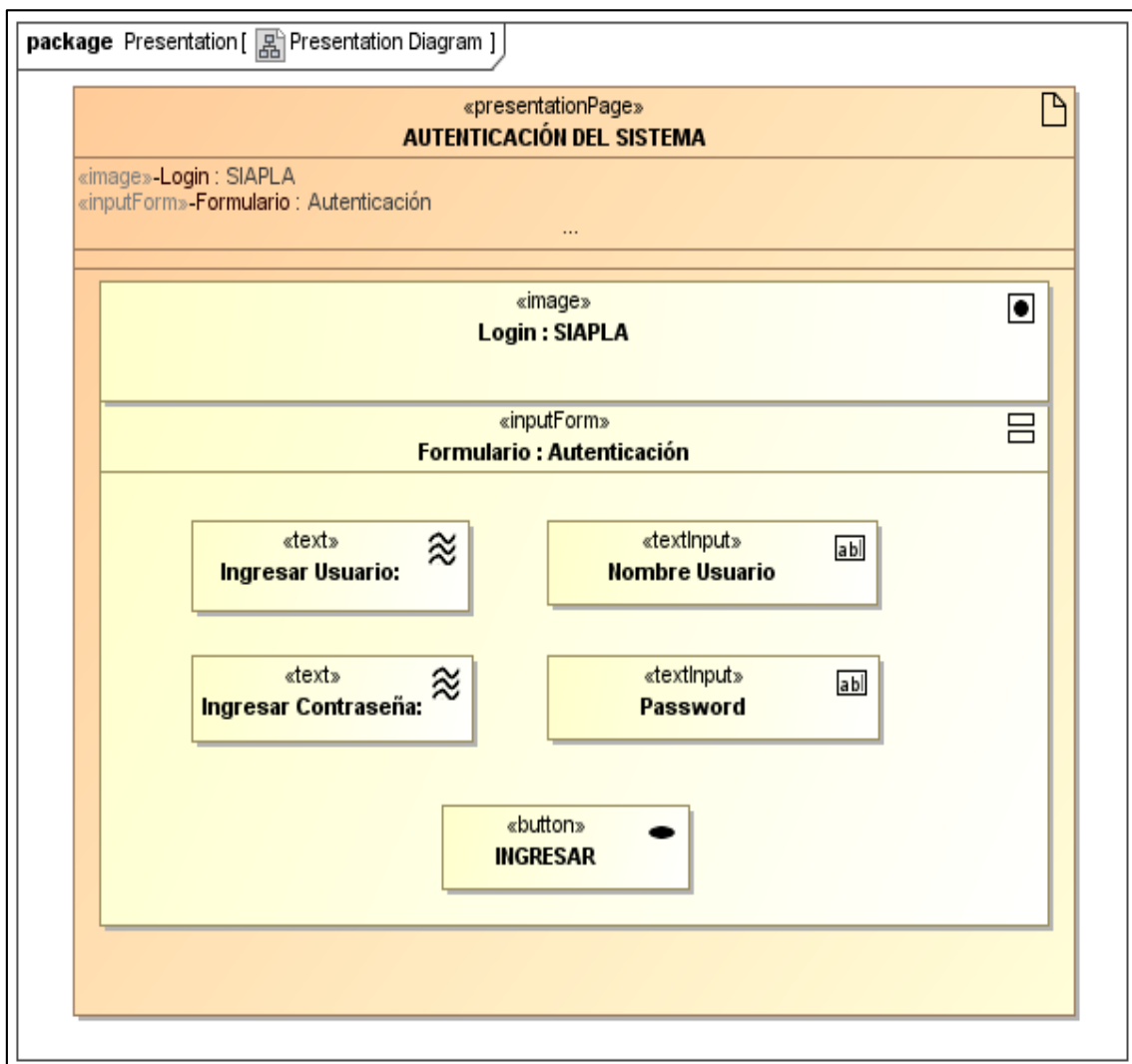


Figura 3.40. Modelo de Presentación de Autenticación del Sistema
Fuente: (Elaboración propia)

En la figura 3.41. se muestra el diseño de inicio del usuario administrador, con los menús correspondientes y privilegios asignados para su interacción con el sistema.

En la figura 3.42. se muestra el diseño de inicio del usuario operador, también con los menús que corresponde y permisos asignados para su interacción con el sistema.

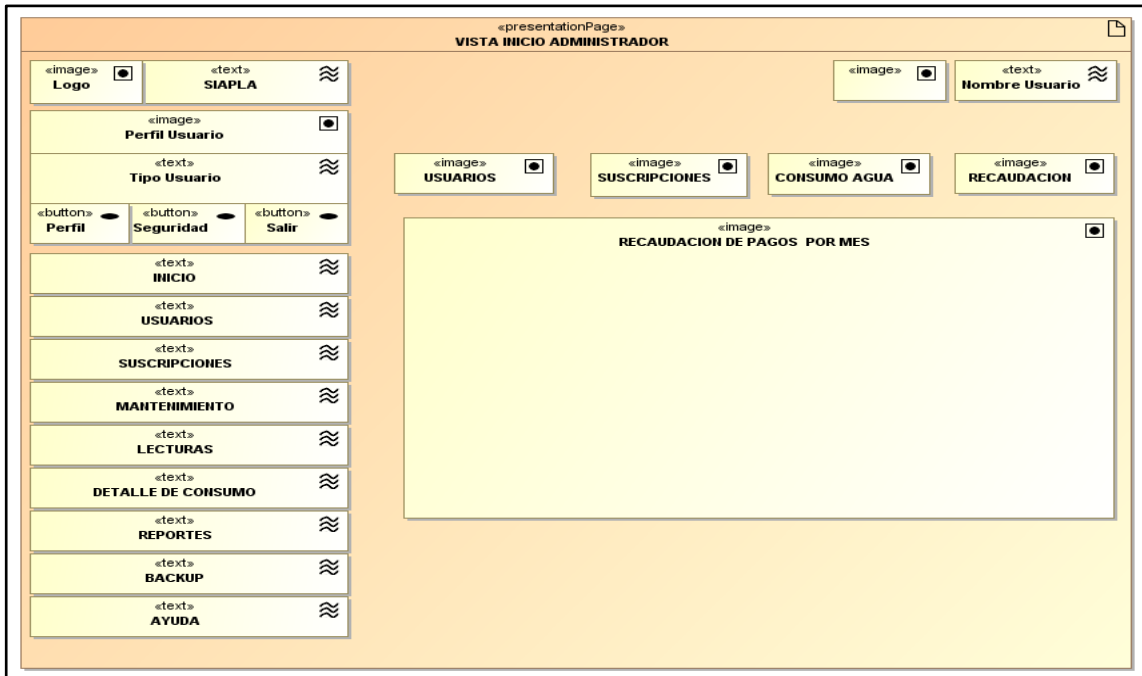


Figura 3.41. Modelo de Presentación de Inicio (Administrador)

Fuente: (Elaboración propia)

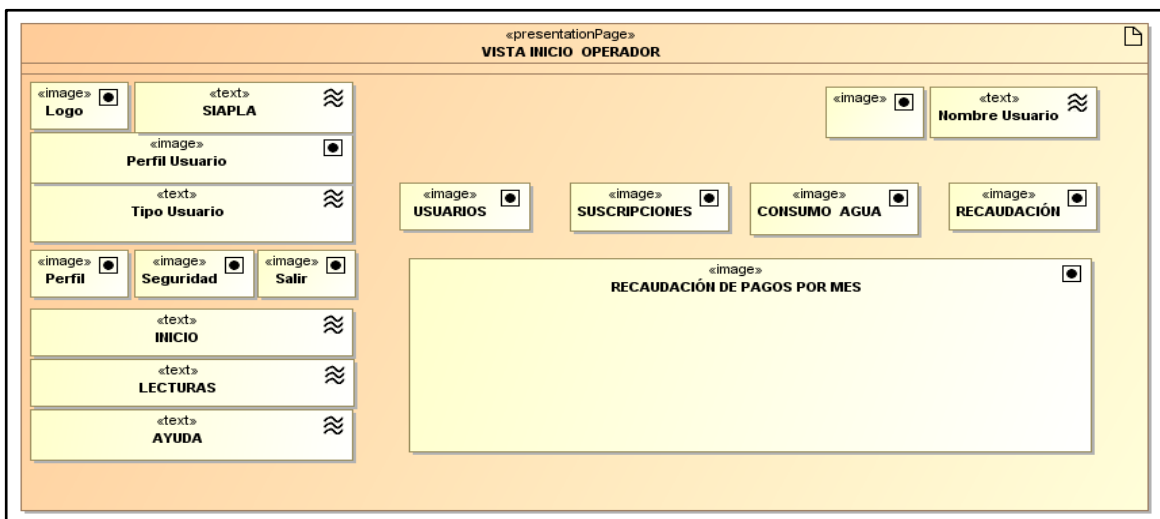


Figura 3.42. Modelo de Presentación de Inicio (Operador)

Fuente: (Elaboración propia)

En la figura 3.43. se muestra la presentación de administración de usuarios y el formulario de registro de nuevos usuarios.

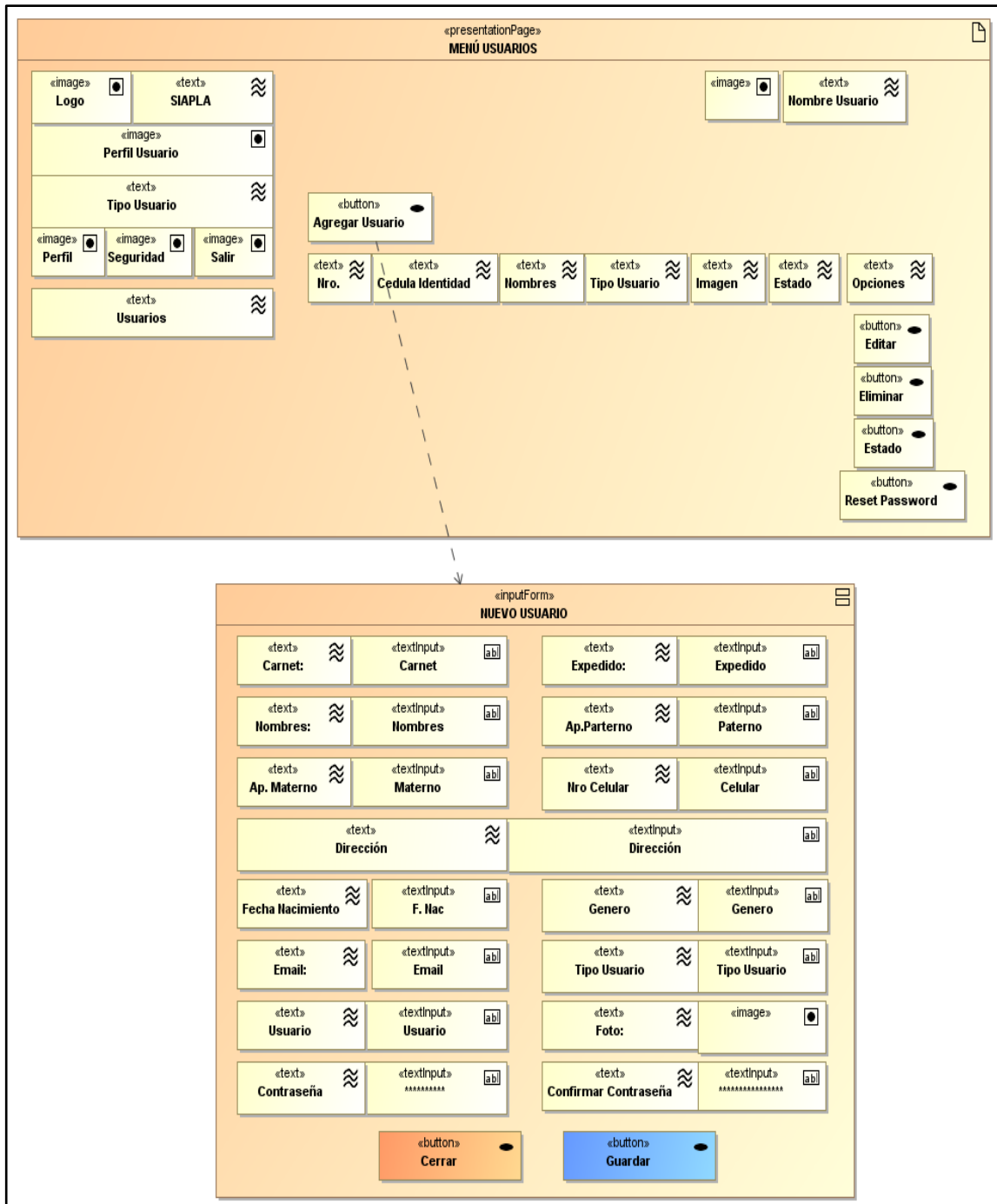


Figura 3.43. Modelo de Presentación Administración de Usuarios (Administrador)
Fuente: (Elaboración propia)

En la figura 3.44. se muestra la presentación de administración de clientes y el formulario de registro de nuevos clientes.

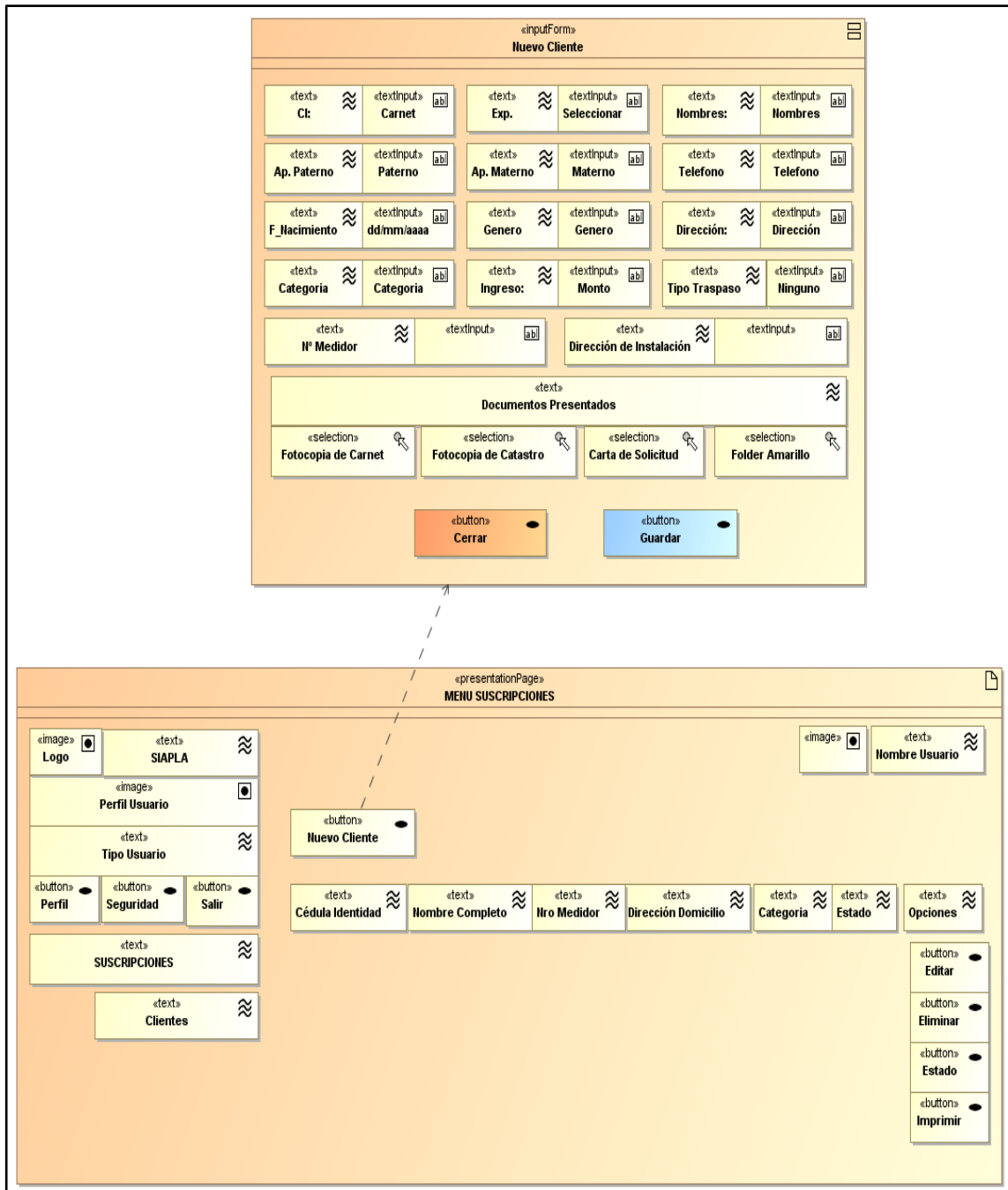


Figura 3.44. Modelo de Presentación Administración de Clientes (Administrador)
Fuente: (Elaboración propia)

En la figura 3.45. se muestra la presentación de administración de lecturas, donde se visualizará el historial de lecturaciones y también el botón agregar lecturas.

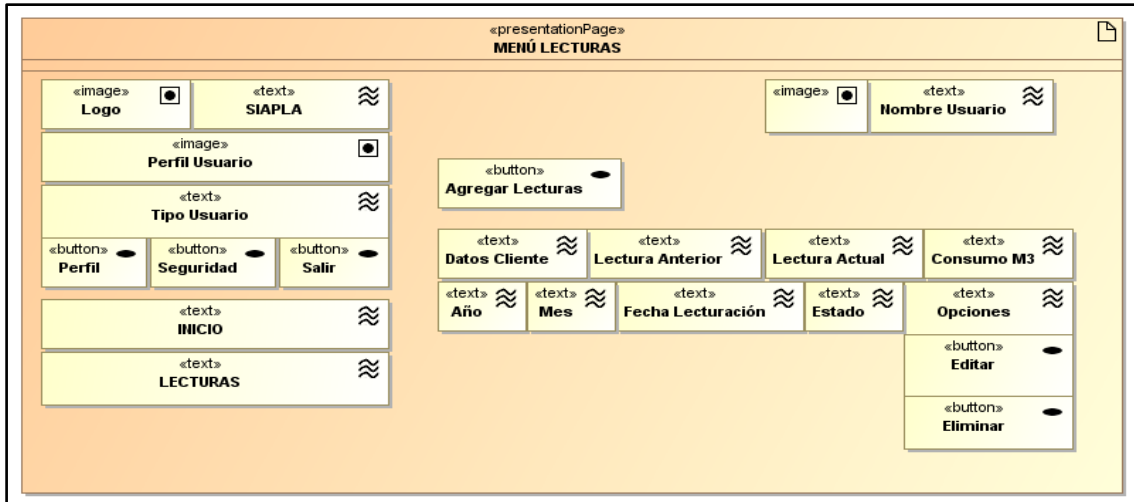


Figura 3.45. Modelo de Presentación Administración de Lecturas (Operador)
Fuente: (Elaboración propia)

En la figura 3.46. se muestra la presentación de administración detalle de consumo, donde se visualizará las listas de consumo por fecha y año, también se podrá realizar el cobro de planillas, finalmente su impresión de la boleta de pagos.

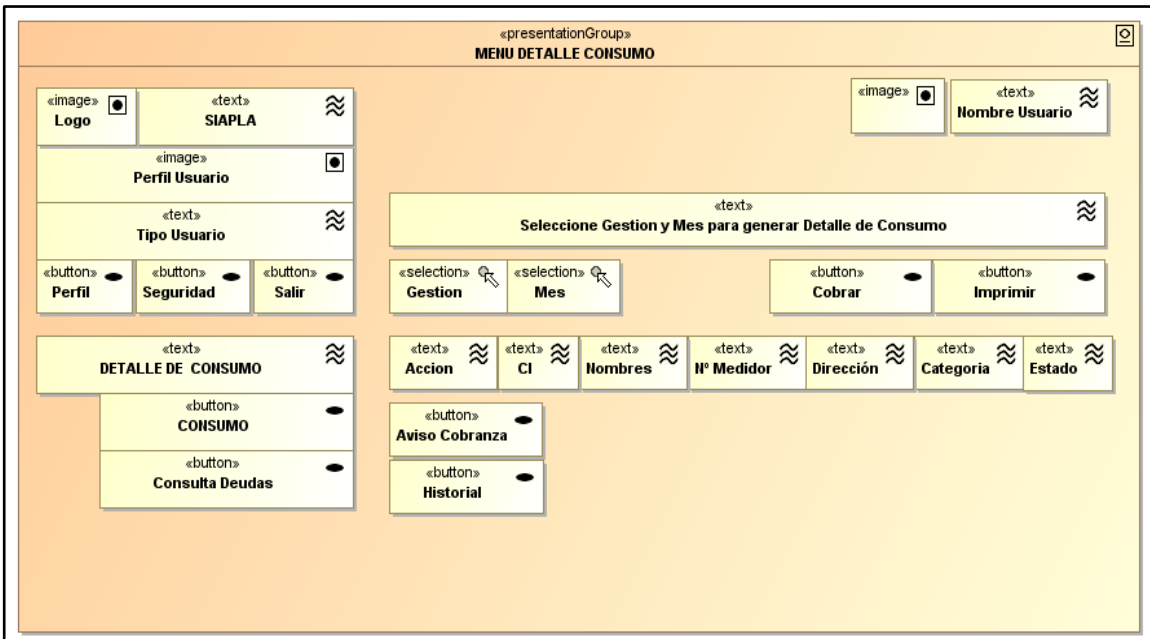


Figura 3.46. Modelo de Presentación Administración de Detalle Consumo (Tesorera)
Fuente: (Elaboración propia)

En la figura 3.47. se muestra la presentación de administración reportes, donde se visualizará tipo de reportes, para que el usuario pueda seleccionar el reporte que requiera.

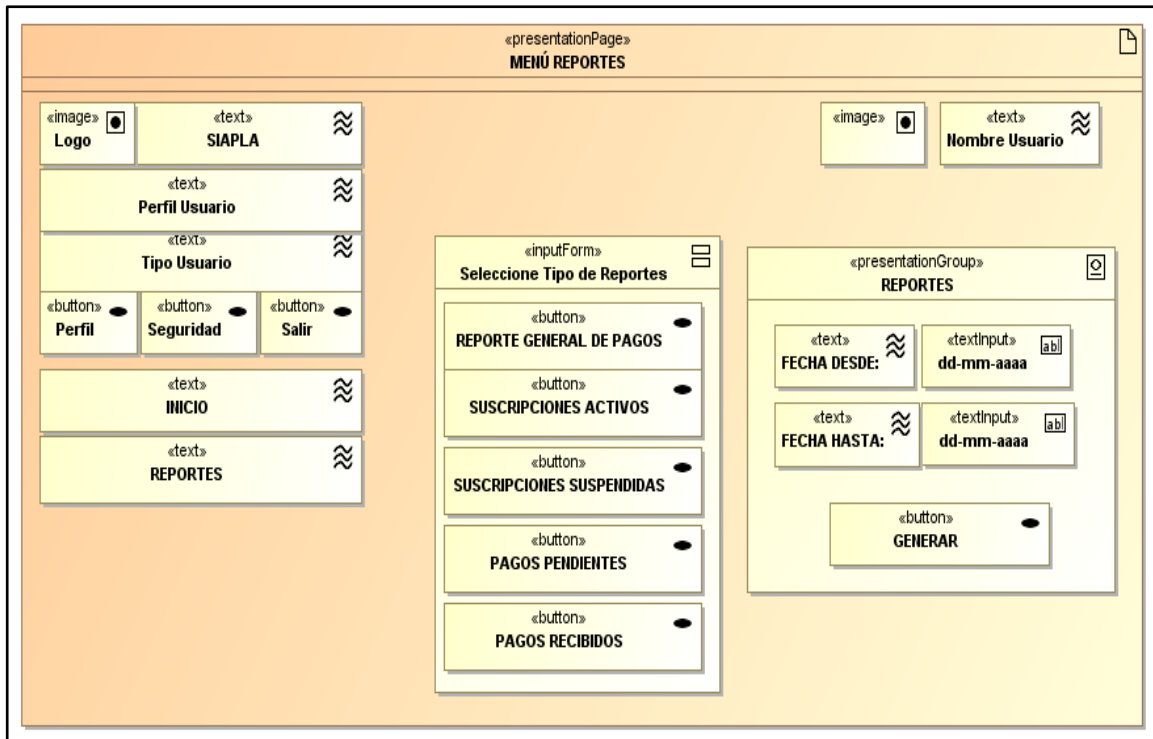


Figura 3.47. Modelo de Presentación Administración de Reportes (Administrador)
Fuente: (Elaboración propia)

3.11. MODELO DE IMPLEMENTACIÓN

En esta fase de implementación consiste en mostrar el desarrollo de la presentación de las interfaces del sistema y sus elementos construidos a partir del diseño del Modelo de Presentación UWE.

3.11.1. Acceso al Sistema

En la figura 3.48. presenta la pantalla autenticación al sistema. En la que el usuario (Administrador, Tesorera y Operador) del sistema, tiene que autenticarse introduciendo su usuario y contraseña.

Los datos serán validados y verificados en la Base de Datos del sistema con la contraseña encriptado.

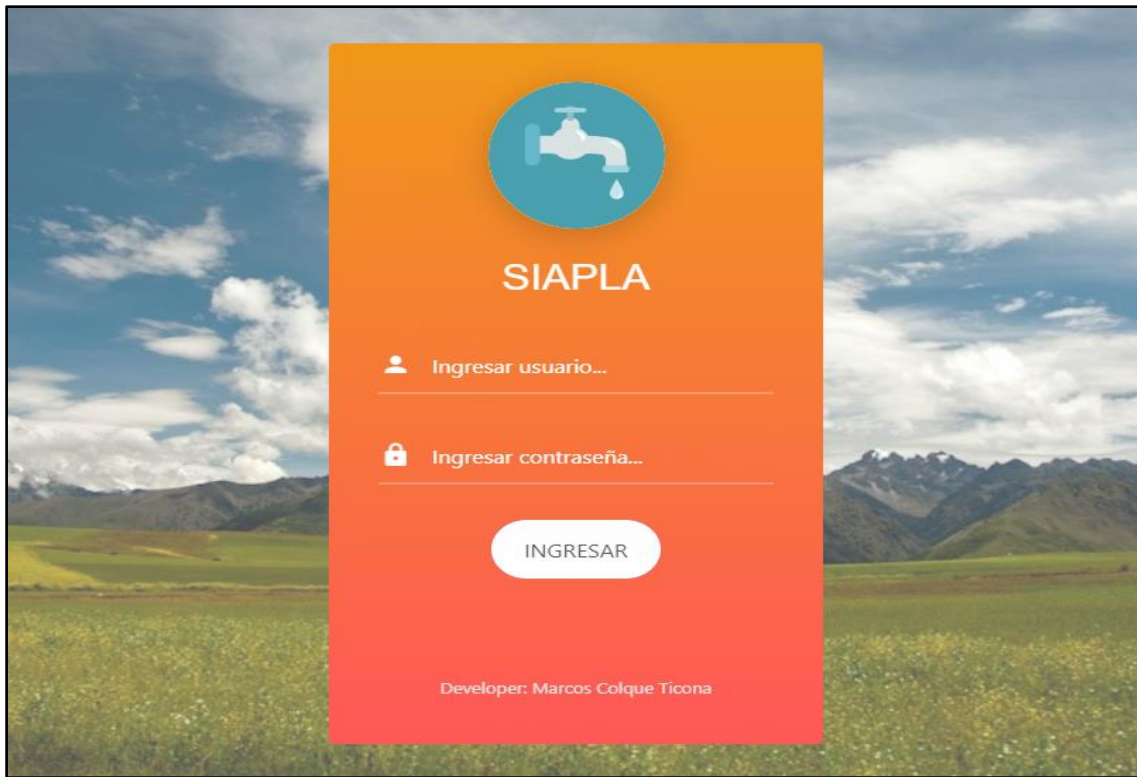


Figura 3.48. Interfaz de Autenticación de Usuario

Fuente: (Elaboración propia)

A continuación, su código HTML de Formulario de Autenticación.

```

<span class="login100-form-title p-b-34 p-t-27" style="font-family: 'Arial',cursive;">
  Siapla
</span>
<div class="wrap-input100 validate-input" data-validate = "Enter username">
<label for="usuarioRegistro"></label>
  <input class="input100" type="text" name="usuario" id="usuario" placeholder="Ingresar usuario..."
  required>
  <span class="focus-input100" data-placeholder="&#xf207;"></span>
</div>
<div class="wrap-input100 validate-input" data-validate="Enter password">
<label for="passwordRegistro"></label>
  <input class="input100" type="password" name="pass" id="pass" placeholder="Ingresar contraseña..."
  required>
  <span class="focus-input100" data-placeholder="&#xf191;"></span>
</div>
<span id="error"></span>
  <input type="hidden" name="valor1" value="<?php echo $token; ?>">
<div class="container-login100-form-btn">
  <button class="login100-form-btn" name="submit" type="submit"> INGRESAR</button>
  ...

```


3.11.2. Menú Principal del Usuario Administrador

En la figura 3.49. donde se muestra el inicio principal del usuario administrador, se puede ver con todo los menús y accesos al sistema, en si es la persona responsable de tener el control total del sistema.

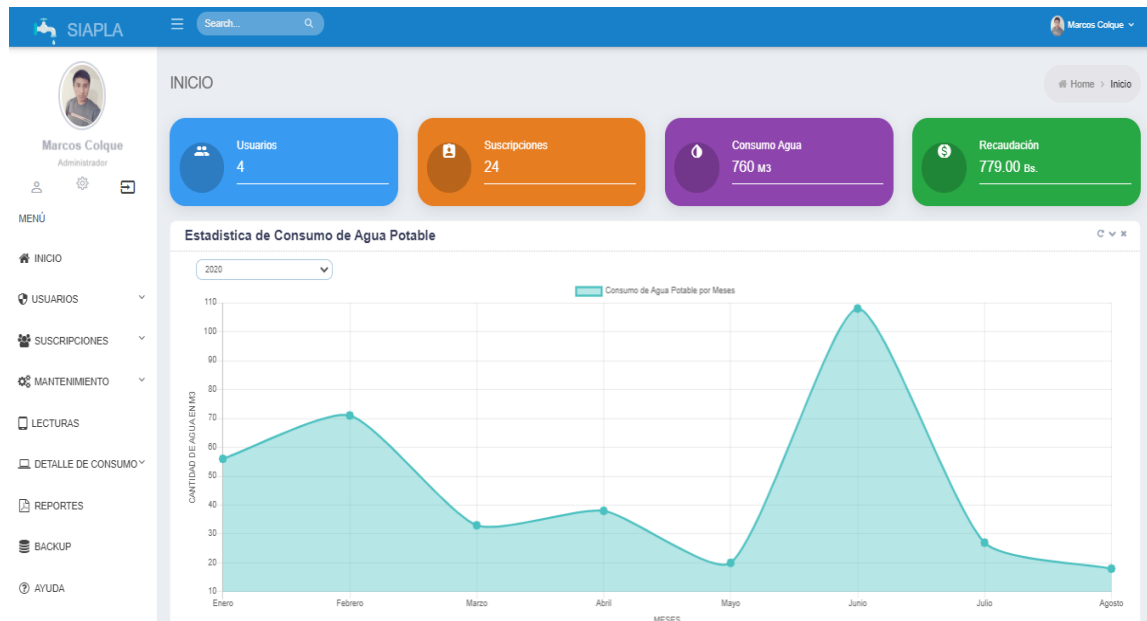


Figura 3.49. Interfaz Menú Principal
Fuente: (Elaboración propia)

A continuación, una parte de código fuente, que permite visualizar los menús de manera dinámica automatizado.

```
<?php if(!empty($menu['admin_usuarios'])) || (!empty($menu['administracion_privilegios'])) ){ ?>
<li class="nav-item <?php if( end($valor)=='admin_usuarios') || (end($valor)=='administracion_privilegios'
) ) echo "active start" ?> " "
<a href="#" class="nav-link nav-toggle">
<i class="material-icons">security</i>
<span class="title">USUARIOS</span>
<span class="selected"></span>
<span class="arrow open"></span>
</a>
<ul class="sub-menu">
<?php if(!empty($menu['admin_usuarios'])) if( $menu['admin_usuarios']=='admin_usuarios'){ ?>
<li class="nav-item <?php if(end($valor)=='admin_usuarios') echo "active" ?>">
<a href="<?php echo base_url(); ?>Controller_sistema/admin_usuarios" class="nav-link nav-toggle">
<i class="fa fa-circle-o"></i>
<span class="title">USUARIOS</span>
</a>
</li>
</ul>
<?php } ?>
<?php if(!empty($menu['administracion_privilegios'])) if( $menu['administracion_privilegios']=='
administracion_privilegios'){ ?>
<li class="nav-item <?php if(end($valor)=='administracion_privilegios') echo "active" ?>">
<a href="<?php echo base_url(); ?>Controller_sistema/administracion_privilegios" class="nav-link
nav-toggle">
<i class="fa fa-circle-o"></i>
<span class="title">PRIVILEGIOS</span>
</a>
</li>
</ul>
<?php } ?>
</li>
<?php } ?>
```

3.11.3. Menú Principal del Usuario Tesorera

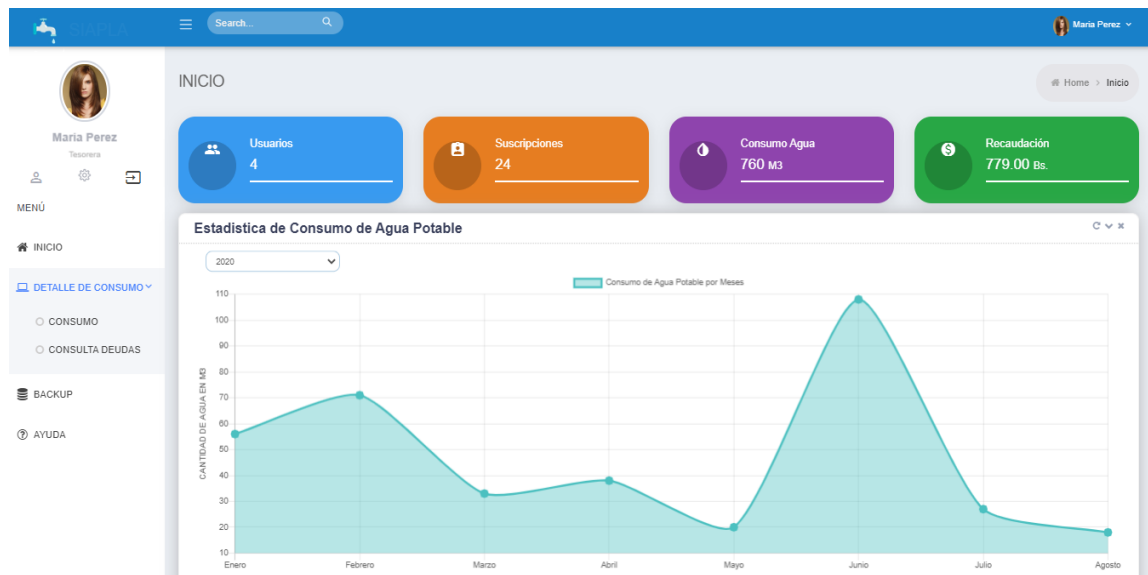


Figura 3.50. Interfaz Menú Principal del Tesorera
Fuente: (Elaboración Propia)

3.11.4. Menú Principal del Usuario Operador

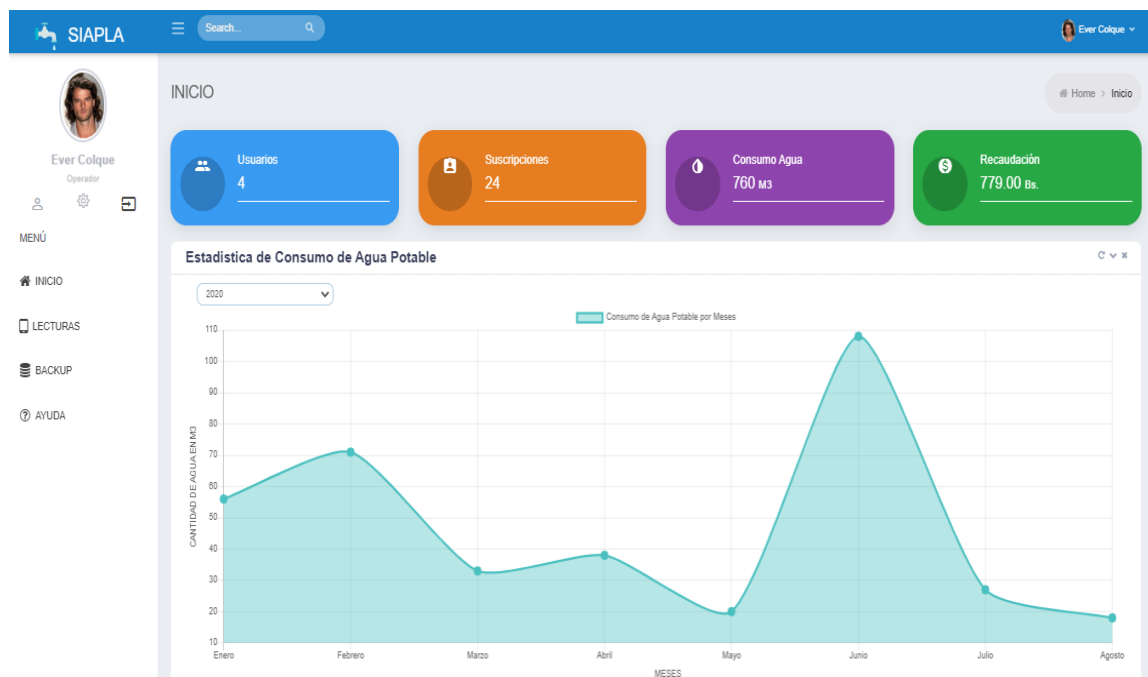


Figura 3.51. Interfaz Menú Principal del Operador
Fuente: (Elaboración propia)

3.11.5. Modulo Usuarios del Usuario Administrador

En la figura 3.52. luego de ingresar el administrador se muestra la interfaz, el Registro de Nuevos Usuarios, para crear nuevo usuario se debe ingresar el número de carnet del usuario a registrar, y de forma automática se mostrará los demás campos requeridos para su registro.

En la figura 3.53. una vez registrado el nuevo usuario se puede visualizar sus datos en una tabla con las opciones de editar, eliminar, cambiar estado y resetear su password.

Figura 3.52. Interfaz Registro de Usuarios

Fuente: (Elaboración propia)

A continuación, su código fuente para verificar su cedula identidad del usuario

```
public function buscar_ci(){
    $obj=$this->Model_Usuario->verificar_carnet($this->input->post('carnet'));
    if ($obj=='0') {
        $this->load->view("administra_usuarios/inserta_persona_usuario_modal");
    }else{
        $objestado=$this->Model_Usuario->verifica_estado_usuario($this->input->post('carnet'));
        if ($objestado==1) {
            echo '<div class="alert label-danger alert-dismissible" role="alert">
                <button type="button" class="close" data-dismiss="alert" aria-label="Close"><span aria-hidden="true"></span></button>
                <strong>!!!El usuario se encuentra registrado</strong>
            </div>';
        }else{
            $data['personas']=$this->Model_Usuario->buscar_persona($this->input->post('carnet'));
            $this->load->view("administra_usuarios/nuevo_usuario_modal",$data);
        }
    }
}
```

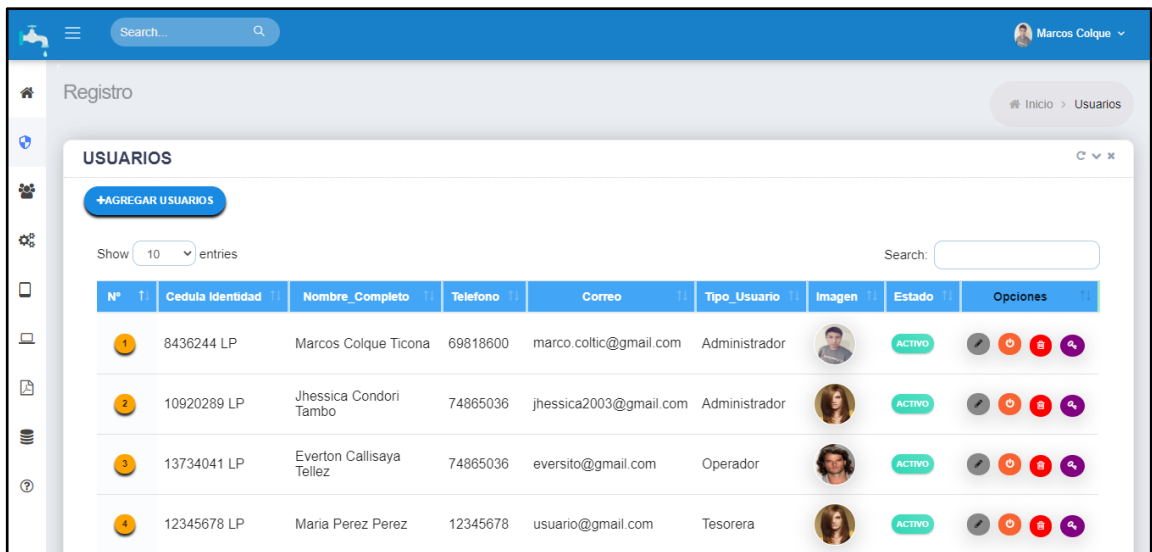


Figura 3.53. Interfaz Administración de Usuarios

Fuente: (Elaboración propia)

A continuación, le mostramos el código fuente en el controlador, como realiza el proceso de obtención de datos del modelo, para luego enviar a la vista.

```

public function admin_usuarios()
{
    if ($this->permisos->tbl_nombre=='admin_usuarios') {
        $data['permisos']=$this->permisos;
        $data['menu']=$this->menu;
        $data['contenido']="administra_usuarios/lista_usuarios";
        $data['usuario']=$this->Model_Usuario->lista_usuario();
        $this->load->view('plantilla',$data);
    }else{
        redirect(base_url().'Controller_sistema/inicio');
    }
}

```

3.11.6. Modulo Clientes del Usuario Administrador

En la figura 3.54. se muestra el formulario para registrar nuevos clientes, donde se deberá asignar un el código de medidor, categoría, ingreso, la dirección donde se va instalar y los documentos a presentar.

En la figura 3.55. se tiene la lista de clientes que han sido registrados, el administrador tendrá las opciones de realizar el de editar, eliminar e imprimir los datos.

NUEVO CLIENTE

● Para registrar los Clientes es necesario que ingrese el Número de Cedula de Identidad del Cliente.

Introduzca Cedula Identidad:

CARNET: <input style="width: 100%;" type="text" value="1234567"/>	EXPEDIDO: <input style="width: 100%;" type="text" value="LP"/>	NOMBRE: <input style="width: 100%;" type="text" value="Nombres..."/>
PATERNO: <input style="width: 100%;" type="text" value="Paterno..."/>	MATERNO: <input style="width: 100%;" type="text" value="Materno..."/>	TELEFONO: <input style="width: 100%;" type="text" value="Telefono..."/>
CELULAR: <input style="width: 100%;" type="text" value="Celular..."/>	F_NACIMIENTO: <input style="width: 100%;" type="text" value="dd/mm/aaaa"/>	GENERO: <input style="width: 100%;" type="text" value="MASCULINO"/>
DIRECCION: <input style="width: 100%; height: 20px;" type="text"/>		

CATEGORIA: <input style="width: 100%;" type="text" value="Domestico"/>	INGRESO: <input style="width: 30%;" type="text" value="\$"/> <input style="width: 30%;" type="text" value="350"/> <input style="width: 30%;" type="text" value=".00"/>	TIPO DE TRASPASO: <input style="width: 100%;" type="text" value="Ninguna"/>
--	--	---

Nº MEDIDOR: <input style="width: 100%;" type="text" value="Numero de Medidor..."/>	DIRECCION DONDE SE VA INSTALAR MEDIDOR: <input style="width: 100%;" type="text" value="Domicilio..."/>
--	--

:: DOCUMENTOS PRESENTADOS ::

Fotocopia de Carnet
 Fotocopia de Catastro
 Carta de Solicitud
 Folder Amarillo

CERRAR
GUARDAR

Figura 3.54. Interfaz de Registro Nuevos Clientes

Fuente: (Elaboración propia)

A continuación, le presentamos una parte de código, donde realiza las validaciones y para luego enviar datos a la vista.

```

public function buscar_cliente()
{
    $obj=$this->Model_cliente->verificar_carnet_cliente($this->input->post("carnet"));
    if ($obj=='0') {
        $data['ingreso']=$this->Model_cliente->costo_ingreso();
        $this->load->view("administra_clientes/form_nuevo_persona_cliente_modal",$data);
    }else{
        $data['personas']=$this->Model_cliente->buscar_persona($this->input->post('carnet'));
        $data['ingreso']=$this->Model_cliente->costo_ingreso();
        $this->load->view("administra_clientes/form_nuevo_cliente_modal",$data);
    }
}

```

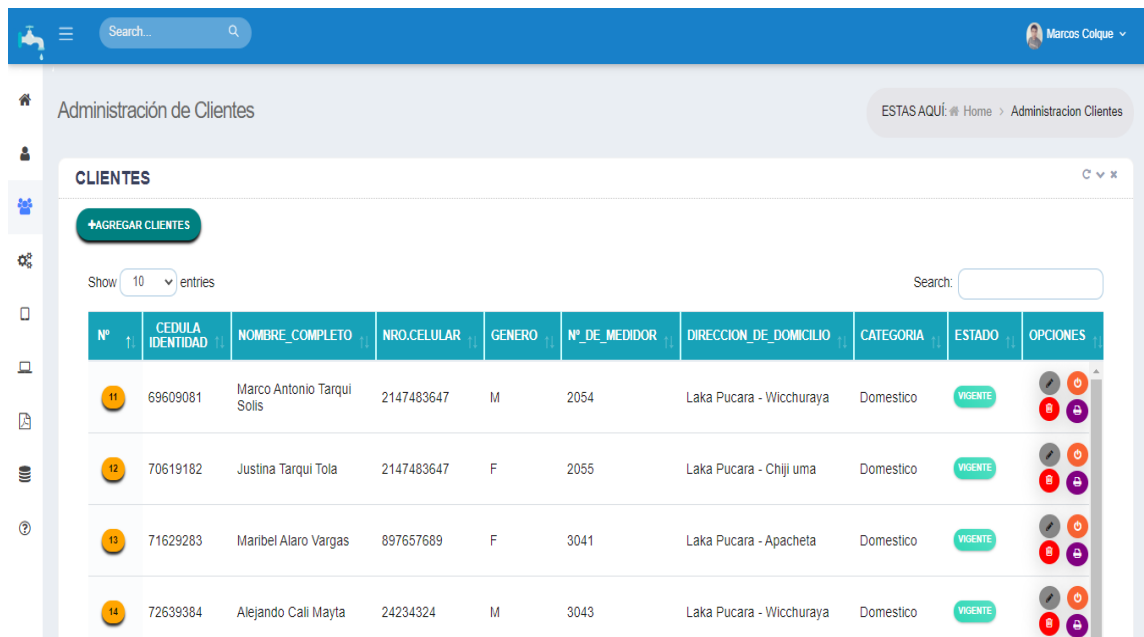


Figura 3.55. Interfaz de administración de clientes
Fuente: (Elaboración propia)

A continuación, comparto el código fuente, donde imprimimos todo el dato recibido del controlador a través de ciclo foreach, para luego mostrar la lista de clientes.

```

<?php $c=1; ?>
<?php if (!empty($cliente)): ?>
<?php foreach ($cliente as $value): ?>
    <tr class="odd gradex">
        <td><span class="btn btn-circle btn-xs" style="background: orange; box-shadow: 0px 3px 3px "><?php
            echo $c ?></span></td>
        <td><?php echo $value->ci ?></td>
        <td><?php echo $value->nombre." ".$value->paterno." ".$value->materno ?></td>
        <td><?php echo $value->celular ?></td>
        <td><?php echo $value->genero ?></td>
        <td><?php echo $value->cl_nro_medidor ?></td>
        <td><?php echo $value->cl_domicilio ?></td>
        <td><?php echo $value->categoria_cliente ?></td>
        <td>
            <?php if($value->cl_estado=="activo"){
                echo "<span class='label label-sm label-success btn-circle'>vigente</span>";
            }else{
                if ($value->cl_estado=="inactivo") {
                    echo "<span class='label label-sm label-danger btn-circle'>suspendido</span>";
                }else{
                    echo "<span class='label label-sm label-warning btn-circle'>trasferido</span>";
                }
            }
        >
    </td>
    <td class="center">
        <?php if ($value->cl_estado=="activo"): ?>
        <a onclick="editar_cliente('<?php echo $value->idcliente ?>')" class="btn btn-tbl-edit btn-xs"
            data-toggle="modal" data-target="#modal-edit">
            <i class="fa fa-pencil"></i>
        </a>
        <button onclick="AlertaActivaDesactiva('<?php echo $value->idcliente; ?>', '<?php echo $value->
            cl_estado; ?>', '<?php echo $value->nombre.' ".$value->paterno.' ".$value->materno ?>')" class="
            btn btn-tbl-delete btn-xs">
            <i class="fa fa-power-off"></i>
        </button>
    </td>
    </tr>
</tr>

```

3.11.7. Módulo de Lecturas del Usuario Operador

En la figura 3.56. permite registrar los datos como el consumo, el mes y el año presionando el botón guardar.

En la figura 3.57. se puede visualizar toda las lecturaciones realizadas a los medidores de cada cliente por el usuario Operador.

REGISTRA LECTURAS

Para registrar la lectura es necesario que ingrese el Numero de Medidor.

Introduzca Numero de Medidor: 2061

Nº MEDIDOR: 2061 CARNET: 77772020

NOMBRE: Juanito PATERNO: Apaza MATERNO: Claros

DIRECCION DE DOMICILIO: Comunidad Janko Marca Sirpa

::: REGISTRA LECTURA DE MEDIDOR :::

CONSUMO: MES: -Mes- GESTION: -Año-

CERRAR GUARDAR

Figura 3.56. Interfaz de Registro de Lecturas.

Fuente: (Elaboración propia)

A continuación, le mostramos código fuente, para el registro de lecturas en el controlador y luego para enviar los datos a modelo.

```
public function registra_lecturas(){
    $gestion=$this->input->post("year");
    $mes=$this->input->post("mes");
    $idcliente=$this->input->post("idcliente");
    $obj=$this->Model_Gestion->verifica_registro_lecturas($idcliente,$mes,$gestion);
    if($obj==1){
        echo "SE ENCUENTRA REGISTRADO LA LECTURA CON ACTUAL MES Y AÑO";
    }else{
        $lectura_a=$this->Model_Gestion->extrae_ultima_lectura_mes($idcliente);
        $lectura_actual=$this->input->post("lectura");
        $lec_anterior_mes=$lectura_a->lectura_antes;
        if ($lectura_actual==$lec_anterior_mes) {
            $consumo=$lectura_actual-($lectura_a->lectura_antes);
            $gestion=$this->input->post("year");
            $mes=$this->input->post("mes");
            $idcliente=$this->input->post("idcliente");
            if (is_null($lectura_a->lectura_antes)) {
                $data=array(
                    'lec_actual'=>$lectura_actual,
                    'lec_anterior'=>$lec_anterior_mes,
                    'lec_consumo'=>$consumo,
                    'lec_gestion'=>$gestion,
                    'lec_mes'=>$mes,
                    'lec_estado'=>'nuevo',
                    'lec_fecha_reg'=>date('Y-m-d'),
                    'lec_id_usuario_sys'=>$this->session->userdata("idusuario"),
                    'idcliente'=>$idcliente
                );
            }
        }
    }
}
```

Registra Lecturas de Medidores

LECTURAS

+AGREGAR LECTURAS

Show 10 entries Search:

Nº	Nº Medidor	Datos de Cliente	Lectura Anterior	Lectura Actual	Consumo M3	Gestion	Mes	Fecha Lectura	Estado	Opciones
21	2052	Maritza Huanca Cayo (66578779 LP)	7	39	32	2019	Agosto	2019-08-31	INACTIVO	[Opciones]
22	2053	Jose Luis Marca Calle (68598980 LP)	9	15	6	2019	Agosto	2019-08-31	ACTIVO	[Opciones]
23	2054	Marco Antonio Tarqui Solis (69609081 LP)	10	15	5	2019	Agosto	2019-08-31	INACTIVO	[Opciones]
24	1031	Juan Apaza Tola (60518172 LP)	73	85	12	2019	Julio	2019-07-27	INACTIVO	[Opciones]

Figura 3.57. Interfaz de Administración de Lecturas
Fuente: (Elaboración propia)

3.11.8. Módulo de Planillas de Consumo del Usuario Tesorera

En la figura 3.58. se visualiza el detalle de consumo, para ver los pagos realizados o pendientes de cada cliente, para ello se debe seleccionar el mes y el año correspondiente, luego se desplegará las listas de clientes, haciendo clic en la pestaña ojos se puede visualizar el aviso de cobranza.

En la figura 3.59. se puede ver el aviso de cobranza con todos los datos del cliente, el consumo de agua correspondiente al mes seleccionada y el historial de pagos de los anteriores meses.

En la figura 3.60. se muestra el formulario de cobranza, con solo ingresando el número de medidor se desplegará los datos del cliente, la planilla de consumo y el costo a cancelar del mes correspondiente.

Finalmente, en la figura 3.61. nos muestra la impresión de la Boleta de Comprobante, al momento de cancelar.

SIAPLA Search... Marcos Colque

Seleccione Gestión y Mes para Generar Detalle de Consumo

GESTION: 2020 MES: Abril COBRAR IMPRIMIR

DETALLE DE CONSUMO

Show 10 entries Search:

	↑ CEDULA IDENTIDAD ↑	↑ NOMBRE_COMPLETO ↑	↑ N° DE MEDIDOR ↑	↑ DIRECCION_DE_DOMICILIO ↑	↑ CATEGORIA ↑	↑ ESTADO ↑
	70619182	Justina Tarqui Tola	2055	Laka Pucara - Chiji uma	Domestico	PAGADO
	71629283	Maribel Alaro Vargas	3041	Laka Pucara - Apacheta	Domestico	PAGADO
	7772020	Juanito Apaza Claros	2061	Comunidad Janko Marca Sirpa	Domestico	PENDIENTE

Figura 3.58. Interfaz de Detalle de Consumo
Fuente: (Elaboración propia)

CAPYSLP
Comité de Agua Potable y Saneamiento Laka Pucara
Janko Marca Sirpa - Comunidad Laka Pucara
Telefono: +58171275469
Corocoro, La Paz - Bolivia

ORIGINAL
Fecha: 2020-06-20
Nro. 7687346583475

AVISO DE COBRANZA

DATOS DEL CLIENTE

NOMBRE: Juanito Apaza Claros
DIRECCIÓN: Comunidad Janko Marca Sirpa

MES/AÑO	NIT/CI	N° MEDIDOR	CATEGORIA
Abril/2020	77772020	2061	Domestico

LECTURA	FECHA	CONSUMO
33	2020-04-06	14
47	2020-05-07	

HISTORIAL				DETALLE	IMPORTE
MES	CONSUMO	MONTO	OBS		
Abr/20	14	15.00		Consumo de Agua Potable	37.00
Mar/20	8	8.00			
Feb/20	12	14.00			
Ene/20	6	8.00	Cancelado		
SON: TREINTA Y SIETE CON 00/100 Bolivianos					37.00

DEUDA A MAYOR A DOS MESES DARA LUGAR A CORTE DE SERVICIO DE AGUA POTABLE

Figura 3.59. Interfaz Aviso de Cobranza
Fuente: (Elaboración propia)

FORMULARIO DE COBRANZA

● Para realizar el cobro correspondiente se debe ingresar el Numero de Medidor del Cliente.
 Introduzca Numero de Medidor:

Nº MEDIDOR:	CARNET:	NUMERO COMPROBANTE:
<input type="text" value="2082"/>	<input type="text" value="77772021"/>	<input type="text" value="000004"/>
NOMBRE:	PATERNO:	MATERNO:
<input type="text" value="Rosa"/>	<input type="text" value="Patzi"/>	<input type="text" value="Tarqui"/>
DIRECCION DE DOMICILIO:		
<input type="text" value="Wicchuraya - Laka Pucara"/>		

PLANILLA DE CONSUMO

Cancelar?	Consumo M3	Gestion	Mes	Fecha_Lectura	Estado	Costo
<input checked="" type="checkbox"/>	7	2020	Enero	2020-02-06	activo	8.00 Bs.
<input checked="" type="checkbox"/>	6	2020	Febrero	2020-03-05	activo	8.00 Bs.
						16.00 Bs.

MONTO A CANCELAR:

DIECISEIS CON 00/100 Bolivianos

CERRAR
COBRAR

Figura 3.60. Interfaz de Formulario de Cobranza
Fuente: (Elaboración propia)

Código fuente que gestiona, las validaciones, como no existe el numero medidor registrado o tiene deudas pendientes, caso contrario nos muestra la planilla de consumo.

```

public function busca_numero_medidor(){
    $obj=$this->Model_Gestion->verifica_medidor($this->input->post("num_medidor"));
    $veri=$this->Model_Gestion->verificar_deuda_medidor_cobros($this->input->post("num_medidor"));
    if ($obj==1) {
        $data['obj1']=$this->Model_Gestion->recupera_datos_cliente($this->input->post("num_medidor"));
        $data['obj2']=$this->Model_Gestion->recupera_datos_cliente_consumo($this->input->post("num_medidor"));
        $data['obj3']=$this->Model_Gestion->extrae_maximo_numcomprobante();
        $this->load->view("administra_detalle_consumo/form_cobros_consumo",$data);
    }else{
        if (empty($veri)) {
            echo '<div class="alert alert-warning" style="color: #000; font-size: 18px;">
                <i class="fa fa-check-circle"></i><strong></strong>NO EXISTE EL NUMERO DE MEDIDOR INGRESADO.
            </div>';
        }else{
            echo '<div class="alert alert-primary " style="background: #ec7063; color: #fff; font-size: 15px;">
                <i class="fa fa-question-circle"></i><strong></strong> El numero de medidor introducido tiene deuda pendiente,
                primero debe regularizar sus deudas.
            </div>';
        }
    }
}

```

Finalmente, con la función guardar_nueva_cobro, recibimos los datos mediante post, luego enviamos al modelo para ser guardado físicamente los datos.

```

public function guardar_nueva_cobro(){
    $fecha=date('Y-m-d');
    $numcomprobante=$this->input->post("comprobante");
    $id_usuario=$this->session->userdata("idusuario");
    $idlectura_cl_medidor=$this->input->post('idlectura_cl_medidor');
    $costo_t=$this->input->post('costo_t');
    $idcosto_de_cobranza=$this->input->post('idcosto_de_cobranza');
    $this->Model_Gestion->guardar_nueva_cobro($fecha,$numcomprobante,$id_usuario,$idlectura_cl_medidor,$costo_t,$idcosto_de_cobranza);
    echo "SE REGISTRO CORRECTAMENTE";
}


```

CAPYSLP
 Comité de Agua Potable Laka Pucara
 La Paz - Bolivia
 Teléfono: +58169818600

ORIGINAL
 Fecha: 2020-07-03
 Número: 1008

BOLETA DE COMPROBANTE DE PAGOS

DATOS DEL CLIENTE
 NOMBRE: Justina Tarqui Tola
 DIRECCION: Laka Pucara - Chiji uma
 CATEGORIA: Domestico
 N° MEDIDOR: 2055



DETALLE CONSUMO					
N°	Consumo M3	Mes/Año	Fecha de Pago	Descripción	Importe
1	27	Julio/2020	2020-07-03	cancelado	21.50
2	18	Agosto/2020	2020-07-03	cancelado	17.00
Total:					38.50

SON: TREINTA Y OCHO CON 50/100 Bolivianos

Figura 3.61. Interfaz de Boleta Comprobante de Pagos
 Fuente: (Elaboración propia)

A continuación, les presento código fuente para generar Boleta Comprobante con librería FPDF.

```

$pdf->Cell(195,5,'DETALLE CONSUMO',1,1,'C',1);
$pdf->SetFont('COURIER','B',10);
$pdf->SetFillColor(178,223,219);
$pdf->SetTextColor(0,0,0);
$pdf->SetDrawColor(8,8,8);
$pdf->Cell(15,5,'Nº',1,0,'C',1);
$pdf->Cell(30,5,'Consumo M3',1,0,'C',1);
$pdf->Cell(35,5,'Mes/Año',1,0,'C',1);
$pdf->Cell(45,5,'Fecha de Pago',1,0,'C',1);
$pdf->Cell(35,5,'Descripción',1,0,'C',1);
$pdf->Cell(35,5,'Importe',1,1,'R',1);
$c=0;$total=0;
$pdf->SetTextColor(0,0,0);
foreach ($obj as $value) {
    $pdf->Cell(15,5,$c=$c+1,1,0,'C');
    $pdf->Cell(30,5,$value->lec_consumo,1,0,'C');
    $pdf->Cell(35,5,$value->lec_mes."/".$value->lec_gestion,1,0,'C');
    $pdf->Cell(45,5,$value->con_fecha_reg,1,0,'C');
    $pdf->Cell(35,5,$value->con_estado,1,0,'C');
    $pdf->Cell(35,5,$value->con_costo_total_mes,1,1,'R');
    $total=$total+$value->con_costo_total_mes;
}
$pdf->Cell(160,5,'Total:',0,0,'R');
$pdf->Cell(35,5,number_format($total,2),1,1,'R');
//Footer de la boleta
$pdf->Cell(200,5,'',0,1);
$pdf->SetFont('TIMES','BI',10);
$pdf->Cell(195,5,convertir1($total),0,0,'L');
$pdf->OutPut();

```

3.11.9. Módulo de Reportes del Usuario Administrador

En la figura 3.62. nos muestra la pantalla de reportes donde podemos seleccionar el tipo de reportes que sea requerido del usuario.

En la figura 3.63. se puede ver el informe de recaudaciones anual, tabulados por meses y el total recaudado.



Figura 3.62. Interfaz de Reportes

Fuente: (Elaboración propia)

Una parte de código de Ajax, para realizar la petición al controlador, luego una vez que devuelva la petición, se mostrara los datos en #ver_tabla

```
<script type="text/javascript">
function mostrarResultadosReportes(){
var anualSelect=document.getElementById('anual').value;
if (anualSelect!='0') {
$.ajax({
url: '<?php echo base_url() ?>Controller_reporte/selecciona_datos_recaudacion',
type: 'POST',
data: {anual: anualSelect},
success:function(dat){
$("#ver_tabla").html(dat);
}
})
}else{
$("#ver_tabla").html('');
}
}
</script>
```

CONTROL DE PAGO DE TARIFAS DE AGUA POTABLE GESTION 2019



N°	NOMBRE USUARIO	N° MEDIDOR	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	Nov	Dic	Total Bs.
1	Huayta Mamani Elmer	1030	8.00	8.00	8.00	13.50	8.00	8.00	8.00	8.00					69.50
2	Apaza Tola Juan	1031	15.50	8.00	19.00	8.00	14.50	8.00	14.00	14.00	8.00	22.00	8.00	8.00	147.00
3	Ticona Marca Alberto	1032	8.00	8.00	8.00	8.00	8.00	8.00	8.00	8.00					64.00
4	Canaviri Reyes Fausto	1033	8.00	8.00	8.00	8.00	8.00	13.50	8.00	8.00					69.50
5	Quenta Churqui Carlos	1034	8.00	8.00	8.00	8.00	8.00	8.00	8.00	8.00					64.00
6	Colque Cala Magdalena	1035							8.00	15.00					23.00
7	Alcocer Teran Fernando	2051							13.50						13.50
8	Huanca Cayo Maritza	2052							8.00	24.00					32.00
9	Marca Calle Jose Luis	2053							8.00						8.00
10	Tarqui Solis Marco Antonio	2054							8.00	8.00					16.00
	TOTALES POR MESES		47.50	40.00	51.00	45.50	46.50	45.50	91.50	93.00	8.00	22.00	8.00	8.00	506.50

Figura 3.63. Informe de Recaudaciones por Meses

Fuente: (Elaboración propia)

El código fuente permite sumar todo el cobro por meses, finalmente el sumatorio total recaudado anual.

```

$total_recaudado_anual=$this->db->query("select sum(consumo_cobransa.con_costo_total_mes) as suma_total_anual from persona inner
join cliente on persona.idpersona=cliente.idpersona inner join lectura_cl_medidor on
cliente.idcliente=lectura_cl_medidor.idcliente inner join consumo_cobransa on
consumo_cobransa.idlectura_cl_medidor=lectura_cl_medidor.idlectura_cl_medidor where lectura_cl_medidor.lec_gestion='$gestion'")
->row();
if (empty($total_recaudado_anual->suma_total_anual)) {
    $total_anual=0;
}else{
    $total_anual=$total_recaudado_anual->suma_total_anual;
}
$fpdf->SetLineWidth(0.5);
$fpdf->Cell(97,5,'TOTALES POR MESES',1,0,'C');
$fpdf->Cell(12,5,$total_enero,1,0,'C');
$fpdf->Cell(12,5,$total_febrero,1,0,'C');
$fpdf->Cell(12,5,$total_marzo,1,0,'C');
$fpdf->Cell(12,5,$total_abril,1,0,'C');
$fpdf->Cell(12,5,$total_mayo,1,0,'C');
$fpdf->Cell(12,5,$total_junio,1,0,'C');
$fpdf->Cell(12,5,$total_julio,1,0,'C');
$fpdf->Cell(12,5,$total_agosto,1,0,'C');
$fpdf->Cell(12,5,$total_septiembre,1,0,'C');
$fpdf->Cell(12,5,$total_octubre,1,0,'C');
$fpdf->Cell(12,5,$total_noviembre,1,0,'C');
$fpdf->Cell(12,5,$total_diciembre,1,0,'C');
$fpdf->Cell(20,5,$total_anual,1,1,'C');

$fpdf->OutPut('Pagos_Anual.pdf','I');
}

```

3.12. CONFIGURACIÓN DE LA RED DE COMPUTADORAS

Para acceder a un servicio de acceso a internet a través de conectividad satelital, se debe disponer de ciertos requisitos y usarla para acceder a la red de computadoras o para establecer zonas de wifi.

Primero se debe instalar la antena en un punto con vista hacia el satélite, luego se hacen las conexiones, se programa el modem y se da de alta el servicio como pueden ver en la figura 3.64.



Figura 3.64. Router wifi para Adsl Telefónica

Para poder configurar la red inalámbrica, necesitas lo siguiente:

Conexión a Internet de banda ancha y módem. Una conexión a Internet de banda ancha es una conexión a Internet a alta velocidad. ADSL (línea de suscriptor digital) y cable son dos de las conexiones de banda ancha más comunes. Puedes obtener una conexión de banda ancha poniéndote en contacto con un proveedor de acceso a Internet (ISP). Normalmente, los ISP que ofrecen ADSL son compañías telefónicas y los ISP que ofrecen cable son compañías de TV por cable. Con frecuencia, los ISP ofrecen módems de banda ancha. Algunos ISP también ofrecen una combinación de módem y router inalámbrico.

Enrutador inalámbrico. Un enrutador envía información entre la red e Internet. Con un enrutador inalámbrico, puedes conectar equipos a la red mediante señales de radio en lugar de cables. Hay varios tipos de tecnologías de red inalámbrica, entre las que se incluyen 802.11a, 802.11b, 802.11g, 802.11n y 802.11ac.

Adaptador de red inalámbrica. Un adaptador de red inalámbrica es un dispositivo que conecta el equipo a una red inalámbrica. Para conectar el equipo portátil o de escritorio a tu red inalámbrica, el equipo debe tener un adaptador de red inalámbrica. La mayoría de los portátiles y tabletas (y algunos equipos de escritorio), se distribuyen con un adaptador de red inalámbrica ya instalado.

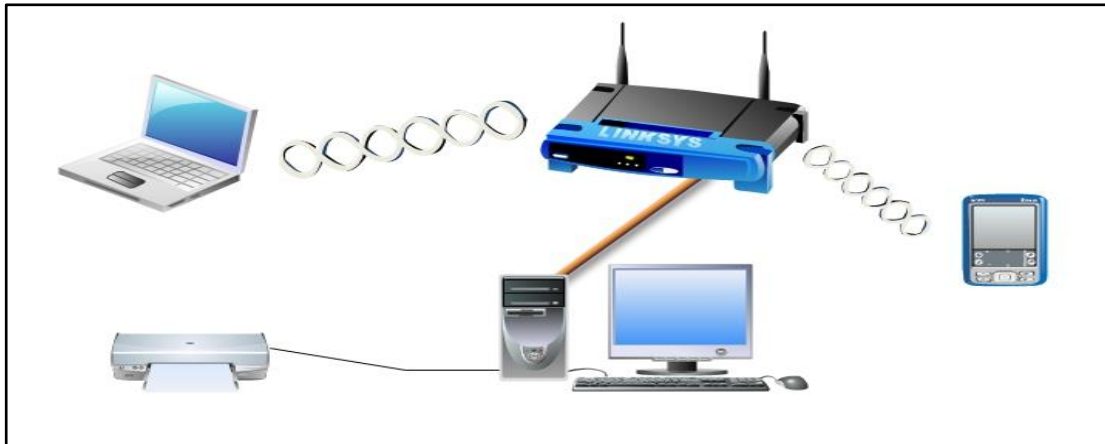


Figura 3.65. Implementación de Red inalámbrica
Fuente: Elaboración Propia

Conectar un equipo a la red inalámbrica

1. Selecciona el icono Red o en el área de notificación.
2. En la lista de redes, elige la red a la que quieras conectarte y, a continuación, selecciona **Conectar**.
3. Escribe la clave de seguridad (denominada normalmente contraseña).
4. Sigue las instrucciones adicionales, si las hubiera.

Figura 3.66. Instrucciones para Conectarse desde un Notebook
Fuente: (Microsoft, 2017)



CAPITULO IV

MÉTRICAS DE CALIDAD Y COSTOS

4.1. MÉTRICAS DE CALIDAD DE SOFTWARE

Desarrollar un software de calidad, es el objetivo de todo desarrollador, por tanto, se le dedica muchos esfuerzos, pero también cabe mencionar que no se logra la perfección en el producto de software, pero se debe tomar en cuenta que todo software debe cumplir y superar las expectativas del cliente.

En la actualidad existen varias técnicas, los estándares y modelos de evaluación para medir la calidad de software. El ISO 9126 es un estándar que especifica los siguientes factores de calidad:

- **Funcionalidad:** permiten calificar si un producto de software maneja en forma adecuada el conjunto de funciones que satisfagan las necesidades para las cuales fue diseñado.
- **Confiabilidad:** Se refiere a la capacidad del software de mantener su nivel de ejecución bajo condiciones normales en un periodo de tiempo establecido.
- **Usabilidad:** Consiste de un conjunto de atributos que permiten evaluar el esfuerzo necesario que deberá invertir el usuario para utilizar el sistema.
- **Eficiencia:** permite evaluar la relación entre el nivel de funcionamiento del software y la cantidad de recursos usados.
- **Mantenibilidad:** se refiere a los atributos que permiten medir el esfuerzo necesario para realizar modificaciones al software, ya sea por la corrección de errores o por el incremento de funcionalidad.
- **Portabilidad:** se refiere a la habilidad del software de ser transferido de un ambiente a otro.

4.1.1. Funcionalidad

Para el cálculo de punto de función se toma en cuenta 5 características que se detallan a continuación.

- **Número de Entradas de Usuario:** El número de entradas de usuario proporciona datos al sistema. Esto para poder realizar las distintas operaciones tales como ser: altas, bajas y modificaciones.

Tabla 4.15.
Entrada de Usuario

Nº	Entrada de Usuario
1	Registro de Usuarios
2	Registro de Clientes
3	Registro de Lecturas
4	Registro de Transferencias
5	Registro de Tarifas
6	Registro de Planillas
7	Registro de Pagos
8	Registro de Multas

Fuente: (Elaboración Propia)

- **Número de Salidas de Usuario:** una salida de usuario es aquella que proporciona información elaborada por el sistema que son transmitidas al usuario. La salida se refiere a informes, pantallas, mensajes de error e alertas de confirmación.

Tabla 4.16.
Salidas de Usuario

Nº	Salidas de Usuario
1	Confirmar datos de usuario
2	Reporte de clientes registrados
3	Reporte de planilla de cobros
4	Informe general de recaudaciones
5	Reporte de multas
6	Reporte de comprobante de pago
7	Reporte de datos estadísticos

Fuente: (Elaboración Propia)

- **Número de peticiones de usuario:** una petición está definida como una entrada interactiva que resulta de la generación de algún tipo de respuesta en forma de salida interactiva.

Tabla 4.17.
Peticiones de Usuario

Nº	Peticiones de Usuario
1	Autenticación de usuario
2	Lista de usuarios registrados
3	Lista de clientes
4	Lista de tarifas
5	Lista de lecturas
6	Lista de planillas
7	Lista de cobranzas
8	Lista traspasos de clientes
9	Datos estadísticos

Fuente: (Elaboración Propia)

- **Numero de archivos:** se cuenta cada archivo maestro lógico. En otras palabras, las tablas existentes en la base de datos.

Tabla 4.18.
Numero de archivos

Nº	Numero de Archivos
1	Modulo menú
2	Privilegio
3	Usuario
4	Tipo usuario
5	Persona
6	Cliente
7	Trasferencia cliente
8	Categoría
9	Lectura de medidor
10	Tarifas
11	Consumo cobranza
12	Multas
13	Corte de Servicio

Fuente: (Elaboración Propia)

- **Numero de interfaces externas:** se cuenta todas las interfaces legibles por el ordenador que son utilizados para transmitir la información.

Tabla 4.19.
Numero de Interfaces Externas

Nº	Interfaces Externas
1	Internet
2	Intranet

Fuente: (Elaboración Propia)

La funcionalidad es medida a través del punto de función(PF), que proporciona una medida objetiva, cuantitativa y auditable del tamaño de la aplicación, basada en la visión del usuario de la aplicación.

La tabla 4.20. muestra la cuenta total de los dominios de la información establecidos en el sistema web de acuerdo a los parámetros de medición.

Tabla 4.20.
Parámetros de medición según Punto de Función

PARÁMETRO DE MEDICIÓN	Factor de Ponderación				TOTAL
	CUENTA	SIMPLE	MEDIO	COMPLEJO	
Nº de entradas de usuario	8	3	4	6	32
Nº de salidas de usuario	7	4	5	7	35
Nº de peticiones de usuario	9	3	4	6	36
Nº de archivos	13	7	10	15	130
Nº de interfaces externas	2	5	7	10	14
Cuenta Total					247

Fuente: (Elaboración Propia)

La relación que permite calcular los puntos de función es la siguiente:

$$PF = cuenta\ total * (grado\ de\ confiabilidad + tasa\ de\ error * \sum fi)$$

Donde:

PF=Medida de Funcionalidad

Cuenta Total = Suma de valor de las entradas, salidas, peticiones, archivos e interfaces externas.

Grado de Confiabilidad = Confiabilidad estimada del sistema

Tasa de Error = Probabilidad subjetiva estimada del dominio de la información, este error estimado es del 1%.

Fi = Valores de ajuste de complejidad que toman los valores de la tabla.

Calculo de punto función ajustada. Los valores de Fi, se obtiene de los resultados de la siguiente tabla 4.21. bajo las ponderaciones descritos en la escala.

Tabla 4.21.
Ponderación de Ajuste de Complejidad

		Factor de Complejidad						
		0	1	2	3	4	5	
Nº	Escala	Sin Importancia	Incidental	Moderado	Medio	Significante	Esencial	Fi
1	¿Requiere el sistema copias de seguridad y de recuperación fiable?						x	5
2	¿Se requiere comunicación de datos?						x	5
3	¿Existe funciones de procesos distribuidos?					x		3
4	¿Es critico el rendimiento?					x		3

5	¿Se actualiza los archivos de forma interactiva?	x	5
6	¿Es complejo el procesamiento interno del sistema?	x	4
7	¿Se ha diseñado el código para ser reutilizado?	x	5
8	¿Sera ejecutado el sistema en SO existente?	x	4
9	¿Se ha diseñado el sistema para soportar múltiples instalaciones?	x	4
10	¿ Se tiene facilidad de modificaciones ?	x	3
11	¿Entrada de datos en línea?	x	5
12	¿Se ha diseñado el sistema para facilitar al usuario el trabajo y encontrar la información ?	x	5
Factor de Complejidad Total (fi)			51

Fuente: (Elaboración Propia)

Con la obtención de los anteriores datos y considerando un grado de confiabilidad del 65% a continuación calculamos el valor del PF:

$$PF = cuenta\ total * (grado\ de\ confiabilidad + tasa\ de\ error * \sum fi)$$

$$PF = 247 * (0.65 + 0.01 * 51)$$

$$PF = 286.52$$

Si consideramos el máximo valor de ajuste de complejidad como $\sum fi = 60$, se tiene:

$$PF = 247 * (0.65 + 0.01 * 60)$$

$$PF_{máximo} = 308.75$$

Entonces si $\sum fi$ es considerada como el 100%, la relación obtenida entre los puntos será.

$$\frac{PF}{PF_{máximo}} = \frac{279.11}{308.75} * 100$$

FUNCIONALIDAD = 90%

Por lo tanto la funcionalidad del sistema es del 90% tomando en cuenta el punto de función máximo.

4.1.2. Confiabilidad

La confiabilidad permite evaluar la relación entre el nivel de funcionalidad y la cantidad de recursos usados. La función a continuación muestra el nivel de confiabilidad del sistema.

$$F(t) = (\text{Funcionalidad}) * e^{-\lambda t}$$

Se observa el trabajo hasta que se produzca un fallo en un instante t, se halla probabilidad de falla con una variable aleatoria continua T, en una función exponencial. La relación es la siguiente:

$$\text{Probabilidad de hallar una falla: } P(T \leq t) = F(t)$$

$$\text{Probabilidad de no hallar una falla: } P(T \leq t) = 1 - F(t)$$

Funcionalidad = 90%

$\lambda = 0.01$ (es decir 1 error en cada 10 ejecuciones)

t = 12 meses

Se realiza el cálculo para la confiabilidad del sistema.

$$F(t) = 0.90 * (e^{(-\frac{1}{6} * 12)})$$

$$F(12) = 0.122$$

$$F(12) = 1 - 0.122$$

$$F(12) = 0.88$$

CONFIABILIDAD = 88%

Con este resultado podemos concluir que la probabilidad que el sistema no presente fallas durante los próximos 12 meses, es de un 88%.

4.1.3. Usabilidad

La medición de usabilidad de uso se puede entender como la facilidad que el usuario tiene que interactuar con el sistema, por lo tanto, se debe realizar una serie de preguntas, en la tabla 4.22. ver el nivel de usabilidad del sistema.

Tabla 4.22.
Ponderación de Usabilidad

Nº	Preguntas	Respuestas		Ponderación %
		SI	NO	
1	¿Puede utilizar con facilidad el sistema?	5	0	100%
2	¿El sistema cuenta con interface agradable a la vista?	5	0	100%
3	¿Las repuestas del sistema son satisfactorias?	4	1	80%
4	¿El sistema es comprensible?	5	0	100%
5	¿Le parece complicada las funciones del sistema?	4	1	80%
USABILIDAD				92%

Fuente: (Elaboración Propia)

De acuerdo a los datos obtenidos en la tabla de usabilidad, se puede concluir que el sistema tiene una usabilidad de 92%.

4.1.4. Mantenibilidad

El mantenimiento se desarrolla para realizar modificaciones al software, ya sea por corrección de errores o por el incremento de funcionalidad.

Para hallar la mantenibilidad del sistema se utiliza el índice de madurez de software (IMS), que proporciona una indicación de la estabilidad de un producto de software, se lo determina con la siguiente formula.

$$IMS = \frac{Mt - (Fc + Fa + FE)}{Mt}$$

Mt = Numero de modulo total de la versión actual

Fc = Numero de módulos de la versión actual que se cambiaron

Fa = Numero de módulos de la versión actual que se añadieron

FE = Numero de módulos de la versión anterior que se eliminaron en la versión actual

$$IMS = \frac{7 - (1 + 0 + 0)}{7} = 0.86$$

El resultado establece un 86%, lo que indica que no requiere de mantenimiento inmediatamente.

4.1.5. Portabilidad

En este caso, se refiere a la habilidad del software de ser transferido de un ambiente a otro, y considera los siguientes aspectos:

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

El sistema fue desarrollado en Php usando framework Codeigniter y la base de datos MySQL, se ejecuta en todos los servidores web. También se comprobó que en los distintos navegadores más usados en nuestra área por lo tanto se le da una ponderación de 93% de portabilidad.

Tabla 4.23.
Calidad Global

CARACTERÍSTICAS DE CALIDAD	PORCENTAJE OBTENIDO
FUNCIONALIDAD	90%
CONFIABILIDAD	88%
USABILIDAD	92%
MANTENIBILIDAD	86%
PORTABILIDAD	93%
TOTAL	90%

FUENTE: (Elaboración Propia)

La calidad global se obtuvo un total de 90%, esto significa que la calidad del sistema es considerada aceptable.

4.2. MÉTODO DE ESTIMACIÓN DE COSTO DE SOFTWARE

Existen distintos métodos para la estimación de costo de desarrollo de software.

4.2.1. Método de estimación de COCOMO

El Modelo Constructivo de Costes (COCOMO) es un modelo matemático de base empírica, utilizado para la estimación de costes de software.

Incluye tres submodelos, cada uno ofrece un nivel de detalle y aproximación, cada vez mayor, a medida que avanza el proceso de desarrollo del software: básico, intermedio y detallado.

COCOMO consta con tres modelos de estimación, los mismos se representan en 3 ecuaciones:

$$E = a(KLDC)^b, \text{ persona} - \text{mes}$$

$$D = c(E)^d, \text{ meses}$$

$$P = \frac{E}{D}, \text{ personas}$$

Donde:

E: Esfuerzo requerido por el proyecto expresado en persona-mes.

D: Tiempo requerido por el proyecto expresado en meses.

P: Número de personas requeridas para el proyecto.

a, b, c y d: Constantes con valores definidos según cada sub-modelo.

KLDC: Cantidad de líneas de código, en miles.

A la vez cada modelo se subdivide en modos, los mismos son:

- **Modo orgánico:** Es un pequeño grupo de programadores experimentados desarrollando proyectos de software en un entorno familiar. proyectos relativamente sencillos, menores de 50 KDLC líneas de código, en los cuales se tiene experiencia de proyectos similares y se encuentran en entornos estables
- **Modo semi – libre o semi acoplado:** Proyectos intermedios en complejidad y tamaño (menores de 300 KDLC), donde la experiencia en este tipo de proyectos es variable, y las restricciones intermedias.
- **Modo rígido o empotrado:** Proyectos bastante complejos, en los que apenas se tiene experiencia y se engloban en un entorno de gran innovación técnica. Además, se trabaja con unos requisitos muy restrictivos y de gran volatilidad.

La tabla 4.24. muestra los coeficientes del proyecto de software de acuerdo a los 3 modos expuestos anteriormente.

Tabla 4.24.
Coeficiente COCOMO

Proyecto de Software	a	b	c	d
Orgánico	2.4	1.05	2.5	0.38
Semi-Acoplado	3.0	1.12	2.5	0.35
Empotrado	3.6	1.20	2.5	0.32

Fuente: (Ingeniería de Software, 2011)

4.2.1.1. Costo y estimación de esfuerzo del desarrollo del proyecto

El proyecto se ha implementado con una cantidad **7132** Líneas de Código en lenguaje PHP.

$$LCD = 7132 \text{ [Lineas de Código]}$$

Convertir LCD a KLCD (Miles de Líneas de Código)

La fórmula para el cálculo de KLCD (Miles de Líneas de Código) está dado por:

$$KLDC = LCD/1000$$

$$KLDC = 7132/1000$$

$$KLDC = 7.132 \text{ [Miles de Lineas de Código]}$$

4.2.1.2. Estimación de esfuerzo del proyecto

a) Esfuerzo Nominal

A continuación, haremos el cálculo del esfuerzo necesario para la programación del sistema para ello utilizamos la siguiente ecuación:

$$E = a(KLDC)^b$$

Para hallar el esfuerzo “E” definidos antes el tipo del proyecto que en nuestro caso es orgánico y utilizamos de los datos de la tabla 4.24. Reemplazamos en la fórmula:

$$E = a(KLDC)^b$$

$$E = 2.4(7.132)^{1.05}$$

$$E = 18.88[\textit{Persona /Mes}]$$

b) Esfuerzo del tiempo del proyecto

Ahora para hallar del proyecto usamos los datos de la tabla 4.24. recordando que el proyecto es de tipo orgánico y reemplazando en la siguiente formula.

$$D = c(E)^d$$

$$D = 2.5 (18.88)^{0.38}$$

$$D = 7.64 \approx 8[\textit{Meses}]$$

c) Esfuerzo de personal del proyecto

Para calcular las cantidades en número de programadores se utiliza la siguiente formula, reemplazando los datos ya encontrados:

$$P = \frac{E}{D}$$

$$P = \frac{18.88}{7.64}$$

$$P = 1.69 \approx 2 [\textit{Programadores}]$$

d) Productividad del proyecto

$$PR = LDC/E$$

$$PR = \frac{7132}{18.88}$$

$$PR = 377.75 \left(\frac{LDC}{\text{persona} - \text{mes}} \right)$$

4.2.1.3. Costo de desarrollo

Finalmente, el costo del desarrollo del proyecto está dado por la siguiente fórmula:

$$\text{Costo del Desarrollo} = \text{Nro programadores} * \text{Tiempo}_{\text{programación}} * \text{Salario}_{\text{estimado}}$$

Teniendo en cuenta el salario promedio de un programador = 400\$us.

$$\text{Costo del Desarrollo} = 2 * 8 * 400$$

$$\text{Costo Total del Proyecto} = 6400[\$]$$

Haciendo una conversión en bolivianos el costo de desarrollo será: **44800 bolivianos.**

4.3. SEGURIDAD DE LA INFORMACIÓN DEL SISTEMA

La norma ISO 27001 es una solución de mejora continua en base a la cual puede desarrollarse un Sistema de Gestión de Seguridad de la Información (SGSI) que permita evaluar todo tipo de riesgos o amenazas susceptibles de poner en peligro la información de una organización tanto propia como datos de terceros, para lo cual se tomara los siguientes controles de seguridad.

4.3.1. Seguridad de base de datos

La información en la institución es muy valiosa, es por ello que a nivel de base de datos se tomó en cuenta la validación de los campos a ingresar, con el propósito de evitar cualquier ataque, como por ejemplo el SQL Injection.

Al mismo tiempo, el framework aplicado, establece una conexión segura con la base de datos utilizada, que es MySQL.

4.3.2. Seguridad de autenticación

El sistema tiene el control de sesión o verificación de la autenticación de un nombre de usuario y contraseña, también cabe resaltar que la contraseña de cada usuario este encriptado con el algoritmo sha512.

Así mismo el sistema cuenta con un módulo de privilegios, para la asignación de roles o permisos a los usuarios, es por ello que se controla también un nivel de acceso a la información.

4.3.3. Seguridad de la aplicación

La manipulación de la información estará habilitada solo para los usuarios a los que se ha concedido la autorización.

Se realiza el registro del usuario que modifica la información la base de datos, para esto se registra en cada tabla el identificar del usuario que modifica la información.

A thick, dark blue vertical bar is positioned on the left side of the page. From the bottom of this bar, several thin, curved lines in shades of blue and grey extend upwards and outwards, creating an abstract, organic shape.

CAPITULO V PRUEBAS Y RESULTADOS

5.1. INTRODUCCIÓN

En este capítulo se comentan las pruebas hechas a Sistema de Información para el Control de Pagos por el Suministro de Agua Potable, que validan a este proyecto como resultado al problema definido en el capítulo 1, y finalmente se comentan los resultados.

Las formas de verificar las diversas funcionalidades se pueden separarse en dos grupos:

- a) Aquellas que reciben un conjunto de entradas más o menos simultánea y a partir generan un resultado y
- b) Las que requieren una serie de interacciones con el usuario, en cada una de las cuales ente introduce una serie de datos. (Fernandez Peña, 2006)

5.2. REGISTRO DE CLIENTES

Tabla 5.25.

Prueba y Resultados Registro de Clientes

SITUACIÓN ANTERIOR	SITUACIÓN ACTUAL
El registro de nuevas suscripciones o clientes se realiza de manera manual, la información de clientes se guardan en los archivos físicos, tarea que efectúa tiene una duración de aproximadamente por cliente de 10 a 15 minutos.	Con el sistema, el registro de clientes lo realiza de manera más rápida en un tiempo de 3 a 5 minutos, de manera que los datos son guardados en el servidor con la finalidad de mantener la información para cuando un cliente lo solicite.

FUENTE: (Elaboración Propia)

Resultados. - Reduce un 33% de tiempo de registro de clientes, lo cual el sistema ayuda de gran manera reduciendo el tiempo de atención, además evitando datos que hayan sido ingresados erróneamente.

5.3. REGISTRO DE LECTURAS

Tabla 5.26.

Prueba y Resultados Registro de Lecturas

SITUACIÓN ANTERIOR	SITUACIÓN ACTUAL
Las lecturas de medidores se realizan cada mes, pero esta actividad de registro lo realiza manualmente anotando en cuadernos, en un tiempo de 3 a 5 minutos.	Actualmente con la implementación del sistema, el registro de lecturas se procede, con solo ingresando el número de medidor, tendrá los datos del cliente y así evitar errores al momento de tomar los datos del medidor, la cual este proceso tiene una duración de 1 minuto.

FUENTE: (Elaboración Propia)

Resultados. - El sistema tiene una gran ventaja reduciendo en un 65% de tiempo para el registro de lecturas.

5.4. GENERACIÓN DE PLANILLAS DE CONSUMO DE AGUA

Tabla 5.27.

Prueba y Resultados Generación de Planillas

SITUACIÓN ANTERIOR	SITUACIÓN ACTUAL
Para realizar el proceso de planillas, una vez que se haya hecho la respectiva lecturación, se procede a generar las planillas acorde a consumo de agua, usando medios como calculadora para dar costos de consumo, la cual se realiza entre 1 a 2 días.	Con el sistema implementado las planillas de consumo se generan de forma automática al momento de ingresar la lectura, sin necesidad de usar otros medios para el costo tarifario.

FUENTE: (Elaboración Propia)

Resultados. - Reduce un 100% de tiempo de trabajo.

5.5. INFORME ANUAL DE PAGOS

Tabla 5.28

Prueba y Resultados Informe Pagos

SITUACIÓN ANTERIOR	SITUACIÓN ACTUAL
Para realizar un informe sobre los pagos y deudas pendientes, los comité de agua potable se reúnen una semana a dos semanas antes para preparar el informe detallado, y así rendir cuentas a la asamblea general de usuarios.	El sistema implementado de manera automática genera el informe detallado, por meses como también anual. Además el sistema permite ver en modo grafico estadístico el consumo de agua.

FUENTE: (Elaboración Propia)

Resultados. - Reduce un 100% de tiempo de trabajo para los de comité de agua potable.

A decorative graphic on the left side of the page consists of a solid dark blue vertical bar. To its right, several thin, curved lines in shades of blue and grey extend from the bottom of the bar towards the right, creating an abstract, organic shape.

CAPITULO VI

CONCLUSIONES Y RECOMENDACIONES

6.1. CONCLUSIONES

Al finalizar del presente proyecto de grado Sistema de Información de Control de Pagos por el Suministro de Agua Potable Caso: Comité de Agua Potable Laka Pucara, se concluye lo siguiente.

- El desarrollo del presente proyecto fue posible gracias a la información facilitada por los miembros de Comité de Agua Potable Laka Pucara.
- El sistema desarrollado gestiona los procesos de suscripción, lecturación, cobranza de tarifas y generación de reportes de manera automática, que permiten un mejor proceso de recaudación por el consumo de agua potable, logrando cumplir adecuadamente con los requisitos especificados por los usuarios finales.
- Con la implementación del sistema web, se logró centralizar y almacenar toda la información referente al registro, control de pagos y generando reportes inmediatos.
- Con el sistema desarrollado se logra permitir al administrador, tesorera y operador la facilidad en la administración en gran medida y el óptimo desempeño de los mismos.
- Las herramientas utilizadas para el diseño y desarrollo son muy confiables y potentes puesto que están siendo actualizadas todo el tiempo por profesionales con conocimientos de tecnologías de información bastantes avanzadas.

6.2. RECOMENDACIONES

- Se recomienda establecer políticas de seguridad, respaldos de información almacenado en la base de datos para garantizar que la información este seguro y podrá ser recuperada inmediatamente en caso que surja algún imprevisto dentro del servidor.
- Es recomendable realizar cambios de contraseñas cada 3 meses, esto para evitar riesgos de ingreso al sistema por personas foráneas.
- Se recomienda implementar un módulo que permita la administración del inventario en almacenes de los materiales, herramientas y equipo correspondientes para servicio de agua potable.
- Se recomienda también a los de comité de agua potable a utilizar y administrar el sistema de acuerdo a las instrucciones brindadas.

BIBLIOGRAFÍA

1. Ludwing-Maximilians. (10 de Agosto de 2016). *UWE - Ingenieria Web basada en UML*. Obtenido de Modelo de Requisitos: <http://uwe.pst.ifi.lmu.de/teachingTutorialRequirementsSpanish.html>
2. *Administración de Proyectos de Software*. (22 de Noviembre de 2015). Obtenido de COCOMO II: <http://adminprojsoft.blogspot.com/2015/11/cocomo-ii.html>
3. Alegsa, L. (27 de Agosto de 2018). *DICCIONARIO DE INFORMÁTICA Y TECNOLOGIA*. Obtenido de Definicion de sistema: <http://www.alegsa.com.ar/Dic/sistema.php>
4. Aramayo Canazas, Y. P. (2009). *Sistema de facturacion de consumo de agua potable para la Provincia Inquisivi*. La Paz - Bolivia.
5. Aranibar, N., Tinoco, J. A., & Ibarra, M. (2011). *Monografía.com*. Obtenido de MySQL WorkBench.
6. Arenols Solano, A. (Enero01 de 2019). *OpenWebinars*. Obtenido de Qué es PHP: Características y usos: <https://openwebinars.net/blog/que-es-php/>
7. Barrera, A. (s.f.). *next_u*. Obtenido de CONOCE LAS VENTAJAS Y DESVENTAJAS DE JAVASCRIPT: <https://www.nextu.com/blog/conoce-las-ventajas-y-desventajas-de-javascript/>
8. Borbón, N. I. (12 de Marzo de 2013). *EVALUACION DE SOFTWARE*. Obtenido de NORMA DE EVALUACION ISO/IEC 9126: <http://actividadreconocimiento-301569-8.blogspot.com/2013/03/norma-de-evaluacion-isoiec-9126.html>
9. Carrodegua, N. (2019). *norfipc.com*. Obtenido de Como instalar y configurar el servidor web Apache en Windows: <https://norfipc.com/internet/instalar-servidor-apache.html>
10. Cedejas Valdez, J. (2014). *Enciclopedia Virtual*. Obtenido de Modelos y metodologías para el desarrollo de software: <https://www.eumed.net/tesis-doctorales/2014/jlcv/software.htm#:~:text=Como%20se%20explic%C3%B3%20en%20el,informaci%C3%B3n%20parcial%20sobre%20el%20proceso.>
11. De Luca, D. (2011). *HTML5*. Buenos Aires: Users.
12. De Luca, D. (2012). *Desarrollo Web Capacitación y Consultoría*. Obtenido de ¿Qué es CSS3?: <https://damiandeluca.com.ar/que-es-css3>
13. Del Valle Roque, D. (25 de Noviembre de 2014). *Estimación de costos de desarrollo de software*. Obtenido de <https://www.gestiopolis.com/estimacion-de-costos-de-desarrollo-de-software/>
14. *Desarrolloweb.com*. (s.f.). Obtenido de Javascript a fondo: <https://desarrolloweb.com/javascript/>

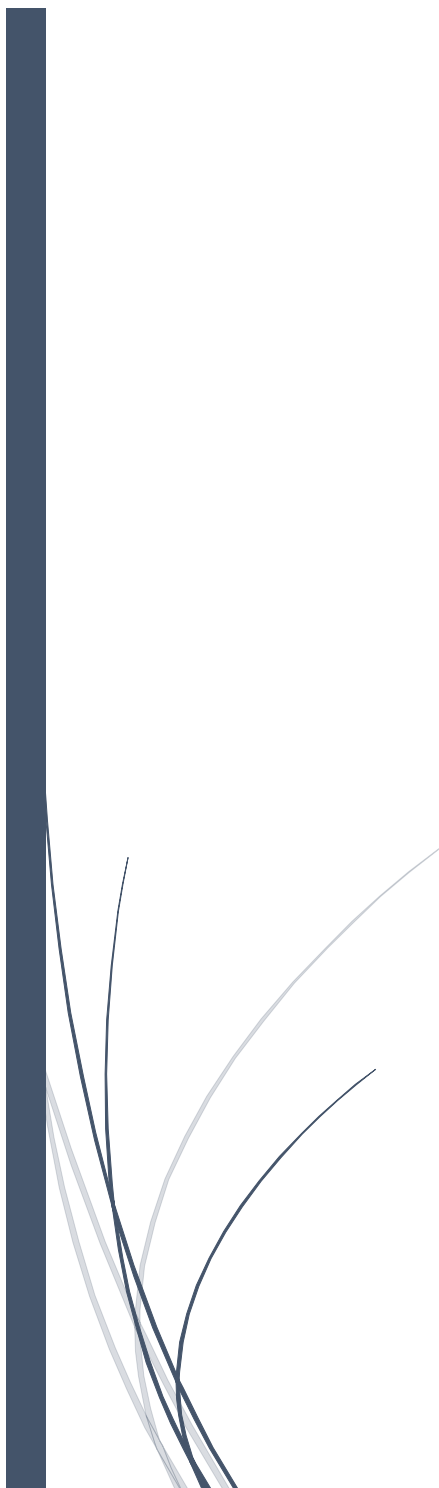
15. *Digital Guide IONOS*. (16 de Marzo de 2020). Obtenido de CodeIgniter, el peso pluma de los frameworks PHP: ionos.es/digitalguide/paginas-web/desarrollo-web/codeigniter-framework-php-rapido-y-versatil/
16. Dip, P. (1 de Mayo de 2011). *Tecnología e Informática*. Obtenido de Sistemas: <http://latecnologiavirtual.blogspot.com/2011/05/sistemas.html>
17. Dipp, C. (2010). *Sistema de Control de Cobros de Agua Potable para comites afiliados utilizando Framework CAKEPHP*. Cochabamba.
18. *Economíasimple.net*. (2016). Obtenido de Definición de Suministros.
19. *ECURED*. (28 de Julio de 2019). Obtenido de COCOMO II: https://www.ecured.cu/COCOMO_II
20. *El Instituto de Nutrición de Centro América y Panamá*. (s.f.). Obtenido de Sistema de Información: <http://www.incap.int/sisvan/index.php/es/acerca-de-san/conceptos/797-sin-categoria/501-sistema-de-informacion>
21. Fernandez Peña, J. (29 de 11 de 2006). *Pruebas de sistema*. Obtenido de <https://www.uv.mx/personal/jfernandez/files/2010/07/Pruebas-de-Sistema.pdf>
22. Figueroa, W. E. (2018). *SISTEMA DE FACTURACIÓN UTILIZANDO LOS FRAMEWORK ANGULAR 5, LARAVEL 5.5 PARA LA JUNTA ADMINISTRADORA DE AGUA POTABLE SAN JOSÉ DE CHORLAVÍ DE LA CIUDAD DE IBARRA – ECUADOR*. Ibarra - Ecuador.
23. Fontan, M. (25 de octubre de 2012). *ASOCIACIÓN DESARROLLADORES WEB DE ESPAÑA*. Obtenido de CodeIgniter, un framework PHP para el desarrollo rápido de aplicaciones web.: <http://www.adwe.es/codigo/codeigniter-framework-php-desarrollo-aplicaciones-web>
24. Fumas Cases, E. (11 de Junio de 2014). *ibrugor*. Obtenido de Apache HTTP Server: ¿Qué es, cómo funciona y para qué sirve?: <https://www.ibrugor.com/blog/apache-http-server-que-es-como-funciona-y-para-que-sirve/>
25. Galeano, L. (3 de Noviembre de 2012). *Planificación de mi proyecto II (Luis Galeano)*. Obtenido de METODOLOGIA UWE APLICADA A MI SOLUCION INFORMATICA DE MI PROYECTO: <http://elproyectodeluisgaliano.blogspot.com/2012/11/metodologia-uwe-aplicada-mi-solucion.html>
26. Gomez, A., López, M., Migani, S., & Otazú, A. (Noviembre de 2010). *COCOMO II Modelo para la Estimación de Costo*. Obtenido de COCOMO: <https://blogadmi1.files.wordpress.com/2010/11/cocomoIIfull.pdf>
27. Gonzales Gutierrez, E. (s.f.). *aprenderaprogramar.com*. Obtenido de Que es PHP: https://www.aprenderaprogramar.com/index.php?option=com_content&view=article

&id=492:ique-es-php-y-ipara-que-sirve-un-potente-lenguaje-de-programacion-para-crear-paginas-web-cu00803b&catid=70&Itemid=193

28. Gonzales Reyes, D., Quijije Lucas, G., Betty Carvajal, N., Manrique Neira, J., & Quijije Toro, K. (2017). *monografias.com*. Obtenido de Telecomunicaciones: Arquitectura cliente/servidor: <https://www.monografias.com/docs114/telecomunicaciones-arquitectura-cliente-servidor/telecomunicaciones-arquitectura-cliente-servidor.shtml>
29. Gonzales, D. E., & Velastegui, D. S. (2008). *Estudio de protocolos entre PDA/PC y su aplicacion en el sistema de facturacion de las EAPA San Mateo*. Chimborazo - Ecuador.
30. *HACK A BOSS*. (19 de Septiembre de 2019). Obtenido de LA HISTORIA DE JAVASCRIPT: <https://hackaboss.com/blog/historia-javascript/>
31. *Ingenieria de Software*. (2011). Obtenido de Modelos empíricos de estimación: <https://ingenieriasoft.webcindario.com/gestion-y-planificacion-de-proyectos/planificacion-de-proyectos-de-software/modelos-empiricos-de-estimacion.html>
32. *Ingenieria Web*. (2012). Obtenido de UWE - UML- based WEB Engineering: <https://sites.google.com/site/ingenieriawebuc/home/contenido/uwe>
33. *ISO 27001: Sistemas de gestión de seguridad de la información*. (s.f.). Obtenido de ORGANISMO DE CERTIFICACIÓN GLOBAL: <https://www.nqa.com/es-mx/certification/standards/iso-27001>
34. Lopez, J. (8 de diciembre de 2010). *INGENIERIA DE SOFTWARE*. Obtenido de INGENIERIA WEB: upolijenny.blogspot.com/2010/12/ingenieria-web.html
35. Medina Prieto, R. (24 de julio de 2017). *MODELOS Y ESTÁNDARES DE CALIDAD APLICADOS AL SISTEMA DE INFORMACIÓN*. Obtenido de LA NORMA ISO/IEC 9126: <http://unidad4rociomp.blogspot.com/2017/07/46.html>
36. *Metodologia UWE*. (25 de Junio de 2015). Obtenido de Metodologia UWE: <https://metodologiauwe.wordpress.com>
37. *Mi carrera universitaria*. (s.f.). Obtenido de Ingenieria de Software: Qué es, objetivos, características y más: <https://micarrerauniversitaria.com/c-ingenieria/ingenieria-de-software/>
38. *Microsoft*. (17 de Octubre de 2017). Obtenido de Configurar una red inalámbrica: <https://support.microsoft.com/es-es/help/17137/windows-setting-up-wireless-network>
39. Morales, A. (09 de Diciembre de 2019). *TodaMateria*. Obtenido de Información: <https://www.todamateria.com/informacion/>
40. *MySQL*. (s.f.). Obtenido de MySQL Workbench: <https://www.mysql.com/products/workbench/>

41. *Neo Attack*. (s.f.). Obtenido de ¿Qué es Apache?:
<https://neoattack.com/neowiki/apache/>
42. Pérez Díaz, A. (s.f.). *AJPD soft*. Obtenido de Apache (Servidor HTTP):
<http://www.ajpdsoft.com/modules.php?name=Encyclopedia&op=content&tid=820>
43. Perez, D. (26 de Octubre de 2007). *MAESTROS DEL WEB*. Obtenido de Qué son las bases de datos: <http://www.maestrosdelweb.com/que-son-las-bases-de-datos/>
44. Pressman , R. S. (2010). *Ingenieria de Software*. Mexico: Mexicana.
45. Raffino, M. (30 de Julio de 2019). *Concepto.de*. Obtenido de RED DE COMPUTADORAS: <https://concepto.de/red-de-computadoras/>
46. Raffino, M. (4 de Julio de 2020). *Concepto.de*. Obtenido de REDES INFORMÁTICAS: concepto.de/redes-informaticas/
47. Raffino, M. E. (14 de Febrero de 2020). *Concepto de* . Obtenido de Concepto de dato en Informatica: <https://concepto.de/dato-en-informatica/>
48. Ramirez, I. (11 de Septiembre de 2019). *EFEECTO DIGITAL*. Obtenido de Los Sistemas de Información: <https://www.efectodigital.online/post/2018/03/02/los-sistemas-de-informaci%C3%B3n>
49. *Reviso*. (2020). Obtenido de ¿Qué es una factura?: <https://www.reviso.com/es/que-es-una-factura>
50. Robledano, Á. (24 de Septiembre de 2019). *OpenWebinars*. Obtenido de Qué es MySQL: Características y ventajas: <https://openwebinars.net/blog/que-es-mysql/>
51. Ruiz Andrade, A. B., & Rosales Terán, A. (Noviembre de 2016). *DESARROLLO E IMPLEMENTACIÓN DE APLICATIVO WEB PARA RESERVAS DE PARQUEADEROS*. Quito. Obtenido de repositorio.puce.edu.ec/bitstream/handle/22000/13122/Tesis
52. Ruiz, A. (20 de Septiembre de 2014). *Blog del Máster en Comunicación y Marketing Digital de la UAB*. Obtenido de ¿Qué es Bootstrap?:
<https://www.mastermarketingdigital.com/everriculum/2014/09/20/que-es-bootstrap/>
53. Schiaffarino, A. (12 de Marzo de 2019). *infranetworking*. Obtenido de Modelo cliente servidor: <https://infimg.com/bimg/2019/02/diagrama-cliente-servidor.jpeg>
54. Solis, J. (26 de Septiembre de 2014). *ARWEB.com*. Obtenido de ¿QUÉ ES BOOTSTRAP Y CÓMO FUNCIONA EN EL DISEÑO WEB?:
<https://www.arweb.com/chucherias/que-es-bootstrap-y-como-funciona-en-el-diseno-web/>
55. Sommerville, I. (2005). *Ingeniería del software*. Madrid: PEARSON EDUCACION S.A.

56. Unir Revista. (11 de Noviembre de 2019). *¿Qué es la certificación ISO 27001 y para qué sirve?* Obtenido de Ingeniería y Tecnología:
<https://www.unir.net/ingenieria/revista/noticias/iso-27001/549204720236/#:~:text=La%20ISO%2027001%20es%20una,y%20aplicaciones%20que%20la%20tratan.>
57. *Vendus*. (s.f.). Obtenido de Facturación: definición y características:
<https://www.vendus.es/blog/facturacion-definicion/>
58. *WIKILIBROS*. (2 de Diciembre de 2017). Obtenido de Programacion en PHP:
https://es.wikibooks.org/wiki/Programación_en_PHP
59. *Wikipedia*. (19 de mayo de 2019). Obtenido de MySQL:
<https://es.wikipedia.org/wiki/MySQL>
60. *Zona Económica*. (17 de Abril de 2019). Obtenido de Concepto de Control:
<https://www.zonaeconomica.com/control>



ANEXOS

ANEXO A

ARBOL DE PROBLEMAS



Fuente: (Elaboración Propia)

ANEXO B

ARBOL DE OBJETIVOS



Fuente: (Elaboración Propia)

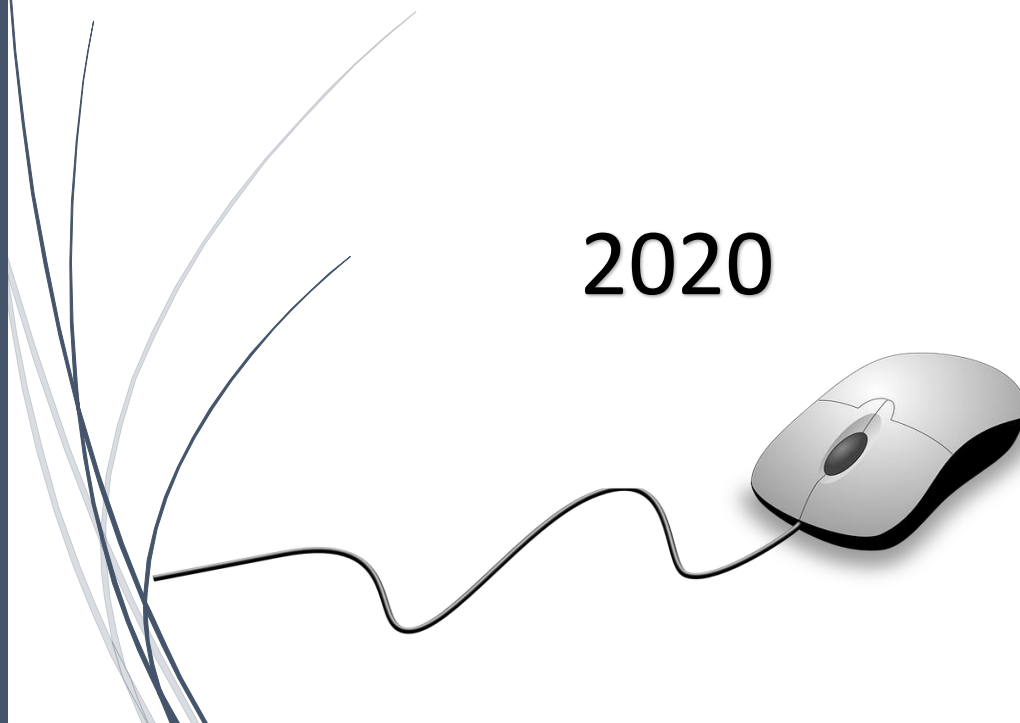
ANEXO C

MANUAL DE USUARIO

**SISTEMA DE INFORMACION PARA EL
CONTROL DE PAGOS POR EL SUMINISTRO
DE AGUA POTABLE**

DESARROLLADOR: MARCOS COLQUE TICONA

2020



Comité de Agua Potable y Saneamiento Laka Pucara

PROV. PACAJES – MUNICIPIO COROCORO

ÍNDICE

1. INTRODUCCIÓN.....	1
2. OBJETIVO DEL SISTEMA.....	1
3. REQUERIMIENTOS DEL SISTEMA.....	1
4. TIPOS DE USUARIOS.....	1
5. IMPLEMENTACIÓN DE SISTEMA.....	1
5.1. Interfaz de Inicio de Sistema	2
5.2. Pantalla Inicio Principal	2
5.3. Pantalla de usuarios (Administrador).....	3
5.4. Interfaz de registro de nuevo usuario (Administrador).....	3
5.5. Administración de Permisos (Administrador).....	5
5.6. Agregar Nuevo Módulo (Administrador).....	5
5.7. Administración de Clientes (Administrador)	6
5.8. Registro de Nuevo Cliente (Administrador).....	7
5.9. Administración de Tarifas (Administrador)	8
5.10. Administración de Multas (Administrador).....	9
5.11. Panel de Lecturas (Operador).....	10
5.12. Registro de lecturas de medidores (Operador)	10
5.13. Detalle de Consumo (Administrador).....	11
5.14. Cobranza (Tesorera).....	13
5.15. Consulta de Multas (Administrador)	14
5.16. Reportes (Administrador).....	15
5.17. BACKUPS (Administrador)	16

1. INTRODUCCIÓN

El presente documento tiene la finalidad de guía para el uso adecuado de los diferentes módulos que contempla el “SISTEMA DE INFORMACIÓN PARA EL CONTROL DE PAGOS POR EL SUMINISTRO DE AGUA POTABLE, el sistema está orientado en entorno web que nos permitirá visualizar en un navegador, de la misma se recomienda que durante la lectura de este documento vaya practicando y constatando con el aplicativo en sí.

2. OBJETIVO DEL SISTEMA

El objetivo es guiar al usuario a través de capturas de imágenes del sistema, para el uso correcto del sistema.

3. REQUERIMIENTOS DEL SISTEMA

- a) Requerimiento de hardware.
 - Computadora personal o Smartphone
 - Conexión a internet.

- b) Requerimiento de software.
 - Sistema operativo Windows o Linux y un navegador.

4. TIPOS DE USUARIOS

El sistema contiene 3 tipos de usuario.

1. Usuario Administrador
2. Usuario Tesorera
3. Usuario Operador

5. IMPLEMENTACIÓN DE SISTEMA

Dentro de su navegador, escriba la siguiente dirección: <https://www.siapla.com>

5.1. Interfaz de Inicio de Sistema

En esta autenticación, el usuario debe ingresar su Usuario y su Contraseña. Primero tiene que estar registrado sus datos en el sistema luego recién el usuario podrá acceder al sistema con sus respectivos privilegios asignados a cada usuario.



5.2. Pantalla Inicio Principal

Una vez que ingresamos con usuario y contraseña al sistema nos mostrara de la siguiente manera el Inicio principal, a continuación, vemos que nos muestra las partes del sistema.



5.3. Pantalla de usuarios (Administrador)

Nos vamos al menú usuarios, realizamos clic en la pestaña Usuarios, luego se desplegará 2 submenús una de esas esta con el mismo nombre usuarios ahí se debe hacer el clic y nos mostrará la lista de usuarios activos e inactivos, con sus respectivos botones editar, eliminar, resetear contraseña, cambiar estado y obviamente en la parte superior está el botón Agregar Nuevo Usuario.

The screenshot shows the 'USUARIOS' management page in the SIAPLA system. The interface includes a search bar at the top right, a table of users, and a sidebar menu. Red callouts with numbers 1 through 3 point to specific elements: '1 click' points to the 'USUARIOS' menu item; '2 Click' points to the 'USUARIOS' sub-menu item; '3 Click añadir nuevo usuario' points to the '+AGREGAR USUARIOS' button. Other callouts include 'Buscar' pointing to the search bar, 'Editar' pointing to the edit button, 'Reset contraseña' pointing to the reset password button, 'cambiar estado' pointing to the status toggle buttons, and 'Eliminar' pointing to the delete button. The table contains the following data:

Nº	Cedula Identidad	Nombre Completo	Telefono	Correo	Tipo Usuario	Imagen	Opciones
1	8436244 LP	Marcos Colque Ticona	69818600	marco.collic@gmail.com	Administrador		ACTIVO
2	10920289 LP	Jhessica Colque Ticona	74865036	jhessica2003@gmail.com	Administrador		INACTIVO
3	12345678 LP	usuario1 hola hola	1232	usuario@gmail.com	Administrador		ACTIVO
4	13734041 LP	Ever Colque Ticona	74865036	eversto@gmail.com	Operador		ACTIVO

5.4. Interfaz de registro de nuevo usuario (Administrador)

1.- Primero nos vamos al botón, **AGREGAR USUARIOS**, hacemos el clic e inmediato te mostrará el formulario de registro, donde pedirá que ingreses cédula de identidad solo número, una vez ingresado el cedula identidad automáticamente te aparecerá los campos para ingresar sus datos personales.

2.- El tipo de usuario podrá se asignado de acuerdo al cargo que tiene la persona a registrar.

3.- En la casilla te pedirá un nombre usuario que la misma no puede ser duplicado, el sistema automáticamente le mostrará un mensaje si está ocupado o no, el nombre usuario ingresado.

4.- para completar su registro deberá introducir la contraseña, que esta le servirá para el acceso al sistema.

5.- finalmente para guardar los datos, se deberá hacer clic en el botón **GUARDAR**.

The image shows a web application window titled "NUEVO USUARIO" (New User). The form contains the following fields and elements:

- Message Box:** "Para registrar los Usuarios es necesario que la persona este registrado, ingrese el codigo o Cedula de Identidad." (To register users it is necessary that the person is registered, enter the code or ID card.)
- CARNET:** Input field with value "1234567890" (marked with a red circle 1).
- EXPEDIDO:** Dropdown menu with value "LI".
- NOMBRE:** Input field with value "Nombres..".
- PATERNO:** Input field with value "Paterno...".
- MATERNO:** Input field with value "Materno...".
- TELEFONO:** Input field with value "Telefono...".
- CELULAR:** Input field with value "Celular...".
- DIRECCION:** Input field.
- F_NACIMIENTO:** Input field with value "dd/mm/aaaa" and a calendar icon.
- GENERO:** Dropdown menu with value "MASCULINO".
- EMAIL:** Input field with value "Email...".
- TIPO USUARIO:** Dropdown menu with value "Administrador" (marked with a red circle 2).
- USUARIO:** Input field with value "user1" (marked with a red circle 3).
- FOTO:** File upload area with "Seleccionar archivo" and "Ningún ...ionado".
- CONTRASEÑA:** Input field with value ".....".
- CONFIRMAR CONTRASEÑA:** Input field with value ".....".
- Feedback:** "Password Correcto" (marked with a red circle 4).
- Buttons:** "CERRAR" (labeled "Botón Cerrar") and "GUARDAR" (labeled "Botón Guardar").

5.5. Administración de Permisos (Administrador)

Para dar permisos de los módulos se debe hacer clic en submenú **PRIVILEGIOS**, luego nos mostrará la lista de todos los módulos asignados a los usuarios, también se podrá desactivar o activar los módulos o en todo caso eliminar, para asignar nuevos módulos se debe hacer clic en **NUEVO MODULO**.

ADMINISTRACION DE PERMISOS

ADMINISTRACION DE PERMISOS

ADMINISTRACION DE PRIVILEGIOS ADMINISTRACION DE MODULOS DE SISTEMA

+ NUEVO MODULO Agregar Nuevo Módulo

Administrador sistema ("Administrador")

NOMBRE MODULO	LISTAR	INSERTAR	EDITAR	ELIMINAR	C_ESTADO	BUSCAR	ESTADO
admin_usuarios	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	ACTIVO
administracion_privilegios	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	ACTIVO
clientes	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	ACTIVO
admin_gestion	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	ACTIVO
admin_lectura	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	ACTIVO
admin_detalle_consumo	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	ACTIVO
admin_multas	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	ACTIVO
admin_detalle_multas	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	ACTIVO
admin_reportes	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	ACTIVO

5.6. Agregar Nuevo Módulo (Administrador)

Para asignar módulos se debe hacer clic en **NUEVO MODULO** y les mostrará un interfaz modal, donde se debe seleccionar el tipo de usuario, y al seleccionar de manera automática se desplegará una lista de módulos, los módulos que corresponde al usuario se debe marcar y por ultimo clic en el Botón **GUARDAR**.

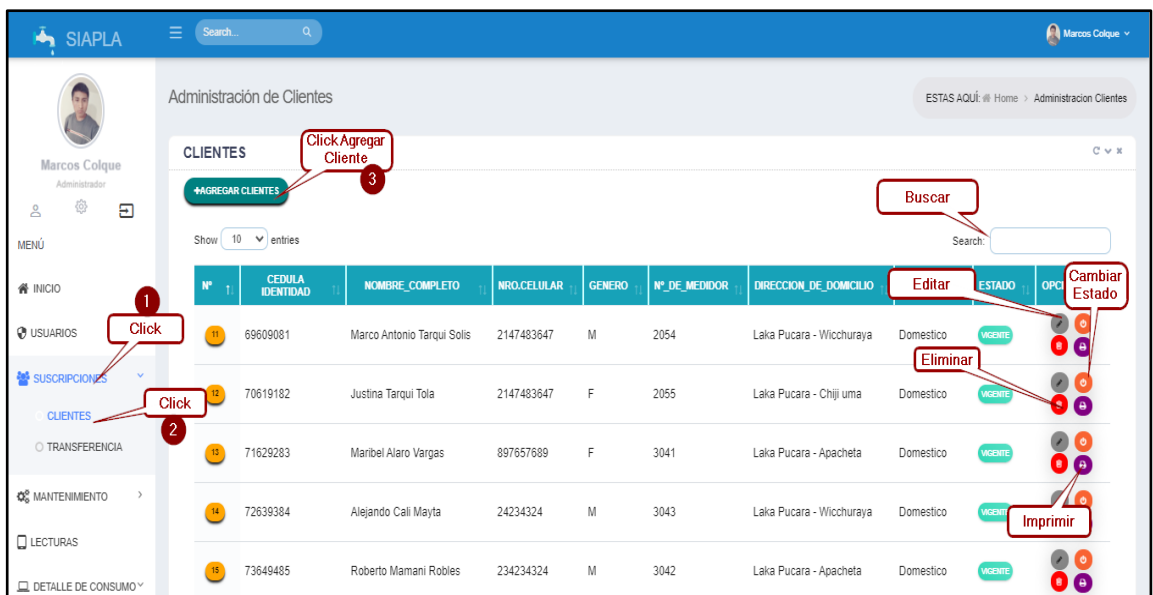


5.7. Administración de Clientes (Administrador)

1.- Hacer clic en el menú **SUSCRIPCIONES**, luego se desplegará los submenús.

2.- Hacer clic en **CLIENTES**, donde le mostrará la lista de clientes con sus respectivas eliminar, editar, Cambiar estado e imprimir, también puede usar el buscador.

3.- En botón **AGREGAR CLIENTES** es donde se puede registrar nuevos clientes.



5.8. Registro de Nuevo Cliente (Administrador)

- 1.- Debe ingresar en número de cedula de identidad del cliente a registrar., luego completar los campos como nombre, apellidos etc.
- 2.- Debe seleccionar la categoría que solicite el cliente.
- 3.- Si el cliente solicita cambio de titularidad, se debe desplegar el tipo de traspaso donde te mostrara otros campos.
- 4.- Se debe registrar la dirección donde se va instalar el medidor.
- 5.- Se debe asignar un numero de medidor.
- 6.- Se debe marcar los documentos presentados por el cliente al momento de registro.
- 7.-Por último, se debe hacer clic en el Botón GUARDAR, para su registro correspondiente.

Para registrar los Clientes es necesario que ingrese el Número de Cedula de Identidad del Cliente. Introduzca Cedula Identidad: 1234567890

CARNET: 1234567890 EXPEDIDO: LP NOMBRE: Nombres...

PATERNO: Paterno... MATERNO: Materno... TELEFONO: Telefono...

CELULAR: Celular... F_NACIMIENTO: dd/mm/aaaa GENERO: MASCULINO

DIRECCION:

CATEGORIA: Domestico INGRESO: Bs. 350 .00 TIPO DE TRASPASO: Ninguno

Nº MEDIDOR: 111213 DIRECCION DONDE SE VA INSTALAR MEDIDOR: Domicilio...

DOCUMENTOS PRESENTADOS ::

Fotocopia de Carnet

Fotocopia de Catastro

Carta de Solicitud

Folder Amarillo

Botón Cerrar CERRAR Botón Guardar GUARDAR

5.9. Administración de Tarifas (Administrador)

1.- Primero se debe hacer clic en el menú **MANTENIMIENTO**, luego se desplegará los submenús.

2.- Hacer clic en el submenú **TARIFAS**, le mostrara la lista de las tarifas ya registrados, también cuenta con las mismas opciones de eliminar, editar, cambiar estado y registrar nueva tarifa.

Registro

TARIFAS

+NUEVA TARIFA Botón Nueva Tarifa

Buscar

Show 10 entries

N°	M3	Costo Básico	Costo Excedente	Categoría	Fecha	Estado	Opciones	Estado
1	10 m3	10.00 Bs.	0.70 Bs.	Comercial	2019-04-16	INACTIVO	🗑️ ⚙️	
2	10 m3	8.00 Bs.	0.50 Bs.	Domestico	2019-07-31	ACTIVO	🗑️ ⚙️	
3	11 m3	10.00 Bs.	0.60 Bs.	Domestico	2019-11-28	INACTIVO	🗑️ ⚙️	Eliminar
4	5 m3	10.00 Bs.	0.20 Bs.	Comercial	2020-07-02	ACTIVO	🗑️ ⚙️	Editar

Showing 1 to 4 of 4 entries

FORMULARIO DE NUEVO COSTO

METRO CUBICO:
Ingrese la cantidad... 1

COSTO BASICO M3:
Bs Ingrese costo... 2

COSTO EXCEDENTE M3:
Bs Ingrese costo excedente 3

CATEGORIA:
Domestico 4

Botón Cerrar CERRAR

Botón Guardar GUARDAR

5.10. Administración de Multas (Administrador)

1.- debe hacer clic en menú MANTENIMIENTO, luego se desplegará el submenú MULTAS.

2.- Hacer clic en submenú MULTAS, le mostrara la lista de multas, con sus respectivas opciones eliminar, editar, cambiar estado y también de agregar nuevo Multa.

The screenshot displays the 'MULTAS' management interface. At the top, there is a search bar and a '+AGREGAR' button. Below is a table with columns for 'Nº', 'Tipo Multa', 'Descripción', 'Costo', 'Estado', and 'Opciones'. The table contains three entries:

Nº	Tipo Multa	Descripción	Costo	Estado	Opciones
1	Reconexión sin autorización	Cuando el usuario a sido suspendido del servicio, y se auto reconecta.	100.00 Bs.	ACTIVO	[Edit] [Delete] [Status]
2	Corte de agua por morosidad	Cuando el usuario a esta en mora por mas de 2 meses.	50.00 Bs.	ACTIVO	[Edit] [Delete] [Status]
3	Conexión clandestina	Persona que realiza conexiones sin previa autorización.	200.00 Bs.	ACTIVO	[Edit] [Delete] [Status]

At the bottom of the table, it says 'Showing 1 to 3 of 3 entries' and includes 'PREVIOUS' and 'NEXT' navigation buttons.

The 'FORMULARIO DE NUEVA MULTA' modal is shown. It contains the following fields and buttons:

- NOMBRE MULTA:** A text input field with the placeholder 'Ingrese el nombre multa...' and a red callout '1'.
- DESCRIPCION:** A text area with a red callout '2'.
- COSTO:** A text input field with a currency selector 'Bs' and the placeholder 'Ingrese costo...', with a red callout '3'.
- Botón Guardar:** A blue button labeled 'GUARDAR' with a red callout 'Botón Guardar'.
- Botón Cerrar:** A red button labeled 'CERRAR' with a red callout 'Botón Cerrar'.

The background shows the 'MULTAS' table from the previous screenshot, partially obscured by the modal.

5.11. Panel de Lecturas (Operador)

1.- El usuario operador debe hacer clic en menú **LECTURAS**, donde se mostrará los registros de toda la lectura con la opción de eliminar y editar, pero si el cliente ya está en estado inactivo no podrá hacer nada.

2.- Para registrar las lecturas debe hacer clic en botón **AGREGAR LECTURAS**, le mostrara una ventana modal, donde le pedirá el número de medidor, y de manera automática desplegara todos los datos del cliente.

3.- También podrá realizar búsquedas con Ej. Nombre de cliente, número de medidor, fecha de lecturación etc.

The screenshot displays the 'Panel de Lecturas' in the SIAPLA system. The main content area is titled 'Registra Lecturas de Medidores' and contains a table of meter readings. The table has the following columns: N°, N° Medidor, Datos de Cliente, Lectura Anterior, Lectura Actual, Consumo M3, Gestion, Mes, Fecha Lectura, Estado, and Opciones. The table lists five entries with their respective details. A sidebar menu on the left includes 'LECTURAS' which is highlighted. A search bar is located at the top right of the table area. Red callouts with numbers 1-3 indicate key interactive elements: 1 points to the 'LECTURAS' menu item, 2 points to the '+AGREGAR LECTURAS' button, and 3 points to the search bar. Other callouts point to 'Editar' and 'Eliminar' buttons in the table's 'Opciones' column.

N°	N° Medidor	Datos de Cliente	Lectura Anterior	Lectura Actual	Consumo M3	Gestion	Mes	Fecha Lectura	Estado	Opciones
11	2061	Juanito Apaza Claros (77772020 LP)	33	47	14	2020	Abril	2020-05-07	ACTIVO	Editar Eliminar
12	2055	Justina Tarqui Tola (70619182 LP)	33	36	3	2020	Abril	2020-05-02	INACTIVO	Editar Eliminar
13	3041	Maribel Alaro Vargas (71629283 LP)	45	66	21	2020	Abril	2020-05-02	INACTIVO	Editar Eliminar
14	2061	Juanito Apaza Claros (77772020 LP)	25	33	8	2020	Marzo	2020-04-06	ACTIVO	Editar Eliminar
15	2055	Justina Tarqui Tola (70619182 LP)	17	33	16	2020	Marzo	2020-04-05	INACTIVO	Editar Eliminar

5.12. Registro de lecturas de medidores (Operador)

1.- El usuario operador para registrar las lecturas debe ingresar el número de medidor, si existe registrado el numero ingresado se desplegará de manera automática los datos del cliente.

2.- Se debe introducir el recorrido del medidor.

3.- Deberá seleccionar el mes que corresponde a la lecturación.

4.- Seleccionar el año.

5.- El botón permite cerrar el formulario o cancelar.

6.- Una vez llenado los datos requeridos en el formulario se deberá guardar haciendo clic en botón **GUARDAR**

5.13. Detalle de Consumo (Administrador)

1.- Hacer clic en el menú **DETALLE CONSUMO**, luego se desplegará los submenús.

2.- Hacer clic en submenú **CONSUMO**, luego mostrara el panel con respectivos botones.

3.- Debe seleccionar el año

4.- Debe seleccionar el mes, una vez realizado se desplegará la lista de clientes

5.- El botón te permite mostrar el historial de consumo del cliente seleccionado

6.- El botón te permite generar el aviso de cobranza del cliente seleccionado

7.- Para realizar la cobranza debe hacer clic en el botón **COBRAR**.

8.- Si, es impresión de boleta de pago debe hacer clic en botón **IMPRIMIR**

SIAPLA

Marcos Colque

Administrador

Seleccione Gestión y Mes para Generar Detalle de Consumo

GESTION: 2020 MES: Abril

Botón Cobrar (7) Botón Imprimir (8)

Seleccionar Año (3) Seleccionar mes (4)

DETALLE DE CONSUMO

Show 10 entries

Buscador (9)

Botón Genera Aviso de Cobranza (5)

Click (1)

Click (2)

Botón Historial Pagos (6)

CEDULA IDENTIDAD	NOMBRE COMPLETO	N° DE MEDIDOR	DIRECCION DE DOMICILIO	CATEGORIA	ESTADO
70619182	Justina Tarqui Tola	2055	Laka Pucara - Chiji uma	Domestico	PAGADO
71629283	Maribel Alaro Vargas	3041	Laka Pucara - Apacheta	Domestico	PAGADO
77772020	Juanito Apaza Claros	2061	Comunidad Janko Marca Sirpa	Domestico	PENDIENTE

Comité de Agua Potable y Saneamiento Laka Pucara
Comunidad Laka Pucara
Celular: 69886683

DATOS DEL CLIENTE

Nombre: Juanito Apaza Claros
Dirección: Comunidad Janko Marca Sirpa

Mes/Año	Nit/Ci	N° Medidor	Categoría
Abril/2020	77772020	2061	Domestico

Lectura	Fecha	Consumo
33	Anterior: 2020-04-06	14
47	Actual: 2020-05-07	

Historial				Detalle	Importe
Mes	Cons	Monto	Obs		
Abr/20	14	15.00		Consumo de Agua	37.00
Mar/20	8	8.00			
Feb/20	12	14.00			
Ene/20	6	8.00	Cancelado		

TREINTA Y SIETE CON 00/100 Bolivianos 37.00

DEUDA MAYOR A DOS MESES DARA LUGAR A CORTE DE SERVICIO DE AGUA

Botón Cerrar (CERRAR) Botón Imprimir (IMPRIMIR)

5.14. Cobranza (Tesorera)

- 1.- Para realizar la cobranza de consumo de agua, debe introducir el número del medidor y automáticamente le mostrara los datos del cliente y la planilla de consumo.
- 2.- Le mostrara el monto total a pagar por el consumo.
- 3.- Hacer clic en el botón cerrar y automáticamente se cerrará el formulario.
- 4.- Para realizar la cobranza se debe hacer clic en botón cobrar, y le mostrará un mensaje de confirmación, luego de manera automática se redirigirá a nueva pestaña para su comprobante de pago en formato pdf y luego para imprimirlo.

FORMULARIO DE COBRANZA

Para realizar el cobro correspondiente se debe ingresar el Numero de Medidor del Cliente.

Introduzca Numero de Medidor: 2051

Nº MEDIDOR: 2051 CARNET: 66578778 NUMERO COMPROBANTE: 001009

NOMBRE: Fernando PATERNO: Alcoocer MATERNO: Teran

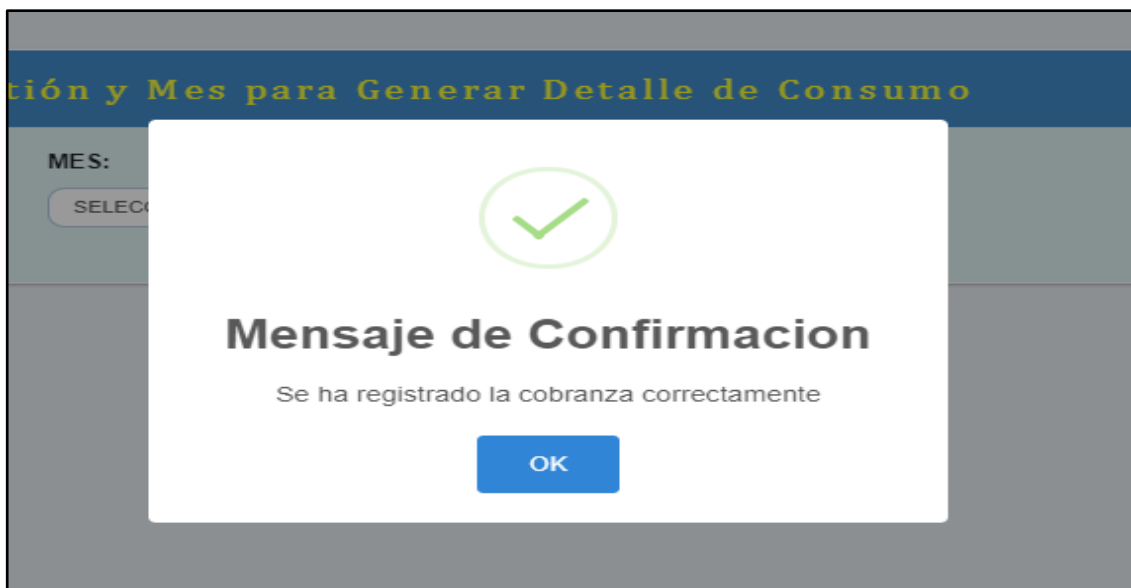
DIRECCION DE DOMICILIO: Laka Pucara - Añawani


PLANILLA DE CONSUMO

Cancelar?	Consumo M3	Gestion	Mes	Fecha_Lectura	Estado	SubTotal
<input type="checkbox"/>	8	2019	Agosto	2019-08-31	activo	8.00 Bs.
						Total a Pagar: 8.00 Bs.

SON: OCHO CON 00/100 Bolivianos

3 CERRAR **COBRAR 4**



<p>CAPYSLP Comité de Agua Potable Laka Pucara La Paz - Bolivia Teléfono: +58169818600</p>	<p>ORIGINAL Fecha: 2020-08-04 Número: 1009</p>																		
<p><u>BOLETA DE COMPROBANTE DE PAGOS</u></p>																			
<p>DATOS DEL CLIENTE NOMBRE: Fernando Alcocer Teran DIRECCION: Laka Pucara - Añawani CATEGORIA: Domestico N° MEDIDOR: 2051</p>																			
<p>DETALLE CONSUMO</p> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr style="background-color: #2e8b57; color: white;"> <th>N°</th> <th>Consumo M3</th> <th>Mes/Año</th> <th>Fecha de Pago</th> <th>Descripción</th> <th>Importe</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>8</td> <td>Agosto/2019</td> <td>2020-07-03</td> <td>cancelado</td> <td>8.00</td> </tr> <tr> <td colspan="5" style="text-align: right;">Total:</td> <td>8.00</td> </tr> </tbody> </table>		N°	Consumo M3	Mes/Año	Fecha de Pago	Descripción	Importe	1	8	Agosto/2019	2020-07-03	cancelado	8.00	Total:					8.00
N°	Consumo M3	Mes/Año	Fecha de Pago	Descripción	Importe														
1	8	Agosto/2019	2020-07-03	cancelado	8.00														
Total:					8.00														
<p><i>SON: OCHO CON 00/100 Bolivianos</i></p>																			

5.15. Consulta de Multas (Administrador)

- 1.- Hacer clic en el menú **DETALLE CONSUMO**, luego se desplegará los submenús.
- 2.- Hacer clic en submenú **CONSULTA DEUDAS**, luego mostrara la lista de clientes suspendidos.
- 3.- Te permite agregar clientes que han incumplido las normativas vigentes.

- 4.- Lista todos los clientes que hayan tenido mora o deudas más de 3 meses.
- 5.- Lista los clientes que han sido sancionados con algún tipo de multa.
- 6.- Lista los clientes que hayan cancelados sus sanciones, y luego para imprimir.

ADMINISTRACION DE MULTAS

DEUDAS PENDIENTES DE CONSUMO EN MORA ADMINISTRACION DE CLIENTES SUSPENDIDOS IMPRIMIR BOLETAS

+AGREGAR

Buscar

N°	CEDULA IDENTIDAD	NOMBRE_COMPLETO	N°_DE_MEDIDOR	FECHA_SUSPENDIDO	TIPO_MULTA	MONTO	ESTADO	OPCION
1	77772021	Rosa Patzi Tarqui	2062	2020-07-01	Reconexión sin autorización	100.00Bs.	INACTIVO	CD B
2	77772021	Rosa Patzi Tarqui	2062	2020-07-01	Reconexión sin autorización	100.00Bs.	INACTIVO	CD B
3	77772020	Juanito Apaza Claros	2061	2020-07-01	Conexión clandestina	200.00Bs.	ACTIVO	CD B
4	77772020	Juanito Apaza Claros	2061	2020-06-24	Reconexión sin autorización	100.00Bs.	INACTIVO	CD B
5	77772020	Juanito Apaza Claros	2061	2020-06-14	Corte de agua por morosidad	50.00Bs.	INACTIVO	CD B

PREVIOUS 1 NEXT

5.16. Reportes (Administrador)

- 1.- Debe seleccionar o hacer el clic en el menú **REPORTES**, luego visualizara el panel de reportes.
- 2.- En el panel donde se debe elegir el reporte que requiera el usuario.
- 3.- Luego le mostrará una ventana donde te pedirá que selecciones la gestión para generar el reporte.
- 4.- Finalmente si existe los datos con el año seleccionado, se desplegará una lista y se puede hacer clic para generar el reporte en archivo pdf.

CONTROL DE PAGO DE TARIFAS DE AGUA POTABLE GESTIÓN 2020

N°	NOMBRE USUARIO	N° MEDIDOR	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	Nov	Dic	Total Bs.
1	Apaza Tola Juan	1031	8.00	21.50											29.50
2	Tarqui Tola Justina	2055	8.00	8.00	16.00	8.00	8.00	35.00	21.50	17.00					121.50
3	Apaza Claros Juanito	2061	8.00												8.00
4	Patzi Tarqui Rosa	2062	8.00	8.00	22.50	52.50	13.50	24.00	14.00	8.00					150.50
5	Alaro Vargas Maribel	3041	17.00	17.00	8.00	18.50									60.50
TOTALES POR MESES			49.00	54.50	46.50	79.00	21.50	59.00	35.50	25.00	0	0	0	0	370.00

5.17. BACKUPS (Administrador)

1.- Se debe hacer clic en Menú Backups.

2.- Para descargar la copia de seguridad se deberá hacer clic en la imagen y automáticamente se descargará el archivo en .ZIP, además de descargar en local también se guarda en el servidor.

3.- Archivo descargado.

SIAPLA

Search...

Marcos Colque

Marcos Colque
Administrador


MENÚ

- INICIO
- ACCESO
- SUSCRIPCIONES
- MANTENIMIENTO
- LECTURAS
- DETALLE DE CONSUMO
- REPORTES
- BACKUPS**

COPIA DE SEGURIDAD

⚠ La información es de suma importancia, es mucho mejor prevenir que lamentar. Es por ello que siempre recomiendo realizar respaldo periódicamente.

Para realizar la copia de seguridad haga clic en la imagen:



1 clic aquí

2 clic para descargar

3 archivo backup en .ZIP

backup-on-2020-0...zip

Mostrar todo

ANEXO D

CÓDIGO FUENTE: CONTROLADOR DE DETALLE CONSUMO

```
class Controller_detalle_consumo extends CI_Controller
{
    function __construct()
    {
        parent::__construct();
        if ($this->session->userdata('is_logued_in')!=true && $this->session->userdata('estado')!='activo') {
            redirect(base_url());
        }else{
            $this->load->library('session');
            $this->load->model('Modelo_seguridad');
            $this->permisos = $this->backend_lib->control();
            $this->menu = $this->backend_lib->lista_meus_sys();
        }
        $this->load->model("Model_Gestion");
        $this->load->library("pdf");
    }
    public function index(){
        redirect(base_url()."Controller_detalle_consumo/admin_consumo");
    }
    public function admin_detalle_consumo(){
        if ($this->permisos->tbl_nombre=='admin_detalle_consumo') {
            $data['permisos']=$this->permisos;
            $data['menu']=$this->menu;
            $data['contenido']="administra_detalle_consumo/form_lista_detalle_consumo";
            $this->load->view("plantilla",$data);
        }else{
            redirect(base_url()."Controller_detalle_consumo/admin_consumo");
        }
    }
    public function selecciona_datos_consumo(){
        $gestion=$this->input->post("ANO");
        $mes=$this->input->post("MES");
        $dato['obj']=$this->Model_Gestion->lista_planilla_pagos($gestion,$mes);
        $this->load->view("administra_detalle_consumo/tabla_lista_consumo",$dato);
    }
    public function busca_numero_medidor(){
        $obj=$this->Model_Gestion->verifica_medidor($this->input->post("num_medidor"));
        $veri=$this->Model_Gestion->verificar_deuda_medidor_cobros($this->input->post("num_medidor"));
        if ($obj==1) {
            $data['obj1']=$this->Model_Gestion->recupera_datos_cliente($this->input->post("num_medidor"));
            $data['obj2']=$this->Model_Gestion->recupera_datos_cliente_consumo($this->input->post("num_medidor"));
            $data['obj3']=$this->Model_Gestion->extrae_maximo_numcomprobante();
            $this->load->view("administra_detalle_consumo/form_cobros_consumo",$data);
        }else{

```

```

        if (empty($veri)) {
            echo '<div class="alert alert-warning" style="color: #000; font-size: 18px;">
                <i class="fa fa-check-circle"></i><strong></strong>NO EXISTE EL NUMERO DE
MEDIDOR INGRESADO.
            </div>';
        }else{
            echo '<div class="alert alert-primary " style="background: #ec7063; color: #fff; font-size:
15px;">
                <i class="fa fa-question-circle"></i><strong> </strong> El numero de medidor introducido
                tiene deuda pendiente, primero debe regularizar sus deudas.
            </div>';
        }
    }
}

```

```

public function guardar_nueva_cobro(){
    $fecha=date('Y-m-d');
    $numcomprobante=$this->input->post("comprobante");
    $id_usuario=$this->session->userdata("idusuario");
    $idlectura_cl_medidor=$this->input->post('idlectura_cl_medidor');
    $costo_t=$this->input->post('costo_t');
    $idcosto_de_cobranza=$this->input->post('idcosto_de_cobranza');
    $this->Model_Gestion-
    >guardar_nueva_cobro($fecha,$numcomprobante,$id_usuario,$idlectura_cl_medidor,$costo_t,$idc
    osto_de_cobranza);
    echo "SE REGISTRO CORRECTAMENTE";
}

```

```

public function genera_boleta_pago(){
    $nro_medidor=$this->input->post("nro_medidor");
    $this->load->view("administra_detalle_consumo/imprimir_boleta");
}

```

```

//genera aviso de consumo//
public function genera_factura(){
    $idcliente=$this->input->post("idcliente");
    $data['obj']=$this->Model_Gestion->genera_factura_cliente($idcliente);
    $nro_medidor=$this->input->post("nro_medidor");
    $data['obj2']=$this->Model_Gestion->ultima_registro_lectura_fecha($nro_medidor);
    $data['obj3']=$this->Model_Gestion->extrae_fecha_anterior_registro_lectura($nro_medidor);
    $data['objh']=$this->Model_Gestion->lista_historial_pagos($nro_medidor);
    $this->load->view("administra_detalle_consumo/form_modal_factura",$data);
}

```

```

//genera boleta de pagos
public function busca_numero_medidor_boleta(){
    $obj=$this->Model_Gestion->verifica_medidor($this->input->post("num_medidor"));
    $verb=$this->Model_Gestion->verificar_deuda_medidor_cobros($this->input-
>post("num_medidor"));
    if ($obj==1) {
        $data['obj1']=$this->Model_Gestion->recupera_datos_cliente($this->input-
>post("num_medidor"));
        $data['obj2']=$this->Model_Gestion->boleta_pagos_datos_cliente_consumo($this->input-
>post("num_medidor"));
        $this->load->view("administra_detalle_consumo/form_comprobantes_cobrados",$data);
    }else{
        if (empty($verb)) {
            echo '<div class="alert alert-warning" style="background: #d35400; color: #fff; font-size:
18px;">
                <i class="fa fa-check-circle"></i><strong></strong>NO EXISTE EL NUMERO DE
MEDIDOR INGRESADO.
            </div>';
        }else{
            echo '<div class="alert alert-primary " style="background: #ec7063; color: #fff; font-size:
15px;">
                <i class="fa fa-question-circle"></i><strong> </strong> El numero de medidor introducido
tiene deuda pendiente, primero debe regularizar sus deudas.
            </div>';
        }
    }
}
}
}

```

ANEXO E

AVAL DE CONFORMIDAD

El Alto, 8 de Julio del 2020

Señor:

Ing. Enrique Flores Baltazar
TUTOR METODOLÓGICO TALLER II

Presente. -

Ref. Aval de conformidad

Distinguido Ingeniero:

Mediante la presente tengo a bien comunicarle mi conformidad del proyecto de grado titulado " SISTEMA DE INFORMACIÓN PARA EL CONTROL DE PAGOS POR EL SUMINISTRO DE AGUA POTABLE CASO: COMITÉ DE AGUA POTABLE LAKA PUCARA", que propone el postulante Univ. Marcos Colque Ticona, con cédula de identidad 8436244 LP. para su defensa pública, evaluación correspondiente a la materia Taller de Licenciatura II, de acuerdo a reglamento vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, reciba saludos cordiales.

Atentamente,



Ing. Gabriel Reynaldo Sirpa-Huayhua
TUTOR REVISOR

AVAL DE CONFORMIDAD

El Alto, 9 de Julio del 2020

Señor:
Ing. Enrique Flores Baltazar
TUTOR METODOLÓGICO TALLER II

Presente. -

Ref. Aval de conformidad

Distinguido Ingeniero:

Mediante la presente tengo a bien comunicarle mi conformidad del proyecto de grado titulado ' SISTEMA DE INFORMACIÓN PARA EL CONTROL DE PAGOS POR EL SUMINISTRO DE AGUA POTABLE CASO: COMITÉ DE AGUA POTABLE LAKA PUCARA' que propone el postulante Univ. Marcos Colque Ticono, con cédula de identidad 8436244 LP. para su defensa pública, evaluación correspondiente a la materia Taller de Licenciatura II, de acuerdo a reglamento vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, reciba saludos cordiales.

Atentamente.



Ing. Juan Regis Muñoz Sirpa
TUTOR ESPECIALISTA

Laka Pucara, julio de 2020

Señor:
Ing. David Carlos Mamani Quispe
**DIRECTOR DE CARRERA INGENIERIA DE SISTEMAS
UNIVERSIDAD PÚBLICA DE EL ALTO**

PRESENTE.-

REF.: AVAL DE CONFORMIDAD

Distinguido Ingeniero:

Por medio de la presente tengo a bien comunicarle mi conformidad del proyecto de grado, para obtención del título en Ingeniería de Sistemas de la Universidad Pública de El Alto, titulado "**SISTEMA DE INFORMACIÓN PARA EL CONTROL DE PAGOS POR EL SUMINISTRO DE AGUA POTABLE CASO: COMITÉ DE AGUA POTABLE LAKA PUCARA**", fue desarrollado en su totalidad y se ha podido constatar que cumple con todo los requerimientos preestablecidos para el presente trabajo, por el Univ. MARCOS COLQUE TICONA con cédula de identidad 8436244 LP, habiendo demostrado responsabilidad y compromiso durante la elaboración del proyecto.

Sin otro particular, reciba saludos cordiales.




Presidente del
COMITÉ DE AGUA
LAKA PUCARA

AVAL DE CONFORMIDAD

El Alto, 28 de Agosto del 2020

Señor:

Ing. David Carlos Mamani Quispe
TUTOR METODOLÓGICO TALLER II

Presente. -

Ref. Aval de conformidad

Distinguido Ingeniero:

Mediante la presente tengo a bien comunicarle mi conformidad del proyecto de grado titulado "**SISTEMA DE INFORMACIÓN PARA EL CONTROL DE PAGOS POR EL SUMINISTRO DE AGUA POTABLE CASO: COMITÉ DE AGUA POTABLE LAKA PUCARA**", que propone el postulante Univ. Marcos Colque Ticona, con R.U.11001111 y cédula de identidad 8436244 LP. para su defensa pública, evaluación correspondiente a la materia Taller de Licenciatura II, de acuerdo a reglamento vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, reciba saludos cordiales.

Atentamente,



Ing. Enrique Flores Baltazar

TUTOR METODOLÓGICO TALLER II