

UNIVERSIDAD PÚBLICA DE EL ALTO

CARRERA INGENIERÍA DE SISTEMAS



PROYECTO DE GRADO

“SISTEMA DE INFORMACIÓN WEB DE CONTROL Y SEGUIMIENTO PARA LA ADMINISTRACIÓN DE SERVICIOS DE INTERNET Y TV CABLE”

CASO: FENIX TELECOM

Para Optar al Título de Licenciatura en Ingeniería de Sistemas

MENCIÓN: INFORMÁTICA Y COMUNICACIONES

Postulante: Jhonny Choque Condori

Tutor Metodológico: M.Sc. Lic. Ing. Marisol Arguedas Balladares

Tutor Especialista: Ing. Noemy Mamani Chuyma

Tutor revisor: M. Sc. Lic. Carmen Vega Flores

EL ALTO – BOLIVIA

2024

DEDICATORIA

Dedico el presente proyecto principalmente a Dios, por darme la fuerza necesaria para culminar esta meta.

A mis padres, por acompañarme en cada paso que doy en la búsqueda de ser mejor persona, profesional y por motivarme a seguir hacia adelante.

A todos mis amigos y compañeros por su ayuda desinteresada durante el transcurso de nuestro tiempo en la carrera.

AGRADECIMIENTOS

Agradezco a Dios por siempre cuidarme, por darme fuerza, salud e inteligencia para culminar con mi trabajo de grado.

A mi padres y hermanos por todo el apoyo incondicional que me brindaron en todo el transcurso de mi carrera.

A mis tutores por su tiempo y paciencia en la revisión del trabajo de grado.

RESUMEN

En los últimos años el servicio de internet, se ha considerado como un servicio básico para la sociedad, como es el servicio de agua, gas y luz, Fénix Telecom es un proveedor de servicios de Internet y Televisión, el cual brinda sus servicios en los departamentos de Cochabamba, Oruro y también en La Paz, sin embargo la empresa necesita promocionar los distintos paquetes de servicios que ofrece, también desea centralizar los datos de los clientes, obtener información en el menor tiempo posible, para que de esta manera se pueda brindar servicios eficientes a todos los clientes de la empresa. Por esta razón es que se desarrolla el Sistema de Información Web de Control y Seguimiento para la administración de servicios de internet y tv cable, para ello se utilizó tecnologías emergentes de software libre como MySQL, un sistema de gestión de bases de datos relacionales, para el almacenamiento de los datos. Laravel en su última versión, es un framework de desarrollo y diseño web basado en PHP, que proporciona una estructura sólida y herramientas avanzadas para la creación de aplicaciones web eficientes. La metodología que se aplicó es UWE (UML - Based Web Engineering), ya que sus fases se adaptan y cumplen con los requisitos para el desarrollo del sistema. En relación a la calidad se aplicó la norma ISO 25000, que permite realizar una evaluación exhaustiva del sistema y asegurar su cumplimiento de los estándares de calidad establecidos. El presente proyecto se desarrolló con una interfaz gráfica amigable diseñada para ofrecer una experiencia intuitiva y fácil de usar para los usuarios, mediante una navegación sencilla.

Palabras clave: framework, metodología, norma.

SUMMARY

In recent years, internet service has been regarded as a basic necessity for society, similar to water, gas, and electricity services. Fénix Telecom is a provider of Internet and Television services, offering its services in the departments of Cochabamba, Oruro, and also in La Paz. However, the company needs to promote the various service packages it offers, centralize customer data, and obtain information as quickly as possible to provide efficient services to all its customers. For this reason, the Web Information System for Control and Monitoring was developed for managing internet and cable TV services. This system utilizes emerging open-source software technologies, such as MariaDB, a relational database management system, for data storage. The latest version of Laravel, a web development and design framework based on PHP, was used to provide a robust structure and advanced tools for creating efficient web applications. The methodology applied is UWE (UML-Based Web Engineering), as its phases adapt to and meet the requirements for system development. Regarding quality, the ISO 25000 standard was applied to conduct a comprehensive system evaluation and ensure compliance with established quality standards. This project was developed with a user-friendly graphical interface designed to offer an intuitive and easy-to-use experience through simple navigation.

Keywords: framework, methodology, standard.

ÍNDICE

1. MARCO PRELIMINAR.....	1
1.1. INTRODUCCIÓN.....	1
1.2. ANTECEDENTES	2
1.2.1. Antecedentes Institucionales	2
1.2.2. Antecedentes afines al Proyecto de Grado.....	4
1.3. PLANTEAMIENTO DEL PROBLEMA.....	6
1.3.1. Problema Principal.....	6
1.3.2. Problemas Secundarios.....	6
1.3.3. Formulación del Problema	7
1.4. OBJETIVOS.....	7
1.4.1. Objetivo General.....	7
1.4.2. Objetivos Específicos	8
1.5. JUSTIFICACIÓN	8
1.5.1. Justificación Técnica.....	9
1.5.2. Justificación Económica	9
1.5.3. Justificación Social	9
1.6. METODOLOGÍA	9
1.6.1. Metodología UWE	9
1.6.2. Métricas de Calidad	10
1.6.3. Costos	10
1.6.4. Seguridad	11
1.6.5. Pruebas de Software	12
1.7. HERRAMIENTAS.....	13
1.7.1. Xampp.....	13
1.7.2. Laravel.....	13
1.7.3. Mysql.....	13
1.7.4. Ajax	14
1.7.5. JavaScript.....	14
1.8. LÍMITES Y ALCANCES	14
1.8.1. Límites.....	14

1.8.2.	Alcances.....	15
1.9.	APORTES.....	15
2.	MARCO TEÓRICO.....	16
2.1.	INTRODUCCIÓN.....	16
2.2.	WEB 2.0.....	16
2.2.1.	Características de la Web 2.0.....	16
2.2.2.	Usos de la Web 2.0.....	17
2.3.	METODOLOGÍA INGENIERÍA WEB BASADA EN UML (UWE).....	18
2.3.1.	Características.....	19
2.3.2.	Modelos de la metodología UWE.....	20
2.4.	ARQUITECTURA DE SOFTWARE (MVC).....	28
2.4.1.	Modelo.....	29
2.4.2.	Vista.....	29
2.4.3.	Controlador.....	30
2.4.4.	Características del Modelo Vista Controlador.....	31
2.5.	HERRAMIENTAS.....	32
2.5.1.	MagicDraw.....	32
2.5.2.	Gestor de base de datos MySql.....	34
2.5.3.	Lenguaje de Programación PHP.....	37
2.5.4.	Bootstrap.....	39
2.5.5.	JavaScript.....	40
2.5.6.	jQuery.....	41
2.6.	MÉTRICAS DE CALIDAD DE SOFTWARE.....	42
2.6.1.	Calidad de Software.....	43
2.6.2.	Norma ISO/IEC 25000.....	44
2.6.3.	Clasificación de las características y sub características.....	46
2.7.	ANÁLISIS DE COSTOS.....	51
2.7.1.	Cocoma II.....	51
2.7.2.	Características.....	52
2.7.3.	Ecuaciones del modelo Cocoma.....	52
2.7.4.	Modelos de COCOMO II.....	53

2.8. SEGURIDAD	56
2.8.1. SGSI (Sistema de Gestión de la Seguridad de la Información)	56
2.8.2. Aspectos de seguridad	57
2.9. PRUEBAS DE SOFTWARE	60
2.9.1. Pruebas de caja blanca	60
2.9.2. Pruebas de ruta básica	60
2.9.3. Pruebas de caja negra	62
2.9.4. Pruebas de estrés	64
3. MARCO APLICATIVO	62
3.1. INTRODUCCIÓN	62
3.2. DESARROLLO DE LA METODOLOGÍA	62
3.2.1. Requerimientos Funcionales	63
3.2.2. Requerimientos no Funcionales	64
3.2.3. Descripción de Actores	65
3.2.4. Diagrama de Caso de Uso General del Sistema	66
3.2.5. Diagramas de Casos de Uso Específicos del Sistema	67
3.2.6. Diseño conceptual	78
3.2.8. Modelo de navegación	79
3.2.9. Modelo de presentación	81
3.2.10. Modelo de implementación	82
4. CALIDAD, PRUEBAS, COSTOS Y SEGURIDAD DEL SOFTWARE	83
4.1. MÉTRICAS DE CALIDAD	83
4.1.1. Funcionalidad	83
4.1.2. Confiabilidad	89
4.1.3. Usabilidad	91
4.1.4. Eficiencia	93
4.1.5. Mantenibilidad	95
4.1.6. Portabilidad	96
4.2. EVALUACIÓN DE COSTOS	98
4.3. SEGURIDAD	102
4.3.1. Seguridad lógica	102

4.3.2.	Seguridad Física.....	103
4.4.	PRUEBAS AL SOFTWARE	104
4.4.1.	Prueba de caja blanca	104
4.4.2.	Prueba de caja negra	95
4.4.3.	Prueba de estrés	115
5.	CONCLUSIONES Y RECOMENDACIONES.....	117
5.1.	CONCLUSIONES.....	117
5.2.	RECOMENDACIONES.....	117
	BIBLIOGRAFÍA.....	119
	ANEXOS	<i>¡Error! Marcador no definido.</i>

ÍNDICE DE TABLAS

Tabla 1. Tabla de entradas de usuario.....	47
Tabla 2. Tabla de salidas de usuario	48
Tabla 3. Tabla de peticiones de usuario.....	48
Tabla 4. Índices	50
Tabla 5 Coeficientes del modelo Básico.	53
Tabla 6 Coeficientes del modelo intermedio.	54
Tabla 7 Atributos de costes totales.	55
Tabla 8 Herramientas para la obtención de requerimientos	62
Tabla 9 Requerimientos Funcionales.....	63
Tabla 10 Requerimientos no funcionales	64
Tabla 11 Descripción de Actores	65
Tabla 12 Descripción del caso de uso - Gestión de usuarios.....	68
Tabla 13 Descripción de caso de uso de gestionar paquetes	70
Tabla 14 Descripción de caso de uso de administradores.	71
Tabla 15 Descripción de caso de uso de gestionar presupuestos.....	73
Tabla 16 Descripción de caso de uso de gestión de servicios adquiridos	74
Tabla 17 Caso de uso de registro de solicitud.....	76
Tabla 18 Diseño conceptual.....	78
Tabla 19 Número de interacciones de usuario	84
Tabla 20 Cantidad de salidas de usuario	84
Tabla 21 Cantidad de peticiones del usuario.	85
Tabla 22 Parámetros de puntos función.....	86
Tabla 23 Punto de función bruto.	86
Tabla 24 Variables de complejidad	87

Tabla 25 Indicadores de facilidad de uso	91
Tabla 26 Usabilidad del sistema	92
Tabla 27 Escala de valores de eficiencia.	93
Tabla 28 Evaluación de la eficacia del sistema.....	94
Tabla 29 Indicadores de mantenibilidad.....	95
Tabla 30 Resultados finales.....	98
Tabla 31 Coeficientes de a, b, c y d de COCOMO II.....	99
Tabla 32 Ecuaciones cocomo II	100
Tabla 33 Cálculo de Atributos FAE.	100
Tabla 34 Prueba de acceso al sistema.	95
Tabla 35 Caso de uso de prueba de registro de usuario.....	114

ÍNDICE DE FIGURAS

Figura 1 Organigrama FÉNIX TELECOM.	3
Figura 2 Modelo de análisis de requerimientos.	23
Figura 3 Modelo de Caso de Uso de la Ingeniería Web.	24
Figura 4 Modelo Contenido de la Ingeniería Web.	25
Figura 5 Modelo Navegación de la Ingeniería Web.	26
Figura 6 Modelo de Presentación de la Ingeniería Web.	27
Figura 7 Flujo de MVC.	31
Figura 8 División de calidad de Software.	45
Figura 9 Diagrama de flujo.	60
Figura 10 Grafo de flujo.	61
Figura 11 Matriz de Grafo.	62
Figura 12 Caso de uso general del sistema.	66
Figura 13 Gestión de usuarios.	68
Figura 14 Caso de uso Gestión de paquetes.	69
Figura 15 Diagrama de caso de uso de administradores.	71
Figura 16 Diagrama de caso de uso asignación de cuadrillas.	72
Figura 17 Caso de uso de servicios adquiridos.	74
Figura 18 Caso de uso Registro de solicitudes.	75
Figura 19 Modelo navegacional - Administrador.	79
Figura 20 Modelo navegacional - Plataforma.	80
Figura 21 Modelo navegacional - Cuadrilla.	80
Figura 22 Modelo de presentación Administrador.	81
Figura 23 Página principal FENIX.	82
Figura 24 Página principal servicios.	83

Figura 25 Página principal planes	84
Figura 26 Iniciar Sesión FENIX.....	84
Figura 27 Prueba de caja blanca	104

TABLA DE ECUACIONES

(1) Fórmula de ajuste.....	88
(2) Probabilidad de fallas	90
(3) Probabilidad de trabajo sin fallas	90
(4) Función de la confiabilidad del sistema	90
(5) Función para determinar la usabilidad	91
(6) Fórmula eficacia	94
(7) Ecuación para el índice de madurez del sistema	95
(8) Fórmula portabilidad	97

CAPÍTULO I

1. MARCO PRELIMINAR

1.1. INTRODUCCIÓN

La importancia de la tecnología hoy en día se ha convertido en una herramienta muy importante para la sociedad, por lo que al Internet podríamos incluirlo a la lista de servicios básicos esenciales, como ser el servicio del agua, luz etc. En tiempos de pandemia se pudo evidenciar la gran demanda del servicio de Internet para las distintas actividades de trabajo y estudios, se pudo constatar que las diferentes empresas proveedoras no satisfacían estas necesidades, ni cumplían con las expectativas de banda ancha de los usuarios. La empresa proveedora de servicios de Internet y Tv. Cable FENIX TELECOM ofrece sus servicios en los departamentos de La Paz y Cochabamba.

El problema con las nuevas tecnologías en la sociedad actualmente, es que estas se van actualizando constantemente, por lo que el flujo de información a través del Internet, está transformando las actividades cotidianas de las empresas, estudiantes, trabajadores, etc. Que permiten mejorar la productividad tanto instituciones públicas y privadas a nivel nacional e internacional de manera eficiente, siendo un apoyo en los procesos administrativos.

La solución que se presenta en este trabajo de grado es diseñar e implementar un Sistema de Información Web para la empresa, que dé a conocer los servicios que ofrece FENIX TELECOM, de esta manera expandir el mercado a grandes oportunidades de crecimiento tanto económicamente como institucionalmente.

FENIX TELECOM SRL es una empresa proveedora de servicios de tv cable e internet, que a lo largo de los años han brindado una atención satisfactoria a sus clientes, sin embargo la empresa se va expandiendo, por lo cual hay nuevas áreas de cobertura, y para ofrecer una atención de calidad y de manera óptima, se necesita implementar un Sistema Web de Información que pueda gestionar la información y las actividades que cumple el negocio, se propone el desarrollo del mencionado sistema web, que podrá cumplir con los requerimientos

específicos por la empresa, aplicando la metodología de desarrollo UWE, también herramientas de desarrollo web como ser el framework Laravel y con el gestor de base de datos MariaDB,

1.2. ANTECEDENTES

1.2.1. Antecedentes Institucionales

La empresa FENIX-TELECOM.SRL creada el 20 de abril de 2018 cuenta con varios años de experiencia en cuanto servicios de internet y televisión, siendo los departamentos de La Paz y Cochabamba, en sus principales ciudades donde se provee los servicios de la empresa.

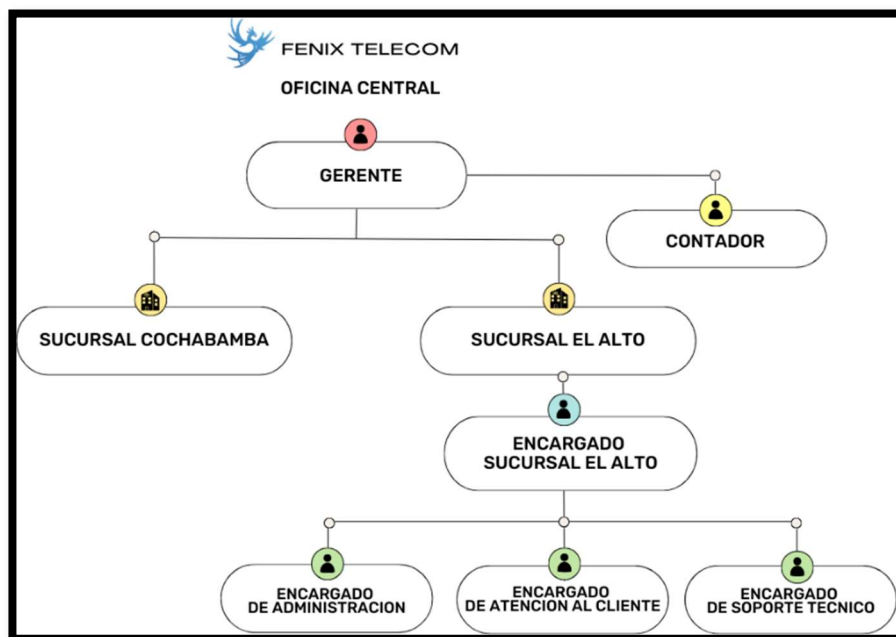
1.2.1.1. Visión. En FENIX-TELECOM.SRL, aspiramos a ser reconocidos como el proveedor líder de servicios de internet, destacando por nuestra capacidad para anticipar y superar las expectativas de nuestros clientes. Buscamos ser pioneros en la introducción de tecnologías emergentes que redefinan los estándares de conectividad, permitiendo un acceso rápido, confiable y seguro a la información global. Con una visión centrada en el futuro, nos esforzamos por ser un catalizador en la construcción de un mundo más conectado, donde la innovación y la accesibilidad digital sean el futuro de hoy.

1.2.1.2. Misión. En FENIX-TELECOM.SRL, nos comprometemos a liderar la revolución digital al proporcionar servicios de internet excepcionales que transformen la vida de las personas y potencien sus experiencias en línea. Nuestra misión es conectar comunidades, hogares y empresas mediante soluciones de conectividad innovadoras y confiables, impulsadas por la excelencia en el servicio al cliente y la tecnología de vanguardia. Estamos dedicados a ser el puente digital que abre un mundo de posibilidades, facilitando la comunicación, el aprendizaje, el trabajo y el entretenimiento.

1.2.1.3. Organigrama. A continuación, se muestra la organización de cargos dentro de la empresa Fénix Telecom.

Figura 1

Organigrama FÉNIX TELECOM.



Nota. Esquema de cargos de la Empresa Fénix TELECOM.

1.2.2. Antecedentes afines al Proyecto de Grado

Haciendo referencia a los trabajos realizados a nivel internacional, nacional se pueden citar a las siguientes:

1.2.2.1. Antecedentes Nacionales

- Marisol Linda Chamaco Mamani (2023), “SISTEMA DE GESTIÓN Y SEGUIMIENTO A CLIENTES PARA LOS SERVICIOS DE INTERNET Y TV CABLE”, el proyecto de grado, que tiene como propósito mejorar la experiencia de atención a clientes de la empresa HLZ que oferta los servicios de internet y televisión por cable, mediante un sistema gestión de información que permite automatizar los diferentes procesos manuales de atención a los clientes. En este sentido se desarrolló un sistema de gestión de información que permite automatizar los diferentes procesos manuales de atención a los clientes. En este sentido se desarrolló un sistema de gestión de información con diferentes módulos, que permite al usuario realizar las siguientes actividades: registro de servicios, registro de pagos, baja de servicio y generación de tickets para asistencia técnica.
- Kevin Luis Cahuaya Catari, (2023), “AMPLIACIÓN DE LA COBERTURA DEL SERVICIO DE BANDA ANCHA VDSL MEDIANTE UNA RED DE TRANSPARENTE INALÁMBRICO DE ALTA CAPACIDAD PARA LA LOCALIDAD DE COPACABANA”, el proyecto tiene el propósito de ampliar la cobertura de Internet de banda ancha VDSL, en la localidad de Copacabana, con el fin de mejorar la calidad de servicio y tener **más** opciones para una buena conexión a Internet, donde resulta decisiva para el desarrollo económico, educación y salud. El proyecto tiene estudio de caso para la empresa AXS Bolivia.

1.2.2.2. *Antecedentes Internacionales*

- Solange Paola Álvarez Cisneros, Christian Iván Caguana Carrillo (2021), “DISEÑO DE UNA PROPUESTA DE SERVITIZACIÓN DE LOS SERVICIOS PARA IMPULSAR LAS ESTRATEGIAS COMERCIALES COMPETITIVAS EN EL GRUPO TV CABLE DE LA CIUDAD DE AMBATO”, la presente investigación fue desarrollada con el objetivo de implementar un proceso de servitización que permita impulsar las estrategias comerciales competitivas en el Grupo Tv Cable de la ciudad de Ambato, de tal forma que se logre solucionar el bajo índice de ingresos durante la pandemia del COVID-19. El estudio posee un enfoque cuantitativo; bajo un paradigma crítico propositivo debido a que se plantea una solución al problema mediante un proceso de servitización; asimismo se basó en un modelo de investigación descriptiva y aplicada, haciendo uso de la técnica de la encuesta a través de la cual los resultados demostraron que el estrato A de clientes de Grupo Tv Cable posee mayor ineficiencia en asistencia técnica.
- Mauricio Merino (2020), “ADMINISTRACIÓN DE PROYECTOS EN UNA EMPRESA DE TELECOMUNICACIONES UBICADA EN LA CIUDAD DE QUITO”, el presente trabajo, tiene como objetivo implementar una oficina de administración de proyectos, de acuerdo al diagnóstico situacional, realizado en el presente trabajo a través de un análisis como ser: 5 Fuerzas de Porter, Análisis Pestel, matriz FODA, se pudo determinar los factores de éxito tanto de empresa como la industria que sirvieron para determinar que una PMO es vital en una organización.

1.3. PLANTEAMIENTO DEL PROBLEMA

El uso de las tecnologías en la sociedad, se ha vuelto muy imprescindible, tanto que, para el uso de este, se necesita una conexión de internet, es por eso que las personas van buscando un proveedor de servicios de banda ancha, conforme a las necesidades de los clientes. La empresa FENIX-TELECOM.SRL viene trabajando muchos años como distribuidor de los servicios de internet, sin embargo, esta empresa no se tiene la información centralizada de todos los clientes, sus estados de pago, lista de ofertas y otros.

1.3.1. Problema Principal

La empresa FENIX-TELECOM.SRL durante el transcurso que ha ido prestando servicios de internet y tv cable, se evidencio la ausencia de organización, planificación y promoción de los servicios que este ofrece, la información se encuentra descentralizada, lo que no permite realizar un control y seguimiento a los datos de los clientes que consumen los servicios de dicha empresa, consecuentemente eso conlleva a pérdidas económicas por falta de toma de decisiones correctas (ver anexo A).

1.3.2. Problemas Secundarios

- No tiene una información ordenada y organizada lo que ocasiona la duplicidad de los datos en los registros.
- Deficiencia del control de los servicios de los clientes y el control de pagos, lo que causa perdidas a la empresa.
- No se cuenta con una página web, donde se brinde la información de todos los paquetes que ofrece la empresa, por lo cual impide el crecimiento de la empresa.

- No se cuenta con acceso inmediato a documentaciones respecto a controles los cuales afectan en la empresa.
- Riesgo de pérdida de la información que puede suscitar una pérdida económica considerable en la empresa.
- Problemas de comunicación interna que pueden afectar las metas y objetivos por cumplir en la empresa
- Falta de información estadística respecto a la cantidad de clientes y servicios en funcionamiento, para la toma de decisiones.

1.3.3. Formulación del Problema

¿De qué manera se podrá mejorar la administración y control, de toda la información respecto a los clientes, servicios de Internet y tv cable, tal que sea eficiente y permita el crecimiento de la empresa FENIX TELECOM SRL?

1.4. OBJETIVOS

El objetivo del presente proyecto de grado nace con una respuesta a las necesidades de la institución, luego de un análisis de los problemas y su relación causa – efecto se realizó el árbol de objetivos (ver anexo B).

1.4.1. Objetivo General

Desarrollar un Sistema de Información Web de Control y Seguimiento para la administración de servicios de Internet y tv cable en la empresa FENIX TELECOM SRL, que genere información confiable y oportuna, con el fin de mejorar la gestión de la información de los clientes y servicios.

1.4.2. Objetivos Específicos

- Centralizar una base de datos relacional, para evitar la duplicidad de datos y se tenga una mejor organización.
- Administrar toda la información de los servicios, áreas de cobertura, vendedores y clientes, de tal forma que se optimice el servicio brindado a sus clientes.
- Crear perfiles de administración y clientes, para administrar las cuentas de los clientes, y estos puedan ver, cambiar o solicitar la baja de su servicio.
- Diseñar una página web con una interfaz amigable e intuitiva, que muestre toda la información de servicios, ubicaciones al cliente, esto ampliaría la cantidad de clientes potenciales para la empresa.
- Agilizar la búsqueda de los registros, y realizar copias de seguridad diaria, de tal forma que se tenga respaldos y acceso a la información 24 horas.
- Generar reportes estadísticos de lo que requiere saber la empresa, para mejorar la atención y optimizar el servicio brindado a los clientes.

1.5. JUSTIFICACIÓN

El desarrollo del Sistema de Información Web de Control y Seguimiento para la Administración de Servicios de Internet y TV Cable, que mejorará la administración de servicios y clientes de la Empresa FENIX TELECOM S.R.L., se justifica principalmente en el manejo eficiente de la información, además se justifica de manera técnica, económica y social de la siguiente forma:

1.5.1. Justificación Técnica

La empresa cuenta con equipos de alta gama, adecuados para que se pueda, acceder al Sistema Web de Control y Seguimiento para la Administración de Servicios, donde además el software para desplegar esta aplicación es de coste cero.

El sistema de control y seguimiento, permitirá a la administración de contar con un recurso tecnológico para facilitar la gestión de la información, convirtiéndose así en una herramienta útil y confiable.

1.5.2. Justificación Económica

Se justifica económicamente ya que el desarrollo del sistema se encontrará bajo la premisa de software libre, que implica la reducción de gastos en licencias de uso. Además, con el sistema, se verificará el estado de los equipos entregados y devueltos por el cliente, evitando pérdidas económicas considerables para la empresa FENIX TELECOM S.R.L.

1.5.3. Justificación Social

Sin duda que los clientes de la empresa serán beneficiadas, con este sistema que proporcionará toda la información respecto a los servicios contratados, con un interfaz de administración ordenada y organizada, mejorando la calidad de atención y servicio. Y además con la expansión de cobertura permitirá expandir sus servicios a zonas alejadas donde otras empresas no tienen cobertura.

1.6. METODOLOGÍA

1.6.1. Metodología UWE

Se aplicará para el desarrollo de la plataforma web el uso de la metodología de desarrollo UWE (Unified Modeling Language). (Ludwig, 2014)

Para su implementación se debe seguir las siguientes etapas:

Análisis de Requerimientos plasma los requisitos funcionales de la aplicación web mediante un modelo de casos de Uso.

- Modelo de Contenido mediante un modelo de clase se define los conceptos a detalle que relación en el sistema.
- Modelo de Navegación en esta etapa se representa la navegación de los objetos dentro de la aplicación un conjunto de estructuras como son índices menús y consultas
- Modelo de presentación representa las interfaces de usuario por medio de vistas abstractas
- Modelo de proceso representa el aspecto que tienen las actividades que se conectan con cada clase de proceso.

1.6.2. Métricas de Calidad

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

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

1.6.3. Costos

La estimación del costo para el desarrollo del software se puede aplicar el método Cocomo II que está basado en cálculo del costo mediante el tiempo esfuerzo y el número de líneas de código del software, Cocomo II consta de tres modelos composición de aplicación, diseño temprano y Post Arquitectura. (Miss Juarez , 2020, pág. 1)

- Modelo de composición de aplicación que es utilizado durante la ingeniería de software, prototipado de interfaces de usuario, interacción del sistema y del software, evaluación del rendimiento y la evaluación del crecimiento de la tecnología.
- Modelo de diseño temprano que se aplica cuando se ha establecido la arquitectura del software.
- Modelo Post Arquitectura aplicado durante la construcción del software.

Estos modelos se adaptan a las necesidades como al tipo y la cantidad de la información. (Miss Juarez , 2020, p. 25)

1.6.4. Seguridad

La norma ISO-27000 es un estándar que define ciertos requisitos para un sistema de gestión de la seguridad de la información (SGSI) que tiene el objetivo de garantizar el control de la seguridad adecuada protegiendo la información de cualquier empresa, institución, organización, etc. Ofreciendo una cantidad de recomendaciones de seguridad. (Baena y otros, 2019)

Entre sus características de las normas ISO – 27000 se basan en:

- Confidencialidad
- Integridad
- Disponibilidad

1.6.5. Pruebas de Software

Las pruebas son muy importantes en la fase de implementación es por eso que es muy necesario realizarlas se propone las siguientes pruebas:

1.6.5.1. Caja Negra. Las pruebas de caja negra, también llamadas pruebas de comportamiento, se enfocan en los requerimientos funcionales del software; es decir, las técnicas de prueba de caja negra le permiten derivar conjuntos de condiciones de entrada que revisarán por completo todos los requerimientos funcionales para un programa. Las pruebas de caja negra no son una alternativa para las técnicas de caja blanca. En vez de ello, es un enfoque complementario que es probable que descubra una clase de errores diferente que los métodos de caja blanca. Las pruebas de caja negra intentan encontrar errores en las categorías siguientes: funciones incorrectas o faltantes, errores de interfaz, errores en las estructuras de datos o en el acceso a bases de datos externas, errores de comportamiento o rendimiento y errores de inicialización y terminación. (Pressman, 2010, p. 423)

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

fronteras y dentro de sus fronteras operativas y revisen estructuras de datos internas para garantizar su validez. (Pressman, 2010, pág. 414)

1.6.5.3. Pruebas de Estrés. Las pruebas de estrés se diseñan para simular situaciones de carga del mundo real. Conforme aumenta el número de usuarios simultáneos de la aplicación o el número de transacciones en línea o la cantidad de datos descargados o subidos estas pruebas de rendimiento ayudarán a responder preguntas sobre el rendimiento de la aplicación. (Pressman, 2010, p. 473).

1.7. HERRAMIENTAS

1.7.1. Xampp

Es un acrónimo de X indicando la disponibilidad de distintos sistemas operativos, Apache, MariaDB, PHP y Perl. Es una distribución de Apache de software libre que actúa como un servidor web libre, fácil de usar y capaz de interpretar páginas dinámicas, siendo su principal objetivo la gestión de base de datos (MySQL y MariaDB), gestión del servidor Apache además de la interpretación del lenguaje de Script (PHP Y Perl). (Valencia Pavón , 2012-2013, pág. 75)

1.7.2. Laravel

Es un framework de código abierto del Modelo Vista Controlador (MVC) ofreciendo un conjunto de herramientas y recursos que tiene por finalidad de desarrollar aplicaciones y sistemas web con PHP de forma sencilla y ordenada, este framework fue desarrollado por Taylor Otwell y un grupo de desarrolladores contribuidores. (Arango Cordoba , 2016, pág. 17)

1.7.3. Mysql

Es un gestor de bases de datos que se encuentra dentro de la categoría de los programas open-source, está basado en el lenguaje de consultas estructurados (SQL) y a su

vez es un lenguaje de base de datos normalizados que puede crear consultas SQL de paso para manipular bases de datos remotas cliente servidor mediante los comandos, clausulas, operadores y funciones de agregado y mediante sus combinaciones realizar instrucciones de creación, actualización y manipulación de la base de datos. (Harb Hoecker, pág. 3)

1.7.4. Ajax

Es el asíncrono de JavaScript y XML que es un conjunto de técnicas de desarrollo web que permite generar aplicaciones web de manera interactiva que accede la comunicación entre el servidor web y el navegador del usuario, a su vez permite que el funcionamiento del sistema web sea veloz, asíncrona procesando las solicitudes del servidor en segundo plano. (Flores Fernandez, 2010, pág. 5)

1.7.5. JavaScript

Es un lenguaje de programación muy conocido. Entre otras funciones, gestiona el contenido dinámico de un sitio web y permite la interacción dinámica del usuario. XML es otra variante de un lenguaje de marcado como HTML, como lo sugiere su nombre: extensible Markup Language. Mientras HTML está diseñado para mostrar datos, XML está diseñado para contener y transportar datos.

Tanto JavaScript como XML funcionan de forma asíncrona en AJAX. Como resultado, cualquier aplicación web que use AJAX puede enviar y recuperar datos del servidor sin la necesidad de volver a cargar toda la página. (B., 2023)

1.8. LÍMITES Y ALCANCES

1.8.1. Límites

- El software solo estará disponible para departamentos con cobertura de los servicios de FENIX TELECOM S.R.L.

- No se subirá los formularios de contratos escaneados en archivo de empresa, de acuerdo a los requerimientos de la empresa.
- No estará enlazado con impuestos nacionales.

1.8.2. Alcances

La plataforma web, mejora el registro, control y seguimiento en los procesos administrativos que realiza la empresa FENIX TELECOM S.R.L. a continuación, se detalla en módulos las actividades:

- Implementación de plataforma web, de fácil acceso y una interfaz agradable al usuario.
- Copia de seguridad diario, de la base de datos que contiene la información de la empresa.
- Publicaciones actualizadas de paquetes, contactos, dirección y horarios de atención.
- Toma de decisiones en la empresa, a partir de los reportes estadísticos.

1.9. APORTES

El Sistema de Información Web de Control y Seguimiento para la Administración de Servicios de Internet Y Tv Cable aportará a la empresa FENIX TELECOM S.R.L. y a sus clientes de las siguientes maneras:

- Manejo eficiente de la administración de los servicios ofertados, por parte de la empresa.

- Centralización de información de los clientes, de forma organizada.
- Atención e información precisa a los clientes de la empresa, respecto a paquetes adquiridos y estados de cuenta.

CAPÍTULO II

2. MARCO TEÓRICO

2.1. INTRODUCCIÓN

En este capítulo se definirán los conceptos y definiciones como base para poder desarrollar el presente proyecto, con bases teóricas de diferentes fuentes donde se extrajo la información.

2.2. WEB 2.0

La web 2.0 es un concepto que empezó a utilizarse en el año 2003, su aparición se debió al auge de diversas aplicaciones como los blogs o las redes sociales que permitieron que los usuarios dejaran de ser mejor sujetos pasivos para adoptar un rol mucho más activo y dinámico. Sin embargo, a pesar de que hace años que existe este término.

En pocas palabras, el concepto de web 2.0 es sinónimo de aplicaciones web que permiten a los usuarios compartir información en Internet. En su momento, cuando este término se puso de moda, una de las plataformas preferidas era Blogger. Posteriormente, Facebook se convirtió en la primera red social que no solo permitía compartir información, sino la posibilidad de interactuar de una forma más inmediata. (Gardey, 2021)

2.2.1. Características de la Web 2.0

Es importante tener en cuenta que no existe una definición precisa de Web 2.0, aunque es posible aproximarse a ella estableciendo ciertos parámetros.

- Una página web que se limita a mostrar información y que ni siquiera se actualiza, forma parte de la generación 1.0. En cambio, cuando las páginas ofrecen un nivel considerable de interacción y se actualizan con los aportes de los usuarios, se habla de Web 2.0.

- Cabe mencionar que las diferencias entre la primera y la segunda era de la Web no se basan en un cambio a nivel tecnológico en los servidores, aunque naturalmente se ha dado un considerable avance en el hardware; es el enfoque de la Red, los objetivos y la forma en la que los usuarios comenzaron a percibir la información en línea lo que caracteriza este renacer, que tuvo lugar silenciosa pero velozmente, a comienzos del nuevo milenio.
- Algunas de las principales características de la web 2.0 son los servicios de redes sociales, las plataformas de auto publicación (por ejemplo, las herramientas de WordPress de creación de blogs y sitios web), las etiquetas (funcionalidad que permite añadir metadatos a páginas web o contenidos multimedia), el botón «me gusta» y la calificación de contenidos (que permite al usuario expresar su opinión sobre el contenido) y los marcadores sociales.

Hasta entonces, Internet era un universo de datos mayoritariamente estáticos, una fuente de consulta revolucionaria que atraía a millones de personas a contemplarla pasivamente. Si bien los foros y el chat datan de la Web 1.0, éstos se encontraban bien diferenciados de los sitios tradicionales (tal y como ocurre en la actualidad); navegar era similar a visitar un gran centro comercial, con infinidad de tiendas, en las cuales era posible comprar productos, pero no alterar sus vidrieras. (Wikipedia, 2021)

2.2.2. Usos de la Web 2.0

La Web 2.0 se caracteriza principalmente por la participación del usuario como contribuidor activo y no solo como espectador de los contenidos de la Web (usuario pasivo). (Wikipedia, 2021)

- Esto queda reflejado en aspectos como:

- El auge de los blogs.
- El auge de las redes sociales.
- Las webs creadas por los usuarios, usando plataformas de autoedición.
- El contenido agregado por los usuarios como valor clave de la Web.
- El etiquetado colectivo (folksonomía, marcadores sociales...).
- La importancia de la larga cola.
- La beta perpetua: la Web 2.0 se inventa permanentemente.
- Aplicaciones web dinámicas.
- La World Wide Web como plataforma.

2.3. METODOLOGÍA INGENIERÍA WEB BASADA EN UML (UWE)

La Ingeniería web basada en UML es una metodología basada en el Proceso Unificado y UML (Lenguaje Unificado de Modelado) para el desarrollo de aplicaciones Web, cubre todo el ciclo de vida de las aplicaciones Web es un lenguaje y una metodología adecuada para modelar el funcionamiento actual de la empresa. (SlideShare, 2015)

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

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

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

Las principales razones para utilizar los mecanismos de extensión de la UML en lugar de una técnica de modelado patentada es la aceptación de la UML en el desarrollo de sistemas de software, la flexibilidad para la definición de un lenguaje de modelado específico de dominio web: un llamado perfil UML y de un amplio soporte de modelado visual por parte de las todas las herramientas existentes de UML CASE. (UWE, 2016)

2.3.1. Características

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

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

los distintos objetos de la aplicación más bien conocida como clases de navegación, y el segundo modelo de estructura de navegación el cual se define la estructura de acceso que son utilizadas en la navegación, es decir todo aquello referente a menús, índices y demás.

2.3.2. Modelos de la metodología UWE

El método UWE consiste en la construcción de seis modelos de análisis y diseño. Dicha construcción se realiza dentro del marco de un proceso de diseño iterativo e incremental. Las actividades de modelado abarcan: el análisis de requerimientos, diseño conceptual, modelo de usuario, diseño de la navegación, de la presentación y diseño de la adaptación. (Thewolf, 2015)

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

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

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

En cuanto a los requisitos, UWE los clasifica dependiendo del carácter de cada uno. Además, distingue entre las fases de captura definición y validación de requisitos (Sanchez, 2017)

2.3.2.1. Modelo de análisis de requerimientos. Cumple un papel primordial en el proceso de producción de software, ya que se enfoca un área fundamental, su principal tarea consiste en la generación de especificaciones correctas que describan con claridad, de esta manera, se pretende minimizar los problemas relacionados por la mala gestión de requerimientos en el desarrollo de sistemas. Sirve como base sólida en el proceso de desarrollo de software, por lo que antes de pasar a aspectos referentes a la administración adecuada de los requerimientos, es importante definir los requerimientos y cuáles serán las características deseables que debería de tener (Chavez, 2012)

2.3.2.2. Requerimientos Funcionales. En general, los requerimientos funcionales describen acciones específicas que el ingeniero de software debe ser capaz de realizar durante el desarrollo de software. Los requerimientos funcionales a menudo se dividen en reglas de negocio y casos de uso. Las reglas de negocio son declaraciones de alto nivel que definen lo que un sistema debe hacer, mientras que los casos de uso son descripciones más detalladas de cómo debe funcionar el sistema. Algunos de los requerimientos más comunes en virtud de él incluyen:

- Las características y funcionalidad deseadas del producto
- Plataformas para desarrollar aplicaciones.
- Especificaciones de diseño en términos de tema, colores y fuentes
- Funcionalidad de backend: integración API y bases de datos.

- Plazos de finalización.

2.3.2.3. Requerimientos no funcionales. Los requisitos no funcionales especifican las características que el software debe tener a lo largo del desarrollo de la aplicación. Estos requisitos suelen clasificarse en tres categorías principales: rendimiento, seguridad y calidad. Los aspectos de rendimiento abarcan la eficiencia y la capacidad de respuesta del sistema. Los aspectos de seguridad garantizan la protección de los datos y el acceso seguro al software. Por último, los aspectos de calidad se enfocan en la mantenibilidad, escalabilidad y fiabilidad del sistema, asegurando que el software cumpla con los estándares y expectativas establecidos.

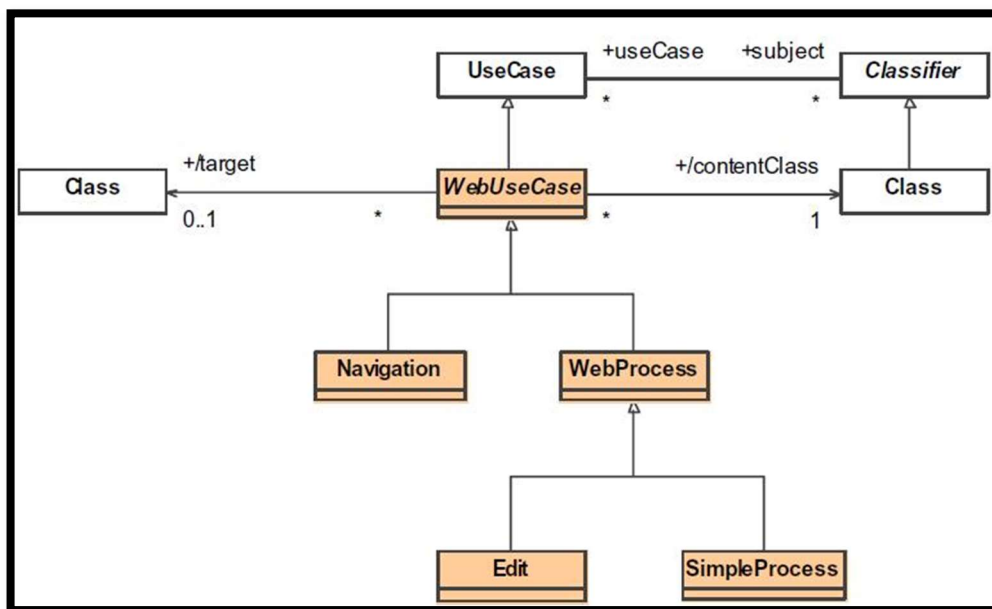
- **Requerimientos de rendimiento.** Los requerimientos de rendimiento suelen dividirse en dos categorías: tiempo de respuesta y rendimiento. El tiempo de respuesta es el tiempo que tarda un sistema en responder a la solicitud de un usuario, mientras que el rendimiento es el número de solicitudes que un sistema puede manejar. Son más críticos para los sistemas interactivos, como las aplicaciones de escritorio y los sitios web, donde los usuarios esperan respuestas inmediatas a sus acciones.
- **Requerimientos de seguridad.** Los requerimientos de seguridad especifican las medidas que un sistema debe tomar para proteger los datos del acceso no autorizado. En algunos casos, los requerimientos de seguridad también pueden especificar el nivel de protección requerido, como confidencial o de alto secreto. Implica autenticación, autorización y cifrado.
- **Requerimientos de calidad.** Especifica el nivel de calidad que debe cumplir un sistema. En algunos casos, los requerimientos de calidad también pueden especificar los métodos utilizados para medir la calidad, como la densidad de

defectos o la satisfacción del cliente. Los requerimientos de calidad son generalmente cuatro medidas de calidad: conformidad, usabilidad, confiabilidad y mantenibilidad.

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

Figura 2

Modelo de análisis de requerimientos.



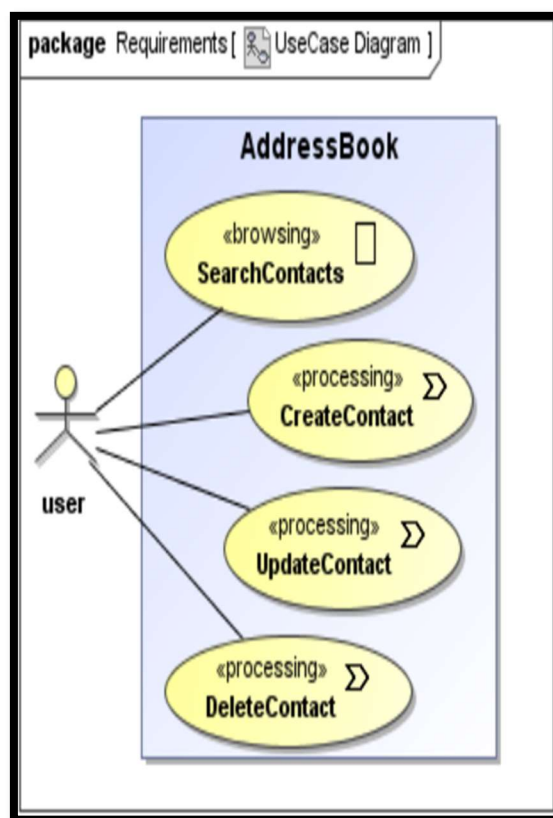
Nota. Figura muestra modelo de análisis. (Vargas, 2017)

2.3.2.4. Modelo de Casos de Uso. Un diagrama de casos de uso ilustra cómo los actores interactúan con el sistema y qué funcionalidades ofrece este en términos de su interacción externa.

Este tipo de diagrama presenta de manera visual la relación entre los diferentes usuarios (actores) y los diversos casos de uso, que son las distintas funciones o servicios que el sistema proporciona, el diagrama de casos de uso sirve para entender y comunicar cómo el sistema se comporta desde la perspectiva de quienes lo utilizan, destacando las conexiones y flujos de actividad que ocurren entre los usuarios y el sistema.

Figura 3

Modelo de Caso de Uso de la Ingeniería Web.



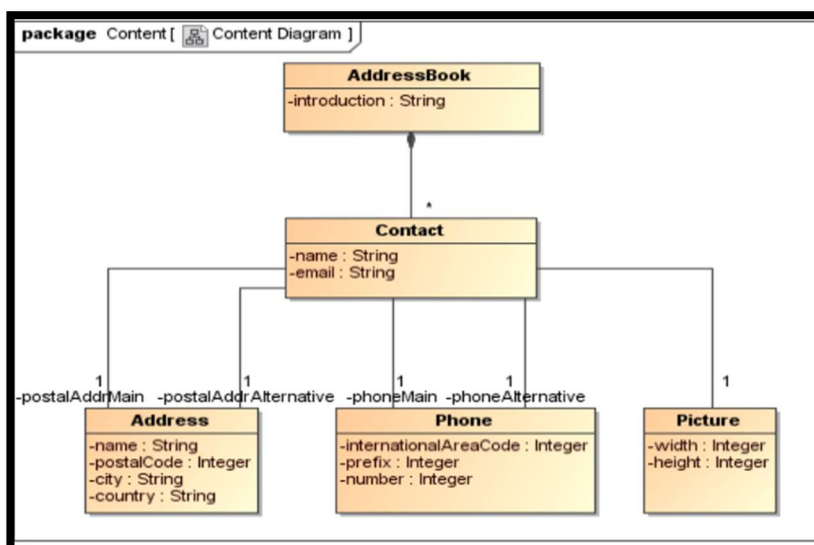
Nota. Figura muestra el formato del modelo. (Vargas, 2017)

2.3.2.5. Modelo contenido. El modelo conceptual de un sistema de software es una representación abstracta externa que utiliza diagramas y notaciones, con diversos grados de formalidad, para describir la información esencial que una persona debe conocer sobre el sistema. Esta información, que abarca la estructura y el comportamiento del sistema, se almacena en la memoria a largo plazo del individuo.

El propósito de este modelo es proporcionar una comprensión clara y organizada del sistema, facilitando su aprendizaje, uso y mantenimiento.

Figura 4

Modelo Contenido de la Ingeniería Web.



Nota. La figura muestra el formato del modelo de contenido para su elaboración. (Vargas, 2017)

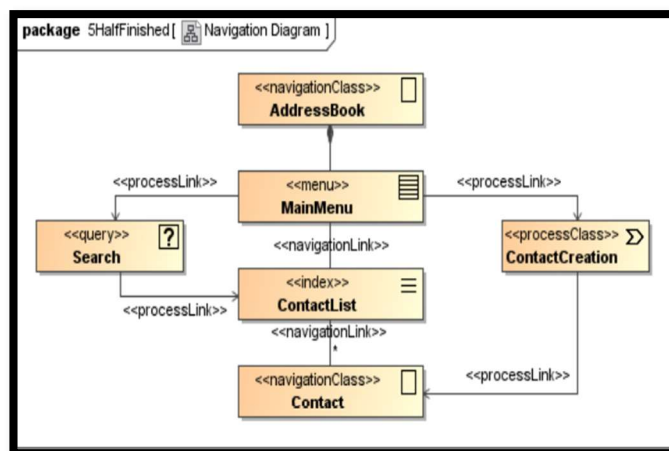
2.3.2.6. Modelo de Navegación. El modelo de estructura de navegación define la estructura de nodos y links de una WebApp mostrando cómo se puede realizar la navegación utilizando elementos de acceso tales como índices, visitas guiadas, consultas y menús.

Los elementos de modelado son:

- a. **Clases de navegación**, que se denotan con (0), representan los nodos navegables de la estructura de hipertexto.
- b. **Links de navegación**, que muestran el vínculo directo entre las clases de navegación.
- c. **Caminos de navegación alternativos**, los cuales son visualizados con el estereotipo <<menu>> ().
- d. **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>>).
- e. **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.
- f. **Links de procesos**, que representan el vínculo entre las clases de proceso y de navegación.

Figura 5

Modelo Navegación de la Ingeniería Web.



Nota. La figura muestra el modelo navegación y su formato de desarrollo. (Vargas, 2017)

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.

2.3.2.7. 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.

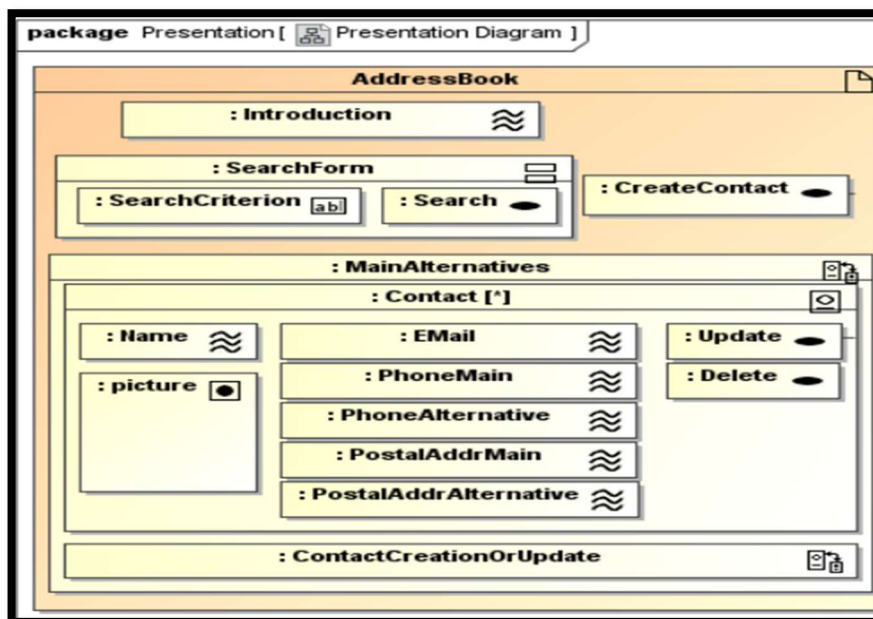
Clases de presentación, las cuales se basan directamente en los nodos del modelo de navegación. Una clase de presentación () está compuesta por elementos de UI tales como, texto (<<text>>), vinculo (<<anchor>>), botón (<<button>>), imagen (<<image>>), formulario (<<form>>), y colección de vínculos (<<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 6

Modelo de Presentación de la Ingeniería Web.



Nota. La figura muestra el modelo presentación. (Vargas, 2017)

2.4. ARQUITECTURA DE SOFTWARE (MVC)

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

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

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

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

2.4.1. Modelo

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

El modelo es el responsable de:

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

2.4.2. Vista

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

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

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

Las vistas son responsables de:

- Recibir datos del modelo y los muestra al usuario.
- Tienen un registro de su controlador asociado (normalmente porque además lo instancia).
- Pueden dar el servicio de "Actualización ()", para que sea invocado por el controlador o por el modelo (cuando es un modelo activo que informa de los cambios en los datos producidos por otros agentes).

2.4.3. Controlador

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

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

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

El controlador es responsable de:

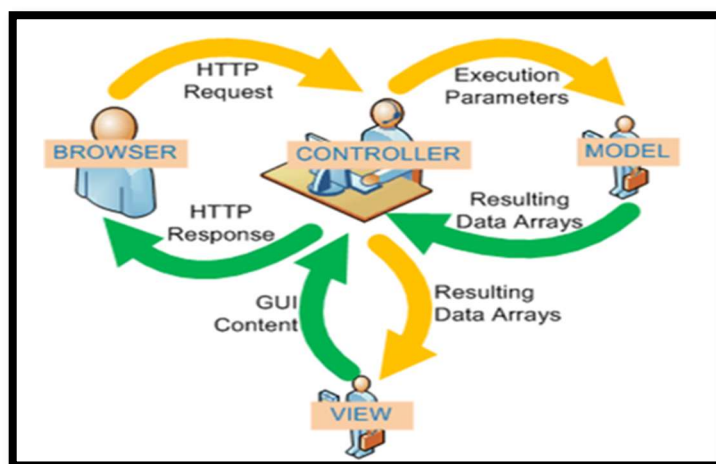
- Recibe los eventos de entrada (un clic, un cambio en un campo de texto, etc.).
- Contiene reglas de gestión de eventos.

2.4.4. Características del Modelo Vista Controlador

Actualmente en los diferentes lenguajes de programación, se ha establecido un orden al momento de desarrollar un sistema, ese orden del modelo vista controlador, que muchos frameworks lo han usado en su estructura al momento de descargarse sus complementos. Aunque se pueden encontrar diferentes implementaciones de MVC, la característica del flujo que sigue el control generalmente es el siguiente:

Figura 7

Flujo de MVC



Nota. La figura muestra el movimiento de flujo del modelo, vista y controlador con el cual se interactúa.

(Vargas, 2017)

- El usuario interactúa con la interfaz de usuario de alguna forma (por ejemplo, el usuario pulsa un botón, enlace, etc.)

- El controlador recibe (por parte de los objetos de la interfaz-vista) la notificación de la acción solicitada por el usuario. El controlador gestiona el evento que llega, frecuentemente a través de un gestor de eventos (handler) o callback.
- El controlador accede al modelo, actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada por el. Los controladores complejos están a menudo estructurados usando un patrón de comando que encapsula las acciones y simplifica su extensión.
- El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se refleja los cambios en el modelo. El modelo no debe tener conocimiento directo sobre la vista. Sin embargo, el patrón de observador puede ser utilizado para proveer cierta Unidad entre el modelo y la vista, permitiendo al modelo notificar a los interesados de cualquier cambio.
- La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente. (Alicante, 2021)

2.5. HERRAMIENTAS

2.5.1. *MagicDraw*

Es una herramienta visual de modelado UML y CASE con soporte de trabajo en equipo. Diseñada para analistas de negocio, analistas de software, analistas de sistemas, programadores, ingenieros de control de calidad y redactores de documentación, esta herramienta de desarrollo dinámica y versátil facilita el análisis y el diseño de sistemas orientados a objetos (OO) y bases de datos. Proporciona el mejor mecanismo de ingeniería de código de la industria (con soporte completo de ida y vuelta para los lenguajes de

programación J2EE, C#, C++, CORBA IDL, .NET, XML Schema, WSDL), así como el modelado de esquemas de bases de datos, la generación de DDL y las facilidades de ingeniería inversa.

MagicDraw, herramienta CASE desarrollada por No Magic. Es compatible con el estándar UML 2.3, desarrollo de código para diversos lenguajes de programación (Java, C++ y C#, entre otros) así como para modelar datos. Cuenta con capacidad para trabajar en equipo y es compatible con varios entornos de desarrollo (IDEs) Diseñada para los analistas del negocio, los analistas del software, los programadores, los ingenieros y los escritores de la documentación. (Magicdraw, 2021)

- Un práctico editor UML. Una interfaz gráfica de usuario amigable y personalizable le permite expresar sus ideas en UML de la forma más rápida y directa posible. Toda la notación UML 2 y la semántica son compatibles.
- Una poderosa herramienta de ingeniería de código. Puede construir un modelo UML, generar código fuente a partir de él, escribir más código a mano o en su IDE favorito, luego revertir el código, hacer algunos cambios en el MagicDraw, y fusionar su modelo con el código de nuevo. No se pierde ningún detalle, no importa el camino (hacia adelante o hacia atrás) que usted vaya. Soporte para Java, C#, C++, CORBA IDL, DDL.
- Una facilidad de informes del modelo UML. Basado en los archivos XSL, puede generar informes HTML personalizados para cada elemento del modelo. MagicDraw le permite seleccionar qué partes del modelo incluir y cómo debe ser el informe.
- Una herramienta de análisis de modelos OO. MagicDraw genera diagramas de dependencia de paquetes, recupera y muestra árboles de herencia, o rastrea

cualquier otro tipo de relación como un parámetro de método o un valor de retorno, tales como dependencias, asociaciones, realizaciones y uso de clases.

- Un generador de modelos OO personalizado. Conjunto predefinido de patrones de diseño incluyendo GoF, Java, patrón de diseño JUnit. Todos los patrones pueden ser extendidos y nuevos patrones pueden ser creados usando Java o Jython.
- Herramienta de modelado en equipo. Más de un desarrollador puede trabajar con el mismo modelo OO. El modelo UML se almacena en el repositorio de Teamwork Server. Cada desarrollador puede bloquear una parte del modelo y trabajar en esa parte individualmente. Más tarde, los cambios pueden confirmarse en el servidor y compartirse con el equipo. El servidor funciona como un sistema de control de versiones de código fuente.
- Herramienta de modelado de bases de datos. Puede diseñar el esquema de la base de datos en un diagrama de clases UML, y luego generar su código DDL. Si desea analizar la estructura de la base de datos existente, puede realizar ingeniería inversa a través del puente JDBC.

2.5.2. Gestor de base de datos MySql

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

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

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

MySQL puede ejecutarse en múltiples plataformas como Linux, Windows, Unix y un esquema de información para definir y administrar sus metadatos. Puede instalarlo en su sistema local o incluso en el servidor.

a) Características clave de MySQL

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

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

- **Facilidad de gestión:** Es bastante fácil de descargar y utilizar el software.
- **Alto rendimiento:** Le proporciona utilidades de carga rápida con diferente caché de memoria.

- Escalable: Con MySQL, puede escalar en cualquier momento. Es realmente fácil crear almacenes de datos que incluyen una enorme cantidad de datos.
- Compatibilidad: MySQL es compatible con todas las plataformas modernas como Windows, Linux, Unix.
- Rendimiento: MySQL le ofrece resultados de alto rendimiento sin perder la funcionalidad esencial.
- Seguridad completa de los datos: Sólo los usuarios autorizados pueden acceder a la base de datos. Completa seguridad de los datos.
- Bajo Costo: Es gratis de usar.
- Eficiencia de memoria: MySQL tiene baja pérdida de memoria.
- Arquitectura Cliente y Servidor: MySQL basa su funcionamiento en un modelo cliente y servidor. Es decir, clientes y servidores se comunican entre sí de manera diferenciada para un mejor rendimiento. Cada cliente puede hacer consultas a través del sistema de registro para obtener datos, modificarlos, guardar estos cambios o establecer nuevas tablas de registros, por ejemplo.
- Compatibilidad con SQL: SQL es un lenguaje generalizado dentro de la industria. Al ser un estándar MySQL ofrece plena compatibilidad por lo que si has trabajado en otro motor de bases de datos no tendrás problemas en migrar a MySQL.
- Vistas: Desde la versión 5.0 de MySQL se ofrece compatibilidad para poder configurar vistas personalizadas del mismo modo que podemos hacerlo en otras bases de datos SQL. En bases de datos de gran tamaño las vistas se hacen un recurso imprescindible.

- Procedimientos almacenados. MySQL posee la característica de no procesar las tablas directamente, sino que a través de procedimientos almacenados es posible incrementar la eficacia de nuestra implementación.
- Desencadenantes. MySQL permite además poder automatizar ciertas tareas dentro de nuestra base de datos. En el momento que se produce un evento otro es lanzado para actualizar registros o optimizar su funcionalidad.
- Transacciones. Una transacción representa la actuación de diversas operaciones en la base de datos como un dispositivo. El sistema de base de registros avala que todos los procedimientos se establezcan correctamente o ninguna de ellas. En caso por ejemplo de una falla de energía, cuando el monitor falla u ocurre algún otro inconveniente, el sistema opta por preservar la integridad de la base de datos resguardando la información.

b) Ventajas de usar MySQL

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

2.5.3. Lenguaje de Programación PHP

PHP es un lenguaje de secuencia de comandos de servidor diseñado específicamente para la Web. Dentro de una página Web puede incrustar código PHP que se ejecutara cada

vez que se visite una página. El código PHP es interpretado en el servidor Web y genera código HTML y otro contenido que el visitante verá.

PHP fue concebido en 1994 y es fruto del trabajo de un hombre, Rasmus Lerdorf. Ha sido adoptado por otras personas de talento y ha experimentado tres transformaciones importantes hasta convertirse en el producto actual. En octubre de 2002, era utilizado por más de nueve millones de dominios de todo el mundo y su número crece rápidamente. Si desea conocer el número actual de sitios que utilizan este lenguaje, visite el sitio <http://www.php.net/usage.php>. PHP es un producto de código abierto, lo que quiere decir que puede acceder a su código. Puede utilizarlo, modificarlo y redistribuirlo sin coste alguno.

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

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

El lenguaje PHP es un lenguaje de programación de estilo clásico, es decir, es un lenguaje de programación con variables, sentencias condicionales, ciclos (bucles),

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

2.5.4. Bootstrap

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

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

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

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

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

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

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

2.5.5. JavaScript

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

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

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

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

2.5.6. *jQuery*

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

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

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

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

A continuación, se presentan las principales características de JQuery:

- Es un software libre, por lo que puede ser empleado por cualquier usuario de manera gratuita.

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

2.6. MÉTRICAS DE CALIDAD DE SOFTWARE

Para asegurar que el proyecto se desarrolle de aplicaron una serie de métricas de calidad seguridad, donde se calcularan los parámetros de calidad que cumple el sistema. En la mayoría de los desafíos técnicos, las métricas nos ayudan a entender tanto el proceso técnico que se utiliza para desarrollar un producto, como el propio producto.

El proceso para intentar mejorarlo, el producto se mide para intentar aumentar su calidad.

El principio, podría parecer que la necesidad de la medición es algo evidente después de todo es lo que nos permite cuantificar y por consiguiente gestionar de forma más efectiva.

La medición es muy común en el mundo de la ingeniería. Medimos potencia de consumo, pesos, dimensiones físicas, temperaturas, voltajes, señales de ruidos por mencionar algunos aspectos. Desgraciadamente la medición se aleja de lo común en el mundo de la ingeniería del software. Encontramos dificultades en ponernos de acuerdo sobre que medir y como va evaluar las medidas.

- Para indicar la calidad del producto.
- Para evaluar la productividad de la gente que desarrolla el producto.
- Para evaluar los beneficios en términos de productividad y de calidad, derivados de uso de nuevos métodos y herramientas de la ingeniería de software.
- Para establecer una línea de base para la estimación
- Para ayudar a justificar el uso de nuevas herramientas o de formación adicional.

2.6.1. Calidad de Software

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

- Seguridad. Debido a iniciativas como cero papeles, mejoramiento ambiental, gestión documental, y la tendencia actual hacia la digitalización de la información de las empresas y la sociedad en general, y debido a que prácticamente todos los dispositivos electrónicos tienen conectividad, se vuelve sensible la información confidencial y protegida por marcos legales. Es por esto que este atributo toma importancia, y se hace imperante mejorar la seguridad y sus atributos de calidad, como la integridad, la privacidad y la confidencialidad, incrementado el nivel de la complejidad y la efectividad de los mecanismos software que prevengan las diferentes modalidades de ataque.

- Disponibilidad y Confiabilidad. La dependencia social, organizacional y de gobierno de los sistemas de información, exige el aseguramiento de una funcionalidad ininterrumpida de todas sus funciones, y cada vez son menos tolerables las interrupciones o fallas.
- Usabilidad. El acceso cada vez mayor a todo tipo de usuario hace que el desarrollo de software sea más exigente en el diseño de interfaz y factores relacionados con la usabilidad, y cada vez se va ir agudizando más este problema en el análisis y diseño, lo cual incrementará los costos.

2.6.2. Norma ISO/IEC 25000

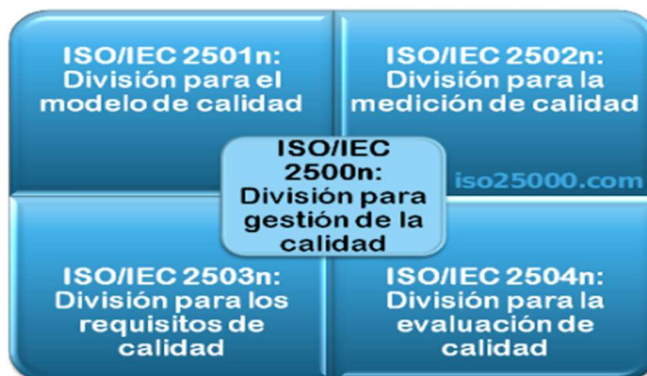
La calidad del producto, junto con la calidad del proceso, es uno de los aspectos más importantes actualmente en el desarrollo de Software. Relacionada con la calidad del producto, recientemente ha aparecido la familia de normas ISO/IEC 25000, que proporciona una guía para el uso de la nueva serie de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

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

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

Figura 8

División de calidad de Software.



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

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

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

- **Vista interna:** Se ocupa de las propiedades del sistema como: el tamaño, la complejidad o la conformidad con las normas de orientación a objetos.
- Puede utilizarse desde las primeras fases del desarrollo, permitiendo detectar deficiencias en el sistema en etapas tempranas del ciclo de vida.
- **Vista externa:** Analiza el comportamiento del sistema en producción y estudia sus atributos, por ejemplo: el rendimiento de un software en una máquina determinada, el uso de memoria de un programa o el tiempo de funcionamiento entre fallos.

- Esta vista necesita que el sistema esté completo y en la fase de producción del producto, siendo muy dependiente de la máquina donde se ejecute.
- **Vista en uso:** Mide la productividad y efectividad del usuario final al utilizar el sistema.

2.6.3. Clasificación de las características y sub características

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

2.6.3.1. Adecuación funcional. La funcionalidad en el modelo de calidad del software ISO/IEC 25000 se refiere a la capacidad del software para proporcionar funciones que satisfacen necesidades explícitas e implícitas cuando se usa bajo condiciones específicas.

La funcionalidad se desglosa en varias subcaracterísticas, cada una evaluando un aspecto particular de esta capacidad:

Para calcular los puntos de función, se consideran cinco características principales del dominio de información:

Número de entradas de usuarios se contabiliza cada entrada de usuario que proporciona datos al sistema, las entradas pueden incluir formularios de registro, comandos de usuario o cualquier otra forma de entrada de información que el sistema deba procesar.

Número de salidas de usuario, este atributo se refiere a cada salida que el sistema proporciona al usuario tanto las salidas pueden incluir informes, reportes, mensajes de advertencia, notificaciones y mensajes de error, cada uno de estos elementos es crucial para la interacción del usuario con el sistema y para asegurar que el usuario reciba la información necesaria de manera oportuna y precisa.

Número de consultas, se refiere a cada solicitud de información que el sistema procesa y responde. Las consultas pueden ser búsquedas en bases de datos, solicitudes de informes o cualquier otra interacción que requiera que el sistema recupere y presente información específica al usuario.

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

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

2.6.3.4. Funcionalidad. Capacidad del producto software para ser entendido, aprendido, usado y resultar atractivo para el usuario, cuando se usa bajo determinadas condiciones.

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

Tabla 1.

Tabla de entradas de usuario

ENTRADAS DE USUARIO	
Administración de usuarios	X
Administración de técnicos	X
Administración de usuarios	X
TOTAL	X

Nota. Describe las entradas de usuario

Además de ser muy importante la cantidad de entradas del usuario, las salidas también se deben tomar en cuenta, en la obtención del porcentaje de usabilidad.

Tabla 2.*Tabla de salidas de usuario*

SALIDAS DE USUARIO	
Administración de usuarios	X
Administración de técnicos	X
Administración de usuarios	X
TOTAL	X

Nota. Describe las salidas del usuario

La importancia de las variables de entrada y salida de usuarios, de acuerdo a los parámetros de la métrica de calidad, enfocado a la usabilidad, también tomo importancia en los que son las peticiones y respuestas del sistema de acuerdo a la siguiente tabla:

Tabla 3.*Tabla de peticiones de usuario*

PETICIONES DE USUARIO	
Administración de usuarios	3
Administración de técnicos	3
Administración de usuarios	8
TOTAL	14

Nota. Tabla que muestra la cantidad de peticiones de usuario.

Fórmula para ajustar puntos de función, que considera factores como la complejidad técnica, el rendimiento y la usabilidad.

(1) *Fórmula de ajuste*

$$PF = \text{conteo total} [0.65 + 0.01 * \sum fi] \quad (1)$$

Donde:

$\sum (fi)$: Sumatoria de los valores de los factores de ajuste.

De esta forma sacamos los resultados de usabilidad.

2.6.3.5. Confiabilidad. La confiabilidad de un sistema se refiere a su capacidad para mantener un nivel de funcionamiento adecuado durante un período de prueba establecido que implica considerar aspectos como la madurez del sistema, su capacidad de recuperación frente a fallos, su tolerancia a posibles errores y su cumplimiento de los estándares de fiabilidad, para evaluar la confiabilidad, se observa el rendimiento del sistema desde el momento en que comienza a funcionar hasta un punto determinado en el tiempo, registrando cualquier incidencia o cambio en su desempeño.

Considerando que se tiene en cuenta que:

(2) *Probabilidad de fallas*

$$P(T \leq t) \Rightarrow \text{Probabilidad de fallas} \quad (2)$$

(3) *Probabilidad de trabajo sin fallas*

$$P(T > t) = 1 - F(t) \Rightarrow \text{Probabilidad de trabajo sin fallas} \quad (3)$$

Se considera la siguiente función al calcular la confiabilidad del sistema.

(4) *Función de la confiabilidad del sistema*

$$R(t) = f * e^{-\mu * t} \quad (4)$$

Donde:

f = Funcionalidad del sistema
 μ = Probabilidad de error del sistema

t = Tiempo de prueba del sistema

2.6.3.6. Usabilidad. Al evaluar la usabilidad de un sistema, es importante considerar el factor humano, asegurando que el software cumpla con los requisitos y expectativas del usuario, por lo tanto, se lleva a cabo una evaluación mediante encuestas dirigidas a los usuarios del sistema.

Estas encuestas permiten recopilar información valiosa sobre la experiencia del usuario, su satisfacción y la facilidad de uso percibida que se utiliza luego para calcular la usabilidad del sistema, aplicando una fórmula diseñada específicamente para medir la eficacia y la experiencia del usuario en la interacción con el software.

Función para determinar la usabilidad

$$FU = \left[\frac{\sum x_i}{n} * 100 \right] \quad (5)$$

Donde:

x_i = Valores de evaluación de cada pregunta

n = Número de preguntas

Tabla 4.

Índices

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

2.6.3.7. **Mantenibilidad.**

Esta métrica evalúa la cantidad de trabajo requerido para realizar cambios en el sistema, ya sea para corregir errores o agregar nuevas funcionalidades. El estándar IEEE94 recomienda utilizar el índice de madurez del sistema como indicador de su estabilidad. Por lo tanto, la ecuación para este índice es la siguiente:

Ecuación para el índice de madurez del sistema

$$IMS = \frac{[Mt - (Fa + Fc + Fd)]}{Mt} \quad (7)$$

Donde:

- Mt = Número de módulos total de la versión actual
- Fa = Número de módulos de la versión actual que se añadieron.
- Fc = Número de módulos de la versión actual que se cambiaron.
- Fd = Número de módulos de la versión anterior que se eliminaron en la versión actual.

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

$$Portabilidad = 1 - \left(\frac{\text{número de día para aportar el sistema}}{\text{número de días para implementar el sistema}} \right)$$

2.7. ANÁLISIS DE COSTOS

2.7.1. *Cocomo II*

Modelo COCOMO II, modelo de estimación que se encuentra en la jerarquía de modelos de estimación de software con el nombre de COCOMO, por Constructive Cost Model

(Modelo Constructivo de Coste). El modelo COCOMO original se ha convertido en uno de los modelos de estimación de coste del software más utilizados y estudiados en la industria. (COCOMO, 2013)

2.7.2. Características

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

2.7.3. Ecuaciones del modelo Cocomo

Las ecuaciones que se utilizan en los tres modelos son:

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

$$T = c(E)^d, \text{ en meses.}$$

$$NP = \frac{E}{T}, \text{ en personas.}$$

Dónde:

- E = Es el esfuerzo requerido por el proyecto, en persona – mes.
- T_{dev} = Es el esfuerzo requerido por el proyecto, en meses.
- P = Es el numero de personas requerido por el proyecto.
- a, b, c y d , Son constantes con valores definidos en una tabla, segun cada submodelo

- **KI** = Es la cantidad de líneas de código, en miles.
- **m(X)** = Es un multiplicador que depende de 15 atributos.
- A la vez, cada SubModelo también se divide en modos que representan el tipo de proyecto, y puede ser:

Modo Orgánico: Un pequeño grupo de programadores experimentados desarrollan software en un entorno familiar. El tamaño del software varía desde unas pocas líneas.

Modo Semilibre o Semiencajado: Corresponde a un esquema intermedio entre el orgánico y el rígido; el grupo de desarrollo puede incluir una mezcla de personas experimentadas y no experimentadas.

Modo Rígido o Empotrado: El proyecto tiene fuertes restricciones, que pueden estar relacionadas con la funcionalidad y/o pueden ser técnicas. El problema a resolver es único y es difícil basarse en la experiencia.

2.7.4. Modelos de COCOMO II

2.7.4.1. Modelo Básico. Se utiliza para obtener una primera aproximación rápida del esfuerzo, y hace uso de la siguiente tabla de constantes para calcular distintos aspectos de costes.

Tabla 5

Coeficientes del modelo Básico.

MODO	A	B	C	D
Orgánico	2.40	1.05	2.5	0.38
Semi acoplado	3.00	1.12	2.5	0.35
Empotrado	3.60	1.20	2.5	0.32

Nota. La siguiente tabla muestra los coeficientes multiplicadores del modelo básico.

Se puede observar que a medida que aumenta la complejidad del proyecto (modo), las constantes aumentan de 2.4 a 3.6, que corresponde a un incremento del esfuerzo del personal. Hay que utilizar con mucho cuidado el modelo básico puesto que se obvian muchas características del entorno

2.7.4.2. Modelo Intermedio. En este modelo se introducen 15 atributos de coste para tener en cuenta en el entorno de trabajo, incrementando así la precisión de la estimación.

Para este ajuste, al resultado de la fórmula general se lo multiplica por el coeficiente surgido de aplicar los atributos que se decidan utilizar.

De acuerdo a la cantidad de líneas de código, bajo ese parámetro podemos establecer las relaciones en cuanto a los valores constantes a reemplazar en las tablas correspondientes.

Los valores de las constantes a reemplazar en la fórmula se muestran en la siguiente tabla:

Tabla 6

Coefficientes del modelo intermedio.

PROYECTO SOFTWARE	A	B	C	D
Orgánico	3.2	1.05	2.5	0.38
Semi acoplado	3.0	1.12	2.5	0.35
Empotrado	2.8	1.20	2.5	0.32

Nota. La siguiente tabla muestra los coeficientes multiplicadores del modelo intermedio.

Se puede observar que los exponentes son los mismos que los del modelo básico, confirmando el papel que representa el tamaño; mientras que los coeficientes de los modos orgánico y rígido han cambiado, para mantener el equilibrio al rededor del SemiLibre con respecto al efecto multiplicador de los atributos de coste.

2.7.4.3. Atributos de Coste

Cada atributo se cuantifica para un entorno de proyecto. La escala es muy baja bajo - nominal - alto - muy alto - extremadamente alto. Dependiendo de la calificación de cada atributo, se asigna un valor para usar de multiplicador en la fórmula el significado de los atributos es el siguiente, según su tipo de:

Tabla 7

Atributos de costes totales.

ATRIBUTOS	VALOR					
	MUY BAJO	BAJO	NOMINAL	ALTO	MUY ALTO	EXTRA ALTO
ATRIBUTOS DE SOFTWARE						
Fiabilidad	0,75	0,88	1,00	1,15	1,40	
Tamaño de Base de datos		0,94	1,00	1,08	1,16	
Complejidad	0,70	0,85	1,00	1,15	1,30	1,65
ATRIBUTOS DE HARDWARE						
Restricciones de tiempo de ejecución			1,00	1,11	1,30	1,66
Restricciones de memoria virtual			1,00	1,06	1,21	1,56
Volatilidad de la máquina virtual		0,87	1,00	1,15	1,30	
Tiempo de respuesta		0,87	1,00	1,07	1,15	
ATRIBUTOS DE PERSONAL						
Capacidad de análisis	1,46	1,19	1,00	0,86	0,71	
Experiencia en la aplicación	1,29	1,13	1,00	0,91	0,82	
Calidad de los programadores	1,42	1,17	1,00	0,86	0,70	
Experiencia en la máquina virtual	1,21	1,10	1,00	0,90		
Experiencia en el lenguaje	1,14	1,07	1,00	0,95		

ATRIBUTOS DEL PROYECTO								
Técnicas actualizadas de programación			1,24	1,10	1,00	0,91	0,82	
Utilización de herramientas de software			1,24	1,10	1,00	0,91	0,83	
Restricciones de tiempo de desarrollo			1,22	1,08	1,00	1,04	1,10	

Nota. La siguiente tabla muestra los coeficientes multiplicadores de los atributos totales.

2.8. SEGURIDAD

2.8.1. SGSI (*Sistema de Gestión de la Seguridad de la Información*)

Toda la información almacenada y procesada por una organización está expuesta ante amenazas de ataque (por intereses comerciales, intelectuales y/o chantaje y extorsión), error (intencionado o por negligencia), ambientales (por ej. Inundación o incendio), fallo en los sistemas (de almacenamiento de datos, informáticos, redes telemáticas), entre otras y también está sujeta a vulnerabilidades que representan puntos débiles inherentes a su propio uso en el ciclo de vida representado a continuación. Permitir que una información precisa y completa esté disponible de manera oportuna para aquellos autorizados que tienen una necesidad es un catalizador para la eficiencia del negocio. Para poder interrelacionar y coordinar las actividades de protección para la seguridad de la información, cada organización necesita establecer su propia política y objetivos para la seguridad de la información dentro de la coherencia del marco de globales de la organización. Una vez fijados los objetivos en seguridad de la información necesitamos asegurar el modo de poder lograrlos eficazmente, en definitiva, un sistema de gestión de la seguridad de la información o SGSI en su forma abreviada.

Por tanto, un SGSI consiste en el conjunto de políticas, procedimientos y directrices junto a los recursos y actividades asociados que son administrados colectivamente por una

organización, en la búsqueda de proteger sus activos de información esenciales. Un SGSI desde la visión del estándar internacional ISO/IEC 27001 es un enfoque sistemático para establecer, implementar, operar, monitorear, revisar, mantener y mejorar la seguridad de la información de una organización y lograr sus objetivos comerciales y/o de servicio (p.ej. en empresas públicas, organizaciones sin ánimo de lucro, etc.).

El alcance de un SGSI puede incluir, en función de dónde se identifiquen y ubiquen los activos de información esenciales, total o sólo una parte de la organización, funciones específicas e identificadas de la organización, secciones específicas e identificadas de la organización, o una o más funciones en un grupo de organizaciones. El término seguridad de la información generalmente se basa en que la información se considera un activo que tiene un valor que requiere protección adecuada, por ejemplo, contra la pérdida de disponibilidad, confidencialidad e integridad.

- **Confidencialidad:** La información no se pone a disposición ni se revela a individuos, entidades o procesos no autorizados.
- **Integridad:** Mantenimiento de la exactitud y completitud de la información y sus métodos de proceso.
- **Disponibilidad:** Acceso y utilización de la información y los sistemas de tratamiento de la misma por parte de los individuos, entidades o procesos autorizados cuando lo requieran.

2.8.2. Aspectos de seguridad

La seguridad de las instalaciones de cómputo, almacén de datos y la información generada es parte de una conversión exitosa, el reconocimiento de la necesidad de seguridad es una consecuencia natural de la creencia de que la información es un recurso organizacional importante. Con las transacciones cada vez más complejas y muchos intercambios

innovadores, la Web ha producido un incremento en las preocupaciones de seguridad para el mundo profesional.

Es útil pensar en la seguridad de sistemas, datos e información en un continuo imaginario que va desde seguridad total hasta acceso abierto por completo. Aunque no hay tal cosa como un sistema totalmente seguro, las acciones de los analistas y usuarios pretenden mover los sistemas hacia el lado más seguro del espectro, disminuyendo la vulnerabilidad del sistema. Se debe observar que conforme más personas en la organización obtienen mayor potencia de cómputo, obtienen acceso a la Web, o se conectan a las intranets y extranets, la seguridad se vuelve cada vez más difícil y compleja.

La seguridad tiene tres aspectos interrelacionados: físico, lógico y conductual. Los tres deben trabajar juntos si deseamos que la calidad de seguridad permanezca alta. Seguridad física La seguridad física se refiere a proteger el sitio donde se encuentra la computadora, su equipo y software a través de medios físicos. Puede incluir acceso controlado a las salas de cómputo por medio de signos legibles por la máquina, sistemas biométricos o un registro de entrada y salida del sistema por un humano, así como el uso de cámaras de televisión de circuito cerrado para supervisar las áreas de cómputo, respaldando con frecuencia los datos y almacenando los respaldos en un área a prueba de fuego o de agua, a menudo en una ubicación remota segura, el equipo de cómputo pequeño se debe asegurar para que un usuario típico no pueda moverlo y se debe garantizar el suministro ininterrumpido de energía eléctrica. Las alarmas que notifican a las personas apropiadas en caso de fuego, inundación o intrusión no autorizada de una persona deben estar en todo momento en funcionamiento activo.

2.8.2.1. Seguridad lógica. La seguridad lógica se refiere a los controles lógicos en el software. Los controles lógicos son conocidos por la mayoría de los usuarios como contraseñas o códigos de autorización de alguna clase. Cuando se usan, permiten al usuario entrar al sistema o a una parte específica de una base de datos con una contraseña correcta. Sin embargo, las contraseñas se manejan de manera descuidada en muchas organizaciones. Los empleados han escuchado por casualidad gritar una contraseña en las oficinas atestadas, grabar las contraseñas para sus pantallas y compartir las contraseñas personales con empleados autorizados que han olvidado las suyas.

Las expectativas conductuales de una organización están implícitas en sus manuales de políticas e incluso en letreros anunciados en las salas de trabajo y los comedores. Sin embargo, la conducta que los miembros de la organización interiorizan también es crítica para el éxito de los esfuerzos de seguridad, parte del aspecto conductual de la seguridad es supervisar la conducta a intervalos irregulares para cerciorarse de que se están siguiendo los procedimientos apropiados y corregir cualesquier conductas que se podrían deteriorar con el tiempo. Hacer que el sistema registre el número de inicios de sesión fallidos de los usuarios es una forma de supervisar si usuarios no autorizados están intentando iniciar sesión en el sistema. La salida generada por el sistema se debe reconocer por su potencial de poner a la organización en riesgo en algunas circunstancias. Los controles para la salida incluyen pantallas que sólo se pueden acceder mediante la contraseña, la clasificación de información (es decir, a quién se puede distribuir y cuándo) y el almacenamiento seguro de documentos impresos y almacenados, sin importar el formato. (Kendall & Kendall, 2011)

2.9. PRUEBAS DE SOFTWARE

2.9.1. *Pruebas de caja blanca*

La prueba de caja blanca se basa en el diseño de casos de prueba que usa la estructura de control del diseño procedimental para derivarlos. Mediante la prueba de la caja blanca el ingeniero del software puede obtener casos de prueba que:

- Garanticen que se ejerciten por lo menos una vez todos los caminos independientes de cada módulo, programa o método.
- Ejerciten todas las decisiones lógicas en las vertientes verdadera y falsa.
- Ejecuten todos los bucles en sus límites operacionales.
- Ejerciten las estructuras internas de datos para asegurar su validez.

Por esto, la prueba de Caja Blanca es crucial para reducir errores y mejorar la calidad del software. (Anonimo, 2011)

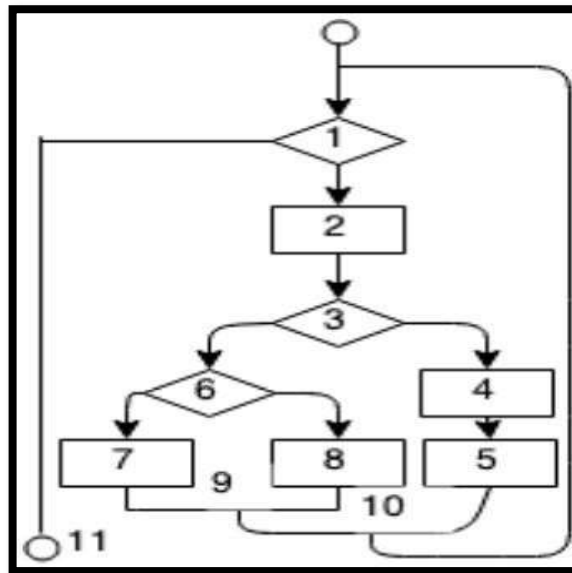
2.9.2. *Pruebas de ruta básica*

Este tipo de pruebas se basa en diseñar un caos de prueba por cada camino, permite obtener una medida de complejidad lógica de un procedimiento y usar esta medida como guía, se debe conocer la presentación que se conoce como grafo de flujo y las rutas de programa independiente.

- Notación de grafo de flujo: se usará para la presentación de flujo de control lógico. Se realiza a partir de PDL (Program Design Language) o pseudocódigo, con el objetivo de mapear el grafo de flujo correspondiente.

Figura 9

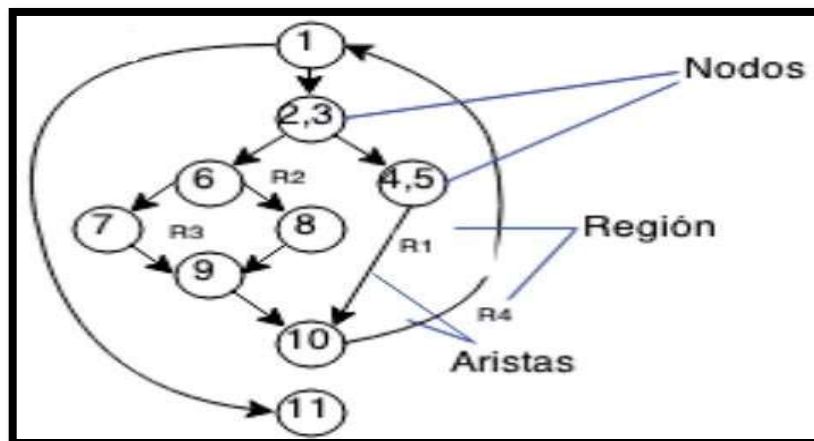
Diagrama de flujo.



Nota. Adaptado de Ingeniería de Software (pág. 415), por (Pressman, 2010).

Figura 10

Grafo de flujo.



Nota. Adaptado de Ingeniería de Software (pág. 415), por (Pressman, 2010).

Complejidad Ciclomática: proporciona una medición de software extremadamente útil, conjunto básico que garantiza que toda la ilustración será ejecutada al menos una vez.

Permite definir el número de caminos independientes del conjunto básico de un programa.

Número de regiones de gráfico de flujo que corresponde a la complejidad ciclomática, la complejidad ciclomática $V(G)$ para un grafo de flujo G se define como:

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

E representa aristas y N nodos del grafo de flujo, también se define como:

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

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

Matrices de Gráfos: Una matriz de grafo es una matriz cuadrada cuyo tamaño (es decir, número de filas y columnas) es igual al número de nodos del gráfico de flujo. Cada fila y columna corresponde a un nodo identificado y las entradas de la matriz corresponden a conexiones (una arista) entre nodos (Pressman, pág.420).

Figura 11

Matriz de Grafo.

Nodo	Conectado a nodo				
	1	2	3	4	5
1			a		
2					
3		d		b	
4		c			f
5		g	e		

Matriz de grafo

Nota. Adaptado de Ingeniería de Software (pág. 420), por (Pressman, 2010).

2.9.3. Pruebas de caja negra

Estas pruebas permiten obtener un conjunto de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. En ellas se ignora la estructura de control, concentrándose en los requisitos funcionales del sistema y ejercitándolos.

La prueba de Caja Negra no es una alternativa a las técnicas de prueba de la Caja Blanca, sino un enfoque complementario que intenta descubrir diferentes tipos de errores a los encontrados en los métodos de la Caja Blanca.

Muchos autores consideran que estas pruebas permiten encontrar:

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

Para preparar los casos de pruebas hacen falta un número de datos que ayuden a la ejecución de los estos casos y que permitan que el sistema se ejecute en todas sus variantes, pueden ser datos válidos o inválidos para el programa según si lo que se desea es hallar un error o probar una funcionalidad. Los datos se escogen atendiendo a las especificaciones del problema, sin importar los detalles internos del programa, a fin de verificar que el programa corra bien.

Para desarrollar la prueba de caja negra existen varias técnicas, entre ellas están:

- Técnica de la Partición de Equivalencia: esta técnica divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del software.
- Técnica del Análisis de Valores Límites: esta Técnica prueba la habilidad del programa para manejar datos que se encuentran en los límites aceptables.

- Técnica de Grafos de Causa-Efecto: es una técnica que permite al encargado de la prueba validar complejos conjuntos de acciones y condiciones.

Dentro del método de Caja Negra la técnica de la Partición de Equivalencia es una de las más efectivas pues permite examinar los valores válidos e inválidos de las entradas existentes en el software, descubre de forma inmediata una clase de errores que, de otro modo, requerirían la ejecución de muchos casos antes de detectar el error genérico. La partición equivalente se dirige a la definición de casos de pruebas que descubran clases de errores, reduciendo así en número de clases de prueba que hay que desarrollar. (Anonimo, 2011)

2.9.4. Pruebas de estrés

Una prueba de estrés (stress) consiste en probar los límites que un sistema puede soportar. En este tipo de pruebas se suele enviar más peticiones de las que el software podría atender normalmente para saber el comportamiento de la aplicación.

Por otro lado, las pruebas de estrés (stress testing) son realizadas sobrecargando un sistema más allá de sus especificaciones, para verificar cómo y cuándo fallará. Dentro de informática podemos colocar una gran carga en la base de datos, entradas (peticiones) continuas al sistema o almacenar información más allá de la capacidad de memoria del sistema. (Platzi, 2018)

CAPÍTULO III

3. MARCO APLICATIVO

3.1. INTRODUCCIÓN

En este capítulo se dará paso al diseño y desarrollo del sistema, los diversos procedimientos más relevantes, con el fin de identificar los requerimientos que se precisan en el sistema, también se identificarán los principales actores, de esta forma se logrará preparar un adecuado desarrollo del sistema.

3.2. DESARROLLO DE LA METODOLOGÍA

Para obtener la base de requerimientos funcionales y no funcionales, debemos conseguir la información clara y precisa a través de la captura, análisis y especificación de requisitos, para el correcto desarrollo de esta fase se empleó el uso de herramientas provistas para la captura de las necesidades a las que debe responder el sistema, aplicando las herramientas descritas.

Tabla 8

Herramientas para la obtención de requerimientos

HERRAMIENTA	CARACTERÍSTICAS
Observación	Se hizo la observación del proceso de solicitud de instalación, asignación a cuadrillas, y alta de servicio de Tv e Internet.
Documentación	Se revisaron los documentos que se manejan dentro del proceso, el tipo de presentación, las normas que se debe seguir y los formatos para la realización de los reportes
Entrevistas	Se realizó entrevistas con el personal involucrado dentro del proceso, quienes son: <ul style="list-style-type: none"> • Personal administrativo

-
- Personal técnico
-

Nota. Herramientas usadas para la ingeniería de requerimientos del sistema.

Después del análisis de requerimientos se empleó las herramientas que provee la metodología UWE.

3.2.1. Requerimientos Funcionales

A continuación, se describen las características mínimas que necesita el sistema para los diferentes procesos.

Tabla 9

Requerimientos Funcionales

ROL	DESCRIPCIÓN	FUNCIÓN
F-1	Acceder al sistema	Accede al sistema 3 tipos de usuarios (Administrador, plataforma, cuadrilla y clientes), mediante su usuario y contraseña
F-2	Gestionar Usuarios	Permite registrar, editar, listar, cambio de contraseña, dar de baja a usuarios
F-3	Administración del software	Permite al usuario con el perfil de Administrador, tener acceso total a los módulos y operaciones de la gestión de la información: creación, listado, actualización, inactivación de los recursos del sistema.
F-4	Gestionar los paquetes de tv e Internet	Permite registrar, editar, listar los paquetes individuales y paquetes, de acuerdo a la grilla de canales y banda ancha.
F-5	Gestionar solicitudes de paquetes	Permite administrar, la designación de la instalación a los técnicos, denominados cuadrillas

F-6	Gestionar Clientes	Permite registrar nuevos clientes, dar de alta y baja, cambio de planes.
F-7	Administrar solicitudes	Dar pronta respuesta, a los mensajes de asistencia técnica de los clientes.
F-8	Seguimiento de pago de servicios	Permite generar los pagos mensuales de acuerdo al paquete adquirido
F-9	Generar reportes de usuarios, clientes y cuadrillas	Permite generar un reporte pdf de los usuarios de acuerdo al rol especificado.
F-10	Reportes de solicitudes de Instalación	Permite generar la lista de solicitudes de instalación, y designar a una cuadrilla a una instalación.
F-15	Modificación del proyecto	Antes de que el proyecto se de alta, se podrá modificar las veces que haga falta.

Nota. Requerimientos funcionales del Sistema, con las respectivas especificaciones.

3.2.2. Requerimientos no Funcionales

Requerimientos no funcionales (RFN), se describen cualidades y características del sistema que el usuario puede visualizar.

Tabla 10

Requerimientos no funcionales

ROL	FUNCIÓN
RNF-1	El Software debe ser compatible en todos los navegadores, debe funcionar correctamente en: Google Chrome, Mozilla Firefox, IE/Edge y otros.
RNF-2	El sistema debe tener una interfaz gráfica, amigable para que el usuario.
RNF-3	El sistema debe tener un menú

RNF-4	El soporte y mantenimiento periódico del sistema asegura un buenrendimiento al sistema.
RNF-5	El sistema deberá responde a todas las solicitudes del usuario
RNF-6	El sistema deberá encriptar la contraseña de los usuarios
RNF-7	La sesión tendrá un determinado periodo de tiempo activo.
RNF-8	El sistema debe ser escalable a agregar nuevos módulos y funciones a futuro.

Nota. Requerimientos funcionales del Sistema, con las respectivas especificaciones.

3.2.3. Descripción de Actores

Este proceso permitirá la identificación de los actores que interactuarán con el sistema se tomó en cuenta los roles que cumplen diferentes personas de la empresa FENIX TELECOM.

Tabla 11

Descripción de Actores

DESCRIPCIÓN	RESPONSABILIDADES
Administrador	Se encarga de gestionar los usuarios dentro del sistema web, la asignación de roles y permisos. Se encarga de las configuraciones del sistema web.
Plataforma	Se encarga de gestionar las solicitudes de instalación. Designar las cuadrillas de instalación
Cuadrilla	Se encarga de la instalación de acuerdo a las solicitudes y designación de plataforma.
Cliente	Administra su información de paquetes adquiridos

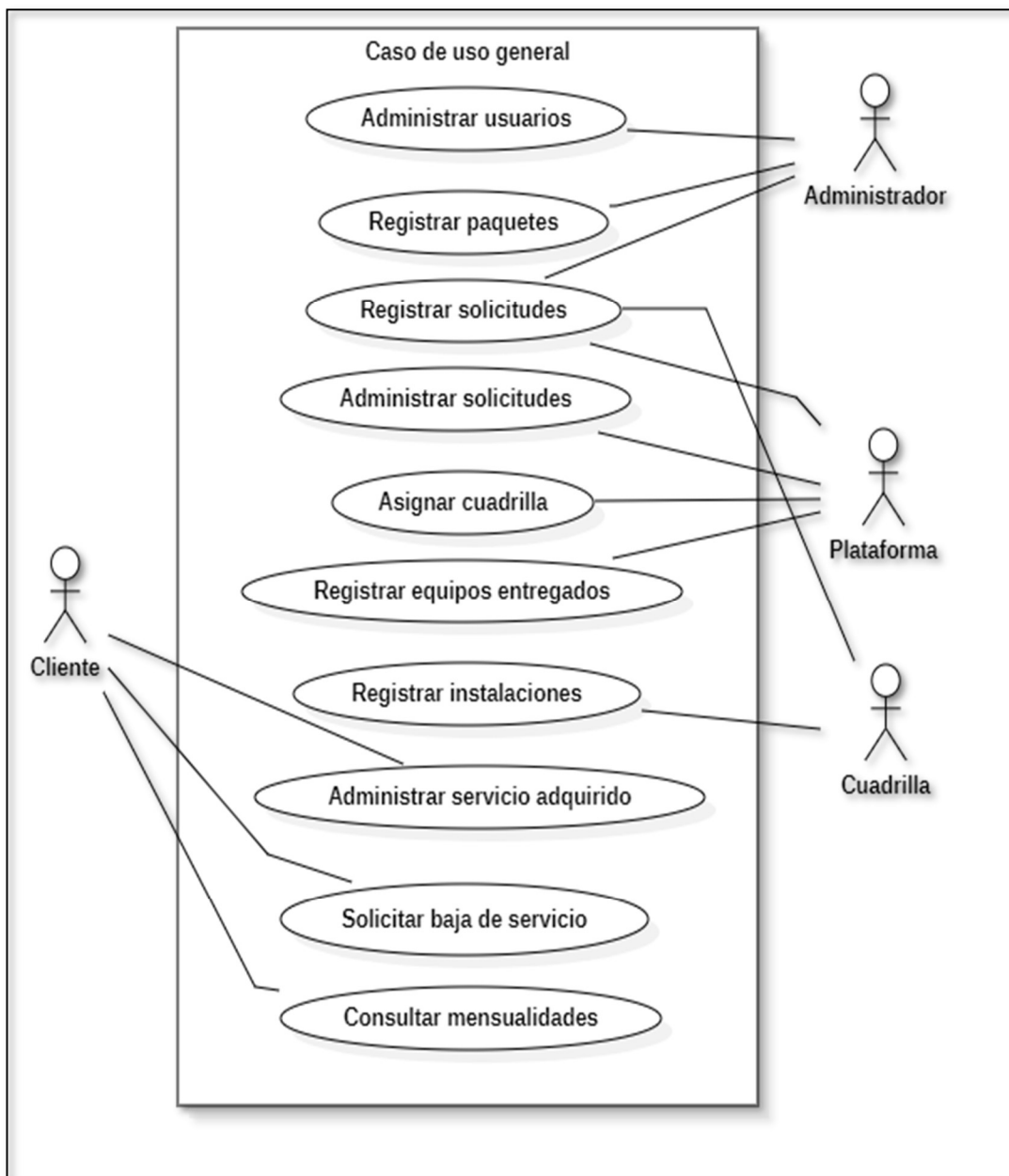
Nota. Descripción de actores que interactúan con el software.

3.2.4. Diagrama de Caso de Uso General del Sistema

En la siguiente figura presentamos el caso de uso general del sistema, donde identificamos a todos los actores.

Figura 12

Caso de uso general del sistema



Nota. Descripción de actores que interactúan con el software.

3.2.5. Diagramas de Casos de Uso Específicos del Sistema

A continuación, se mostrará los diagramas de caso de uso en específico de los procesos del sistema.

Figura 13
Gestión de usuarios

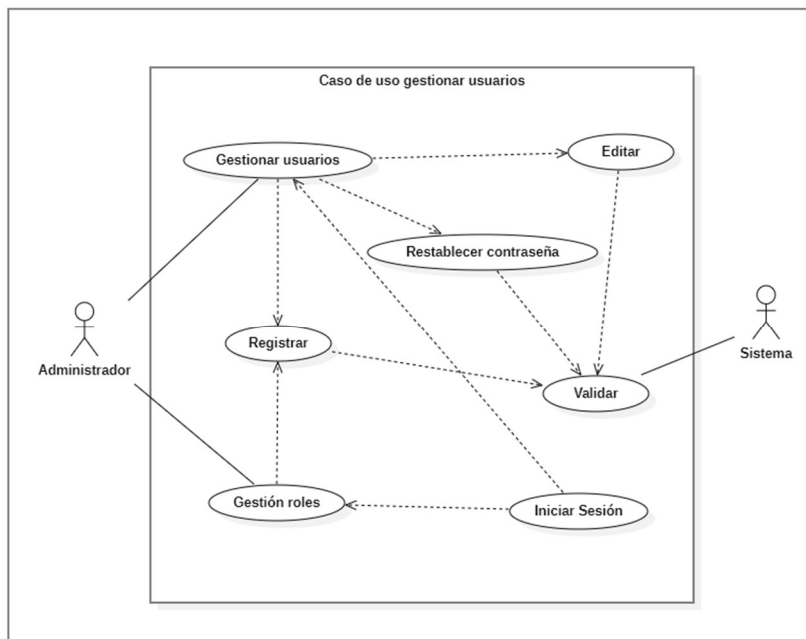


Tabla 12
Descripción del caso de uso - Gestión de usuarios.

CASO DE USO	GESTIÓN DE USUARIOS
Identificador:	CU01
Descripción:	El sistema posibilita administrar a los usuarios que se han registrado en él.
Actores:	Administrador
Precondiciones:	Para acceder al panel de administración de usuarios, es necesario que el usuario esté registrado en el sistema, haya iniciado sesión satisfactoriamente y entre al panel correspondiente.
Flujo de Eventos:	El proceso inicia cuando el usuario elige la función de Administración de usuarios. 1. Se muestra al usuario una lista de los usuarios registrados en el sistema.

2. El usuario solicita realizar acciones como crear, modificar, restablecer contraseña, activar o desactivar un usuario.
3. El sistema solicita la información necesaria para llevar a cabo las acciones mencionadas.
4. Luego de validar los datos proporcionados, el sistema ejecuta la acción solicitada por el usuario.

Post Condiciones:

El usuario puede realizar operaciones como crear, editar, restablecer la contraseña, activar o desactivar un usuario.

Flujos alternativos:

Si el usuario intenta crear o modificar un usuario con información que ya existe, se mostrará un mensaje de error indicando "El usuario ya está registrado".

Figura 14

Caso de uso Gestión de paquetes

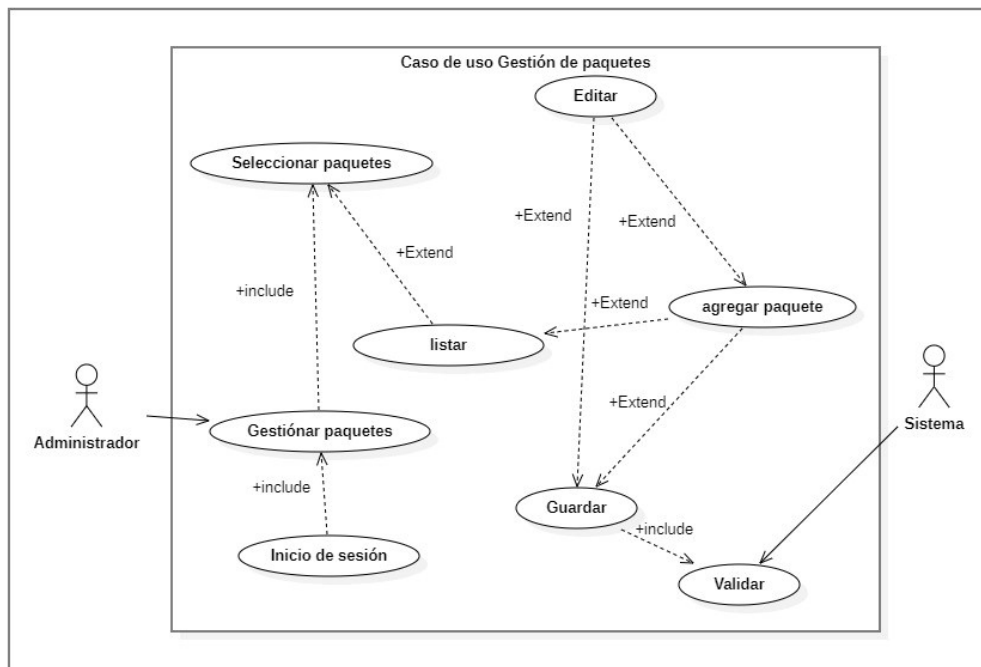


Tabla 13

Descripción de caso de uso de gestionar paquetes

CASO DE USO	GESTIÓN DE PAQUETES
Identificador:	CU02
Descripción:	El sistema permite gestionar los paquetes que se encuentran registrados en el sistema.
Actores:	Administrador
Precondiciones:	Para acceder al panel de administración de paquetes, es necesario que el usuario esté registrado en el sistema, haya iniciado sesión satisfactoriamente y entre al panel correspondiente.
Flujo de Eventos:	<p>El proceso inicia cuando el usuario elige la función de Administración de paquetes</p> <ol style="list-style-type: none"> 1. Se muestra al usuario una lista de los paquetes registrados en el sistema. 2. El usuario solicita realizar acciones como crear o modificar un paquete 3. El sistema solicita la información necesaria para llevar a cabo las acciones mencionadas. 4. Luego de validar los datos proporcionados, el sistema ejecuta la acción solicitada por el usuario.
Post Condiciones:	El usuario puede realizar operaciones como crear o editar al paquete
Flujos alternativos:	Si el usuario intenta crear o modificar un paquete con información que ya existe, se mostrará un mensaje de error indicando "El paquete ya está registrado".

Figura 15

Diagrama de caso de uso de administradores.

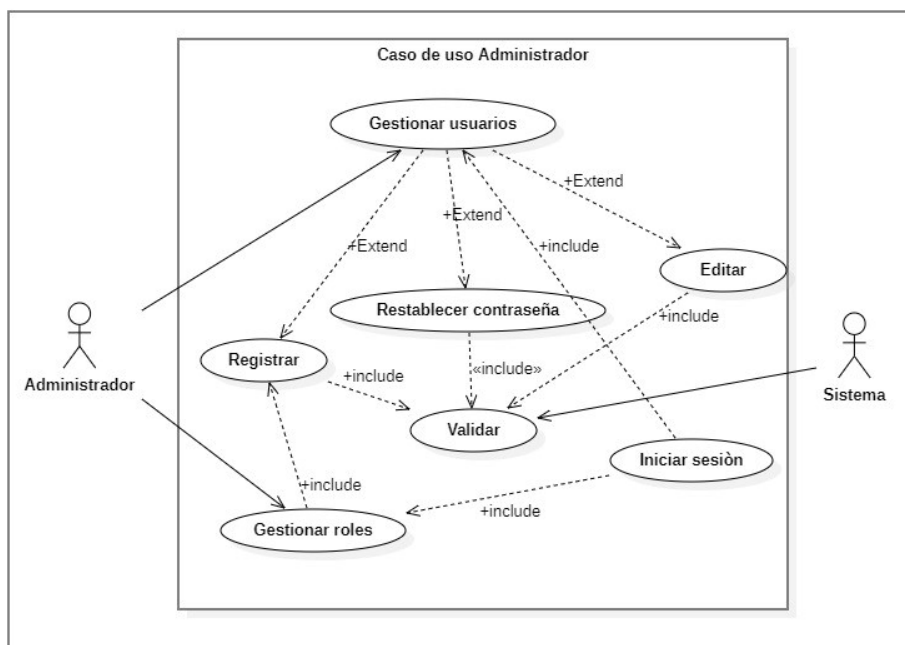


Tabla 14

Descripción de caso de uso de administradores.

CASO DE USO	ADMINISTRADORES
Identificador:	CU03
Descripción:	El sistema posibilita administrar a los usuarios que se han registrado en él.
Actores:	Administrador
Precondiciones:	Para acceder al panel de administración de usuarios, es necesario que el usuario esté registrado en el sistema, haya iniciado sesión satisfactoriamente y entre al panel correspondiente.
Flujo de Eventos:	El proceso inicia cuando el usuario elige la función de Administración de usuarios. <ol style="list-style-type: none"> 1. Se muestra al usuario una lista de los usuarios registrados en el sistema.

2. El usuario solicita realizar acciones como crear, modificar, restablecer contraseña, activar o desactivar un usuario.
3. El sistema solicita la información necesaria para llevar a cabo las acciones mencionadas.
4. Luego de validar los datos proporcionados, el sistema ejecuta la acción solicitada por el usuario.

Post Condiciones:

El usuario puede realizar operaciones como crear, editar, restablecer la contraseña, activar o desactivar un usuario.

Flujos alternativos:

Si el usuario intenta crear o modificar un usuario con información que ya existe, se mostrará un mensaje de error indicando "El usuario ya está registrado".

Figura 16

Diagrama de caso de uso asignación de cuadrillas

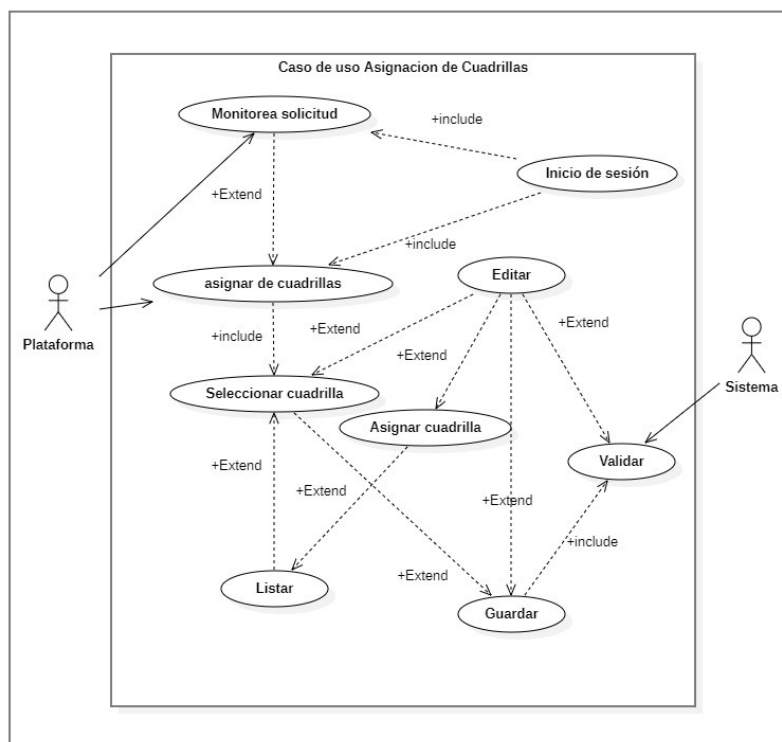


Tabla 15

Descripción de caso de uso de gestionar presupuestos.

CASO DE USO	ASIGNACIÓN DE CUADRILLAS
Identificador:	CU04
Descripción:	El sistema posibilita administrar a las cuadrillas que se han registrado en el sistema,
Actores:	Administrador, plataforma
Precondiciones:	Para acceder al panel de administración cuadrillas, es necesario que el usuario esté registrado en el sistema, haya iniciado sesión satisfactoriamente y entre al panel correspondiente e ingresar al panel de cuadrillas para asignar las instalaciones.
Flujo de Eventos:	<p>El caso se inicia al acceder al menú de cuadrilla</p> <ol style="list-style-type: none"> 1. El sistema habilita al usuario para seleccionar una gestión específica. 2. Se presenta al usuario un listado de todas las cuadrillas registradas 3. El sistema solicita los detalles de la asignación de cuadrillas 4. Una vez completada la información, el usuario presiona el botón de guardar. 5. El sistema realiza la validación de los datos ingresados.

Post Condiciones: La plataforma logra asignar a una cuadrilla a una solicitud

Flujos alternativos: La plataforma deberá validar para su asignación.

Figura 17

Caso de uso de servicios adquiridos

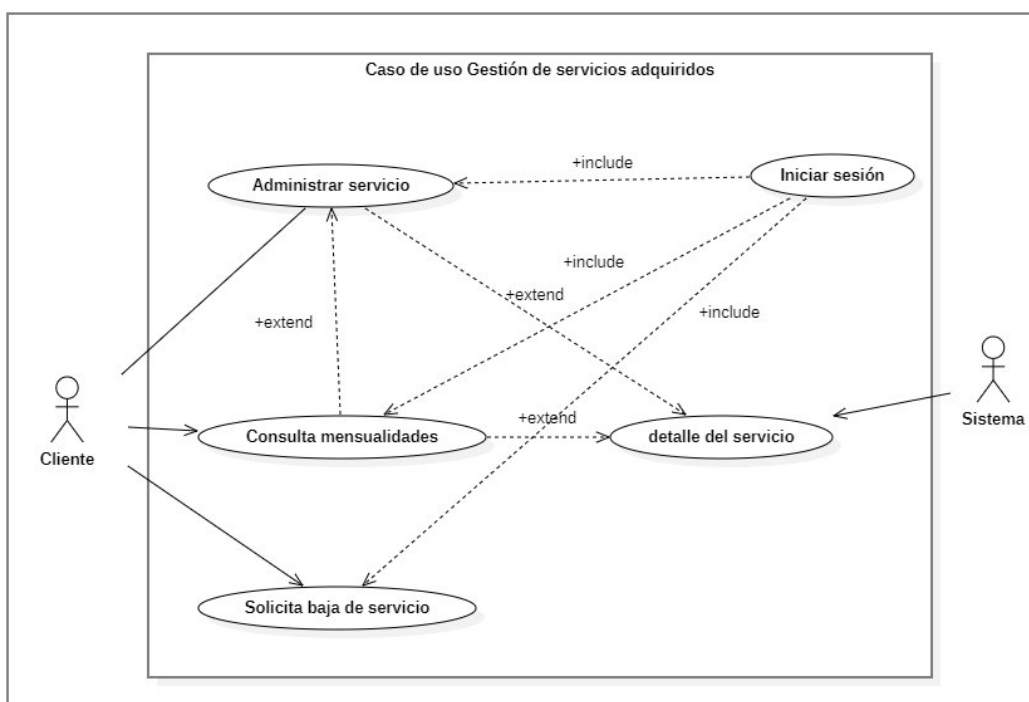


Tabla 16

Descripción de caso de uso de gestión de servicios adquiridos

CASO DE USO	GESTIÓN DE SERVICIOS
Identificador:	CU04
Descripción:	El sistema posibilita administrar los servicios adquiridos por el cliente que se han registrado en el sistema,
Actores:	Cliente

Precondiciones:	Para acceder al panel de administración de servicios, es necesario que el usuario esté registrado en el sistema, haya iniciado sesión satisfactoriamente y entre al panel correspondiente e ingresar al panel de servicios
------------------------	--

Flujo de Eventos:	<p>El caso se inicia al acceder al menú del cliente</p> <ol style="list-style-type: none">1. El sistema habilita al usuario para seleccionar una gestión específica.2. Se presenta al usuario un listado de los servicios adquiridos3. El sistema solicita los detalles de los servicios4. Una vez completada la información, el usuario presiona el botón de guardar.5. El sistema realiza la validación de los datos ingresados.
--------------------------	--

Post Condiciones:	La plataforma logra registrar y/o administrar los servicios del cliente
--------------------------	---

Flujos alternativos:	La plataforma deberá validar para el registro
-----------------------------	---

Figura 18

Caso de uso Registro de solicitudes

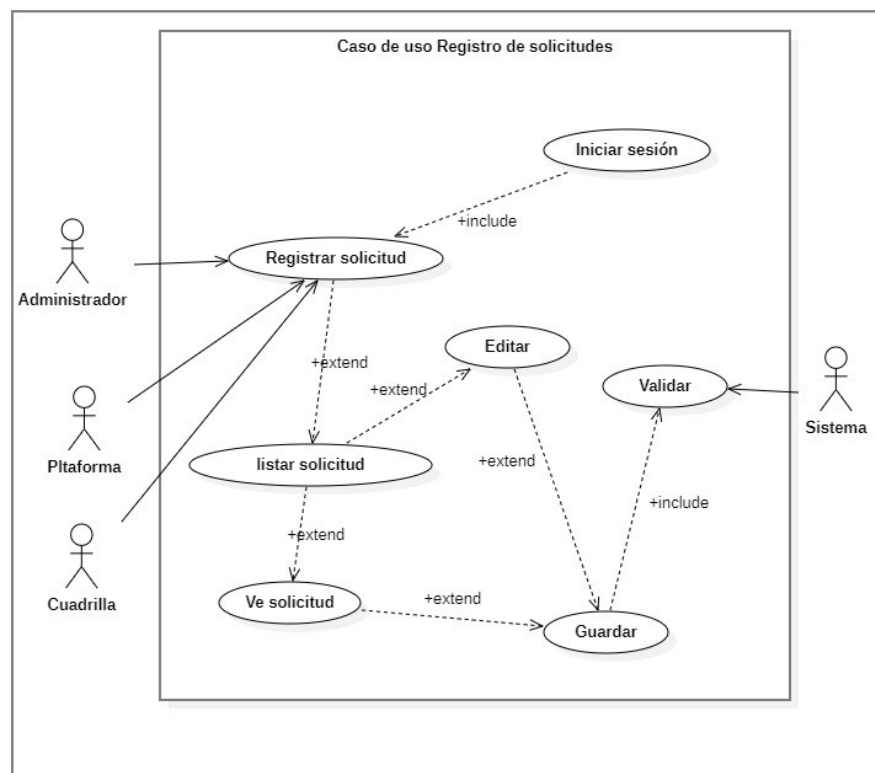


Tabla 17

Caso de uso de registro de solicitud

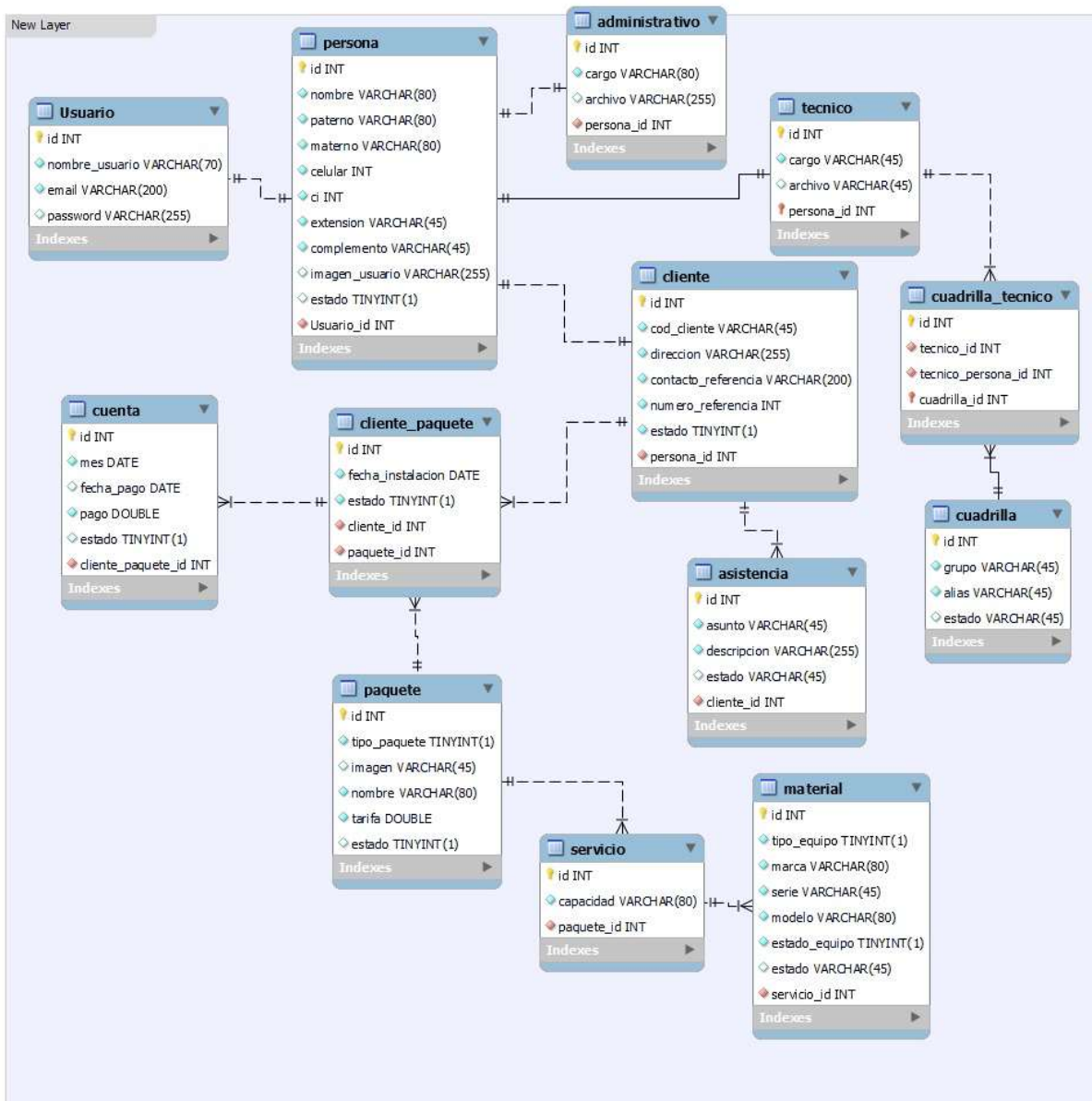
CASO DE USO	REGISTRO DE SOLICITUD
Identificador:	CU05
Descripción:	El sistema posibilita registrar las solicitudes de los clientes adquiridos
Actores:	Administrado, plataforma, cuadrilla
Precondiciones:	Para acceder al panel de administración de solicitud es necesario que el usuario esté registrado en el sistema, haya iniciado sesión satisfactoriamente y entre al panel correspondiente e ingresar al panel
Flujo de Eventos:	El caso se inicia al acceder al menú de cada usuario 1. El sistema habilito al usuario para seleccionar una gestión específica.

	<ol style="list-style-type: none">2. Se presenta al usuario un listado de las solicitudes3. Una vez completada la información, de la solicitud presiona el botón de guardar.4. El sistema realiza la validación de los datos ingresados.
Post Condiciones:	El administrado, plataforma y cuadrilla logra registrar las solicitudes en el sistema
Flujos alternativos:	El administrado, plataforma y cuadrilla deberá validar para el registro

3.2.6. Diseño conceptual

Tabla 18

Diseño conceptual



3.2.8. Modelo de navegación

Figura 19

Modelo navegacional - Administrador

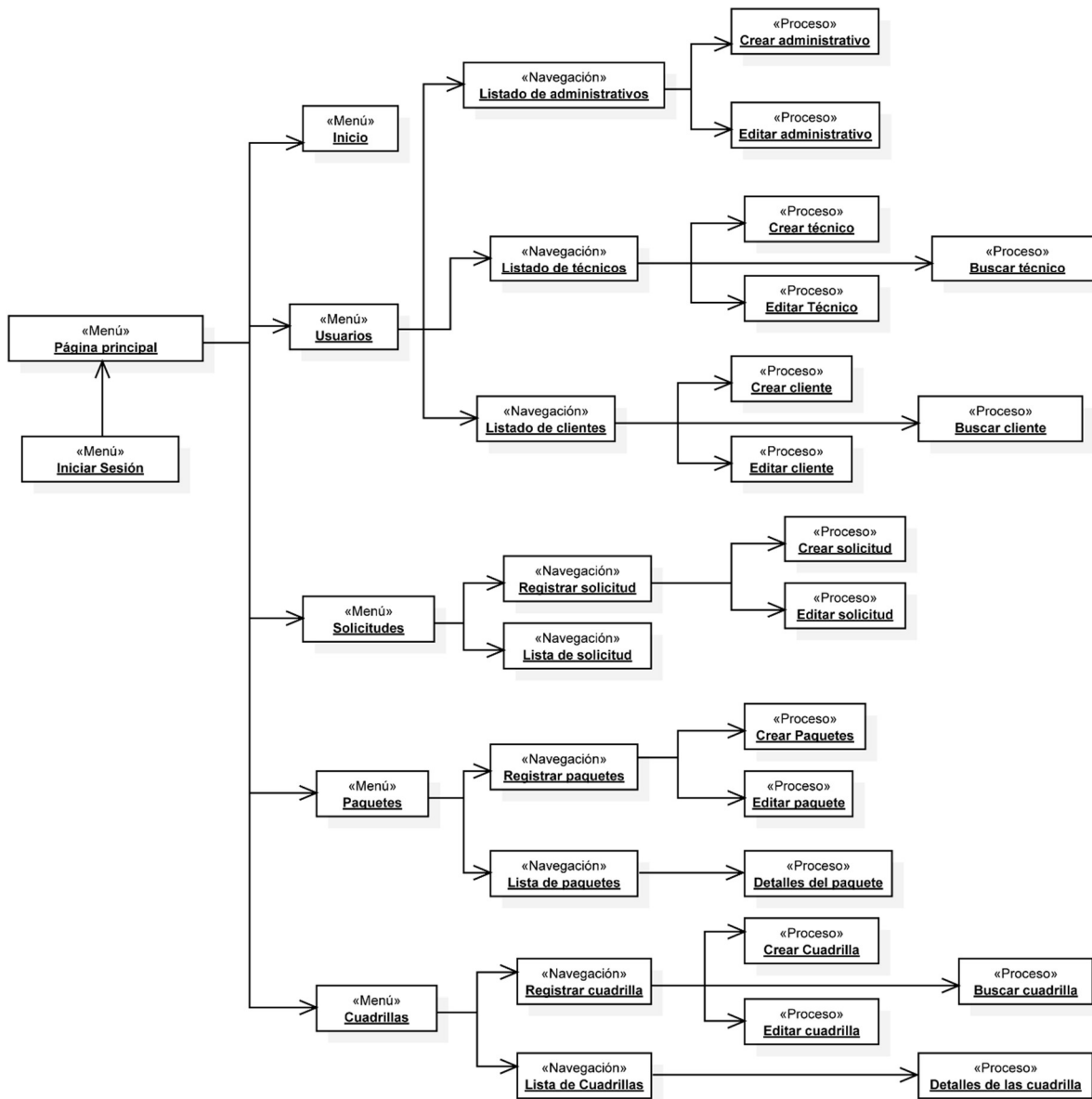


Figura 20
Modelo navegacional - Plataforma

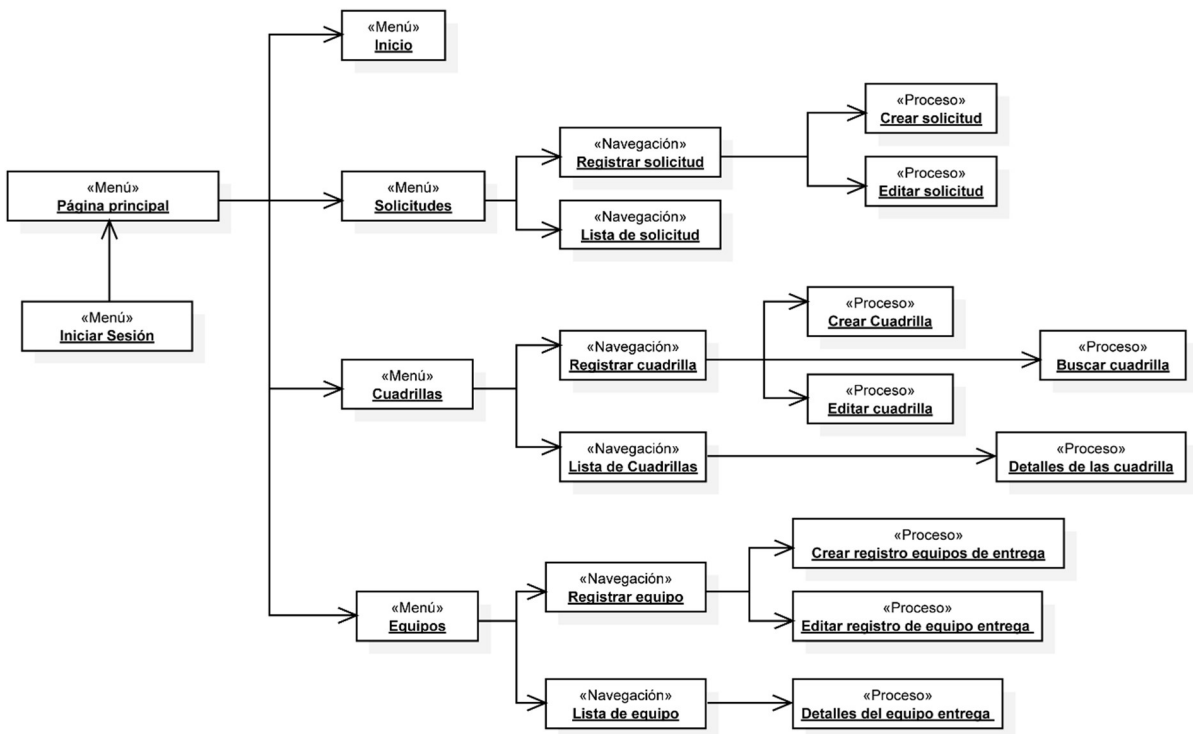
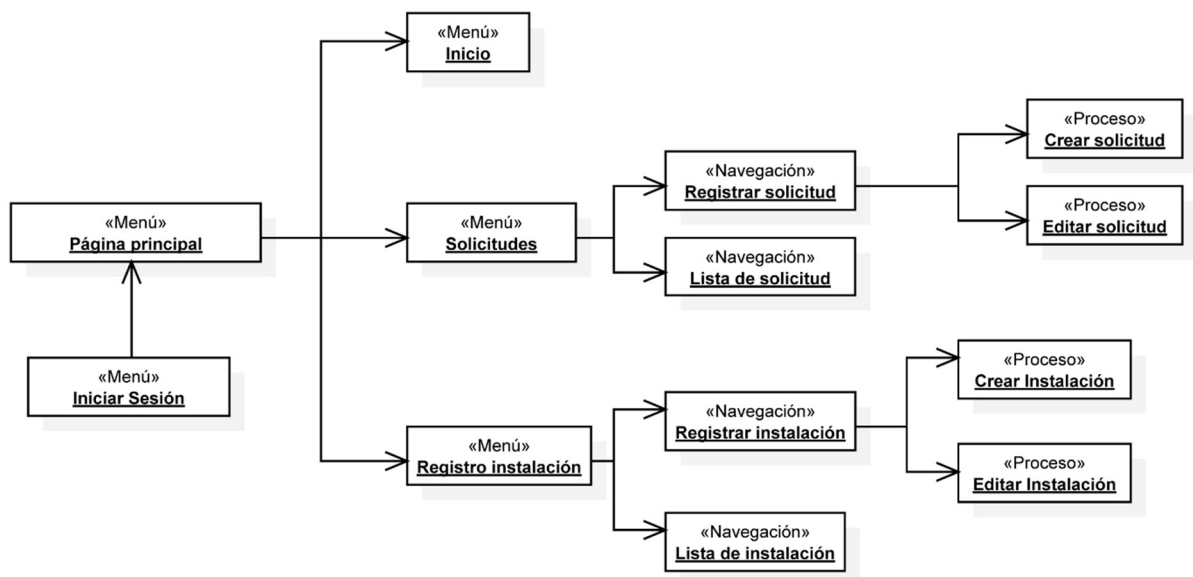


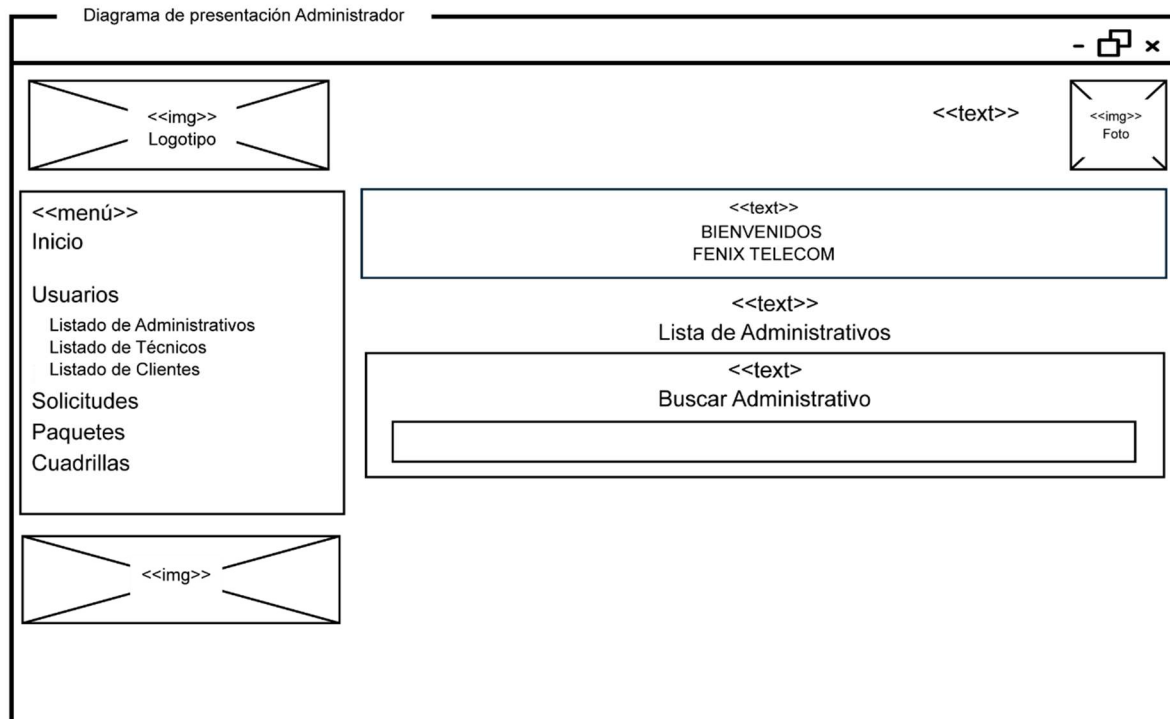
Figura 21
Modelo navegacional - Cuadrilla



3.2.9. Modelo de presentación

Figura 22

Modelo de presentación Administrador



3.2.10. Modelo de implementación

Figura 23

Página principal FENIX

FENIX TELECOM

8 MEGAS
120 BS
VER PLANES

NUESTROS SERVICIOS

INTERNET
Compartibilidad con otros dispositivos
VER MAS

TV CABLE
Compartibilidad con otros dispositivos
VER MAS

INCLUYE

INSTALACION
Compartibilidad con otros dispositivos
VER PLANES

CONFIGURACION
Compartibilidad con otros dispositivos
VER PLANES

TUS SERVICIOS BAJO TU CONTROL
La manera más sencilla de realizar consultas y administrar tus servicios.

PRECIOS A TU ALCANCE
La manera más sencilla de realizar consultas y administrar tus servicios.

ASESORAMIENTO PERSONAL
La manera más sencilla de realizar consultas y administrar tus servicios.

SOLICITA TU PLAN

FENIX TELECOM

INFORMACIÓN DE CONTACTO

CONTACTO
 • Teléfono: +593 9 950 00000
 • Email: info@fenixtelecom.com.ec
 • WhatsApp: +593 9 950 00000

SIGUEMOS

UBICACIÓN

QUITANDI
 • Dirección: La Paj
 • Calle: 20 de Agosto, Cuy. Alameda de la Paj, Quito, Pinar de la Paj, Pinar de la Paj, Pinar de la Paj

AVILAJO
 • Dirección: La Paj
 • Calle: 20 de Agosto, Cuy. Alameda de la Paj, Quito, Pinar de la Paj, Pinar de la Paj

ESTÁ EMPRESA ESTÁ REGULADA Y FISCALIZADA POR:

ATT

Figura 24
Página principal servicios



Figura 25

Página principal planes



Figura 26

Iniciar Sesión FENIX



CAPÍTULO IV

4. CALIDAD, PRUEBAS, COSTOS Y SEGURIDAD DEL SOFTWARE

4.1. MÉTRICAS DE CALIDAD

Para asegurar el correcto desarrollo del proyecto, se han implementado métricas de calidad que permiten evaluar diversos parámetros del sistema. Estas métricas son fundamentales para especificar de manera ordenada las características y atributos del software. Entre los diversos modelos y criterios de calidad para productos de software, uno de los más destacados es el ISO 25000. Este estándar ofrece un marco integral para evaluar la calidad del software y se utiliza para medir varios aspectos, incluyendo la funcionalidad, fiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad del sistema

4.1.1. *Funcionalidad*

Se llevará a cabo una exhaustiva evaluación para determinar si el software cumple con los requisitos funcionales establecidos. Este proceso garantiza que el programa posea las características esenciales para satisfacer las demandas y expectativas del usuario, asegurando su plena funcionalidad.

4.1.1.1. *Punto función.* El proceso de desarrollo de la técnica de estimación de software se apoya en la consideración de cinco atributos clave, los cuales son fundamentales para asegurar tanto la precisión como la eficacia en la estimación del software. Uno de estos atributos cruciales es la cantidad de entradas de usuario, estas entradas representan la información proporcionada al sistema por parte de los usuarios.

La importancia de esta métrica radica en su capacidad para influir en la complejidad y el esfuerzo requeridos tanto para el desarrollo inicial como para el mantenimiento continuo del sistema, cada entrada de usuario tiene el potencial de impactar la arquitectura del software, lo que a su vez repercute directamente en su diseño y en su capacidad de mantenimiento a largo

plazo. Por lo tanto, es esencial tener en cuenta estos factores durante el proceso de estimación, ya que contribuyen significativamente a una estimación precisa y a una gestión efectiva del proyecto de desarrollo de software.

Tabla 19

Número de interacciones de usuario

ENTRADAS DE USUARIO	
Administración de usuarios	1
Administración de técnicos	1
Administración de usuarios	30
TOTAL	32

Nota. Entradas de usuario de acuerdo al software en desarrollo

Salidas de usuario, la cantidad de resultados que el sistema entrega incluye diversos tipos de outputs, como mensajes, notificaciones, informes y alertas. Estos outputs son cruciales para la interacción del usuario con el sistema y su capacidad de responder a las necesidades informativas.

Tabla 20

Cantidad de salidas de usuario

SALIDAS DE USUARIO	
Administración de usuarios	3
Administración de técnicos	3
Administración de usuarios	21
TOTAL	27

Nota. Entradas de usuario de acuerdo al software en desarrollo

Peticiones de usuario, es la cantidad de veces que una entrada del usuario desencadena una respuesta interactiva del software, lo que refleja la capacidad del sistema para reaccionar y procesar información en tiempo real.

Tabla 21

Cantidad de peticiones del usuario.

PETICIONES DE USUARIO	
Administración de usuarios	3
Administración de técnicos	3
Administración de usuarios	8
TOTAL	14

Nota. Entradas de usuario de acuerdo al software en desarrollo

Cantidad de archivos, este aspecto se refiere al total de archivos que maneja el sistema. Puede incluir tanto grupos lógicos de datos, que agrupan información relacionada de manera coherente, como archivos individuales que almacenan datos específicos.

Cantidad de interfaces externas, este parámetro indica el total de interfaces que el sistema puede leer o interpretar, es decir, todas las conexiones externas con otros sistemas, aplicaciones o dispositivos que intercambian información con él. Las interfaces externas pueden incluir APIs, servicios web, archivos de intercambio de datos y cualquier otra forma de comunicación legible por máquina. La cantidad de interfaces externas es importante porque influye en la integración y la interoperabilidad del sistema, así como en la complejidad de su desarrollo y mantenimiento.

Tabla 22

Parámetros de puntos función

PARÁMETROS DE MEDICIÓN	CUENTA
Número de entradas de Usuario	32
Número de salidas de usuario	27
Número de peticiones de usuario	14
Número de archivos	20
Número de interfaz externa	2

Nota. Tabla de los parámetros de medición

Después de contar los parámetros de función, se procede al cálculo de los puntos de función sin ajustar en la tabla siguiente, este proceso implica sumar las diversas métricas recolectadas, proporcionando una medida cuantitativa que refleja la funcionalidad total del sistema antes de considerar cualquier ajuste de complejidad.

Tabla 23

Punto de función bruto.

PARÁMETRO DE MEDICIÓN	CUENTA	FACTOR	TOTAL
Número de entradas de Usuario	32	6	192
Número de salidas de usuario	27	7	189
Número de peticiones de usuario	14	6	84
Número de archivos	20	15	300
Número de interfaz externa	2	10	20
TOTAL			785

Nota. Tabla que muestra los parámetros de medición.

En la tabla se muestra la suma total de los puntos de función sin ajustar, para obtener los valores ajustados se aplican factores de complejidad específicos que se detallan a continuación en la tabla 24. Estos factores consideran diversas características del sistema

Ajustar los puntos de función iniciales con estos factores permite una evaluación más precisa y realista de la carga de trabajo y los recursos necesarios para el desarrollo del software, este método asegura que todas las particularidades y desafíos del proyecto se reflejen adecuadamente en la estimación final.

Tabla 24

Variables de complejidad

IMPORTANCIA	0%	20%	40%	60%	80%	100%
ESCALA	NO INFLUENCIA	INCIDENTAL	MODERADO	MEDIO	SIGNIFICATIVO	ESENCIAL
FACTOR	0	1	2	3	4	5
¿Requiere el sistema copias de seguridad y de recuperación fiable?					X	5
¿Se requiere comunicación de datos?					X	5
¿Existe funciones de procesos distribuidos?					X	4
¿Es crítico el rendimiento?				X		4
¿El sistema web será ejecutado el SO Actual?					X	5
¿Se requiere una entrada interactiva para el sistema?				X		5
¿Se requiere que el sistema tenga entradas a datos con múltiples ventanas?				X		4

¿Se actualiza los archivos de forma interactiva?	X	5
¿Son complejas las entradas, salidas, los archivos o las peticiones?	X	5
¿Es complejo el procesamiento interno del sistema?	X	4
¿Se ha diseñado el código para ser reutilizado?	X	5
¿Se ha diseñado el sistema para facilitar al usuario el trabajo y ayudarlos a encontrar la información?	X	5
TOTAL		56

Nota. Aplicación de las métricas de calidad

Fórmula para ajustar puntos de función, que considera factores como la complejidad técnica, el rendimiento y la usabilidad.

(5) *Fórmula de ajuste*

$$PF = \text{conteo total} [0.65 + 0.01 * \sum fi] \quad (1)$$

Donde:

$\sum (fi)$: Sumatoria de los valores de los factores de ajuste.

Se obtiene:

$$\sum (fi) = 56$$

$$PF = 785 * [0.65 + 0.01 * 56]$$

$$PF = 949,85$$

Con el límite superior establecido en la suma de los factores de ajuste $\sum(fi) = 70$, se realiza el cálculo del valor siguiente en la secuencia, este proceso es fundamental para mantener la coherencia en la evaluación de la complejidad del sistema.

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

$$PFmax = 785 * [0.65 + (0.01 * 70)]$$

$$PF\ max = 1059.75$$

La medición de la funcionalidad se basa en comparar el valor máximo posible con el valor obtenido del punto de función ajustado, esta relación proporciona información crucial sobre la eficiencia y la complejidad del sistema evaluado.

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

$$Funcionalidad = \frac{949.85}{1059.75} * 100$$

$$Funcionalidad = 0.8962 * 100\%$$

$$\mathbf{Funcionalidad = 89.62\ \%}$$

Entonces, la eficacia del sistema alcanza el 89.62 %, indicando su alta capacidad operativa con un riesgo mínimo de fallos (10.38%),

4.1.2. Confiabilidad

La confiabilidad de un sistema se refiere a su capacidad para mantener un nivel de funcionamiento adecuado durante un período de prueba establecido que implica considerar aspectos como la madurez del sistema, su capacidad de recuperación frente a fallos, su tolerancia a posibles errores y su cumplimiento de los estándares de fiabilidad, para evaluar la confiabilidad, se observa el rendimiento del sistema desde el momento en que comienza a

funcionar hasta un punto determinado en el tiempo, registrando cualquier incidencia o cambio en su desempeño.

Considerando que se tiene en cuenta que:

(6) Probabilidad de fallas

$$P(T \leq t) \Rightarrow \text{Probabilidad de fallas} \quad (2)$$

(7) Probabilidad de trabajo sin fallas

$$P(T > t) = 1 - F(t) \Rightarrow \text{Probabilidad de trabajo sin fallas} \quad (3)$$

Se considera la siguiente función al calcular la confiabilidad del sistema.

(8) Función de la confiabilidad del sistema

$$R(t) = f * e^{-\mu * t} \quad (4)$$

Donde:

f	=	Funcionalidad del sistema
μ	=	Probabilidad de error del sistema
t	=	Tiempo de prueba del sistema

Durante un periodo de prueba de 21 días, se registra una falla por cada 10 ejecuciones del sistema.

$$F(t) = 0.89 * e^{-\frac{1}{10} * 22}$$

$$F(t) = 0.1089 * 100\%$$

$$F(t) = 10.89 \%$$

Sustituyendo:

$$P(T \leq t) = 0.1089 = 10.89\%$$

$$P(T > t) = 1 - 0.1089$$

$$P(T > t) = 0.8911 * 100\%$$

$$P(T > t) = 89.11 \%$$

Por lo tanto, la confiabilidad del sistema desarrollado se sitúa en un sólido 88.11%, verificada a lo largo de un periodo de prueba de 21 días, este resultado confirma la consistencia y fiabilidad del sistema en condiciones operativas durante este período de evaluación establecido.

4.1.3. Usabilidad

Al evaluar la usabilidad de un sistema, es importante considerar el factor humano, asegurando que el software cumpla con los requisitos y expectativas del usuario, por lo tanto, se lleva a cabo una evaluación mediante encuestas dirigidas a los usuarios del sistema.

Estas encuestas permiten recopilar información valiosa sobre la experiencia del usuario, su satisfacción y la facilidad de uso percibida que se utiliza luego para calcular la usabilidad del sistema, aplicando una fórmula diseñada específicamente para medir la eficacia y la experiencia del usuario en la interacción con el software.

(9) Función para determinar la usabilidad

$$FU = \left[\frac{\sum x_i}{n} * 100 \right] \quad (5)$$

Donde:

x_i = Valores de evaluación de cada pregunta

n = Número de preguntas

Tabla 25

Indicadores de facilidad de uso

ESCALA	VALOR
Muy bueno	5
Bueno	4

Regular	3
Malo	2
Pésimo	1

Nota. Aplicación de las métricas de calidad

Tabla 26

Usabilidad del sistema

PREGUNTA	SI	RESULTADO xi
¿Puede Utilizar con facilidad el sistema?	9	0.8
¿Puede Controlar operaciones que el sistema solicita?	8	0.8
¿Las Respuestas del sistema son complicadas?	2	0.9
¿El Sistema permitió la retroalimentación de información?	9	0.9
¿El sistema cuenta con interface agradable a la vista?	9	0.9
¿La respuesta del sistema es satisfactoria?	9	0.9
¿Le parece complicada las funciones del sistema?	8	0.8
¿Se hace difícil o dificultoso aprender a manejar el sistema?	1	0.8
¿Los resultados que proporciona el sistema facilitan el trabajo?	10	1
¿Durante el uso del sistema se produjo errores?	2	0.8

TOTAL	8.6
--------------	------------

Nota. *Aplicación de las métricas de calidad*

A partir de los datos recabados a través del cuestionario, llevamos a cabo un análisis exhaustivo para evaluar la usabilidad del sistema. Este proceso implica calcular diversos indicadores y métricas que reflejen la facilidad con la que los usuarios interactúan con el sistema.

$$FU = \left[\frac{8.6}{10} * 100 \right]$$

$$FU = [0.86 * 100]$$

$$FU = 86\%$$

Con esta información, podemos afirmar que la facilidad de uso del sistema alcanza un porcentaje del 87%, este dato proporciona una medida concreta de la experiencia del usuario y su capacidad para interactuar efectivamente con el sistema.

4.1.4. Eficiencia

La eficacia evalúa si el sistema utiliza eficientemente sus recursos. Para determinar su nivel, se considera una escala específica como referencia.

Tabla 27

Escala de valores de eficiencia.

ESCALA	VALOR
Excelente	5
Bueno	4
Aceptable	3
Deficiente	2
Pésimo	1

Nota. Aplicación de las métricas de calidad

Para evaluar la eficacia del sistema, se cuenta con la tabla siguiente.

Tabla 28

Evaluación de la eficacia del sistema.

PREGUNTA	PORCENTAJE
¿La distribución y estilo de la interfaz permite que un usuario introduzca con eficiencia las operaciones y la información?	5
¿Una secuencia de operaciones (o entrada de datos) puede realizarse con facilidad de movimientos?	4
¿Los datos de salida están presentados de modo que se entienden de inmediato?	5
¿Las operaciones jerárquicas están organizadas de manera que minimizan la navegación del usuario para hacer que alguna se ejecute?	4
¿Procesa y responde adecuadamente cuando realiza alguna consulta o búsqueda?	5
TOTAL	23

Nota. Tabla que muestra las preguntas de eficacia del sistema

Para determinar la eficacia, se emplea la fórmula siguiente.

(10) *Fórmula eficacia*

$$E = \frac{\sum x_i}{n} * \frac{100}{n} \quad (6)$$

Donde:

$\sum x_i$ = Sumatoria de los valores de eficiencia

n = Número de preguntas

Sustituyendo se tiene lo siguiente:

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

$$E = 92 \%$$

Con estos datos, determinamos que la eficiencia del sistema se sitúa en el 88%, lo que indica que los recursos del sistema se están utilizando de manera efectiva y óptima.

4.1.5. **Mantenibilidad**

Esta métrica evalúa la cantidad de trabajo requerido para realizar cambios en el sistema, ya sea para corregir errores o agregar nuevas funcionalidades. El estándar IEEE94 recomienda utilizar el índice de madurez del sistema como indicador de su estabilidad. Por lo tanto, la ecuación para este índice es la siguiente:

(11) Ecuación para el índice de madurez del sistema

$$IMS = \frac{[Mt - (Fa + Fc + Fd)]}{Mt} \quad (7)$$

Donde:

Mt = Número de módulos total de la versión actual

Fa = Número de módulos de la versión actual que se añadieron.

Fc = Número de módulos de la versión actual que se cambiaron.

Fd = Número de módulos de la versión anterior que se eliminaron en la versión actual.

Tabla 29

Indicadores de mantenibilidad.

DESCRIPCIÓN	VALOR
Mt	5

Fa	0
Fc	1
Fd	0

Nota. *Aplicación de las métricas de calidad*

Sustituyendo en la ecuación se tiene:

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

$$IMS = 0.8 * 100\%$$

$$**IMS = 80%**$$

En conclusión, el índice de mantenibilidad del sistema desarrollado alcanza el 80% indicando que el sistema tiene una alta capacidad para ser modificado, corregido y mejorado, facilitando su adaptación a nuevas necesidades y requisitos a lo largo del tiempo.

4.1.6. Portabilidad

La capacidad de un software para ser transferido de un entorno a otro incluye varios aspectos importantes:

- Adaptabilidad
- Habilidad del software para ajustarse a diferentes entornos sin requerir modificaciones adicionales.
- Facilidad de Instalación
- Esfuerzo requerido para instalar el software en un entorno específico.
- Conformidad

- Verificación de que el software cumple con estándares o convenciones de portabilidad.
- Capacidad de sustitución
- Facilidad y esfuerzo necesarios para sustituir el software con otro producto que tenga funciones similares.

El sistema, desarrollado con Laravel, puede implementarse en cualquier servidor con Apache y las herramientas necesarias.

Como la tecnología web es ejecuta fácilmente en cualquier dispositivo con conexión a Internet y un navegador.

(12) *Fórmula portabilidad*

$$Portabilidad = 1 - \left(\frac{\text{número de día para aportar el sistema}}{\text{número de días para implementar el sistema}} \right) \quad (8)$$

Sustituyendo la fórmula

$$Portabilidad = 1 - \left(\frac{1}{9} \right)$$

$$Portabilidad = 0.88 * 100\%$$

$$Portabilidad = 88\%$$

Con base en los resultados obtenidos, se puede determinar que la calidad total del sistema a partir de los parámetros medidos de esta manera se presenta un resumen de los resultados obtenidos:

Tabla 30
Resultados finales

CARACTERÍSTICAS	RESULTADOS
Funcionalidad	89.62 %
Confiabilidad	89.11 %
Usabilidad	86 %
Eficiencia	92 %
Mantenibilidad	80%
Portabilidad	88%
Evaluación total	87.45 %

Nota. Aplicación de las métricas de calidad

4.2. EVALUACIÓN DE COSTOS

Para llevar a cabo la finalidad de este proyecto es fundamental la planificación y estimación de los costos desde el inicio hasta su finalización, para calcular el costo total del sistema de FENIX TELECOM, se considerarán los siguientes costos:

El modelo COCOMO II permite hacer estimaciones basadas en el tamaño del software y un conjunto de factores de costo y de escala. COCOMO II cuenta con tres modelos diferentes, estos modelos ayudarán a estimar los costos de manera efectiva, asegurando que se consideren todas las variables relevantes para una planificación financiera adecuada del proyecto.

Orgánico, adecuado para proyectos relativamente simples, con menos de 5000 líneas de código, que implican procesamiento de datos y el uso de bases de datos centradas en transacciones y recuperación de información.

Semiacoplado, dirigido a proyectos de complejidad y tamaño intermedios, donde la experiencia varía y las restricciones son moderadas.

Empotrado, aplicable a proyectos muy complejos, con poca experiencia previa y un entorno de alta innovación técnica.

Para estimar los costos de este proyecto se utilizará el submodelo de post - arquitectura.

Para calcular el esfuerzo, es necesario determinar la variable KLOC (kilo-líneas de código). En este proyecto, se implementarán 8504 líneas de código. Realizando las conversiones necesarias se obtienen los valores requeridos.

$$LCD = 8517$$

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

$$KLCD = \frac{8517}{1000}$$

$$KLCD = 8,517$$

Es un modelo Intermedio y semi acoplado.

Tabla 31

Coeficientes de a, b, c y d de COCOMO II.

PROYECTO SOFTWARE	A	B	C	D
Orgánico	3.2	1.05	2.5	0.38
Semi acoplado	3.0	1.12	2.5	0.35
Empotrado	2.8	1.20	2.5	0.32

Nota. Tabla de aplicación de los coeficientes de COCOMO II

Dado que el proyecto no excede las 50000 líneas de código, se aplicarán los coeficientes para proyectos semi - acoplados, a continuación, se presentan las ecuaciones correspondientes al coeficiente seleccionado según las líneas de código para calcular el costo del software.

Tabla 32

Ecuaciones cocomo II

VARIABLE	ECUACIÓN	TIPO / UNIDAD
Esfuerzo requerido por el proyecto	$E = a * (KLDC)^b * FAE$	Persona/mes
Tiempo requerido por el proyecto	$T = c * (E)^d$	Mes
Número de personas requeridos para el proyecto	$NP = \frac{E}{T}$	Personas
COSTO TOTAL	$CT = SueldoMes * NP * T$	\$us

Nota. Tabla que muestra las ecuaciones que aplica COCOMO II

Para obtener los valores de FAE, se recurrirá a la tabla de multiplicadores de atributos.

Tabla 33

Cálculo de Atributos FAE.

Atributos	VALOR					
	Muy bajo	Bajo	Nominal	Alto	Muy alto	Extra alto
ATRIBUTOS DE SOFTWARE						
Fiabilidad	0,75	0,88	1,00	0,15	1,40	
Tamaño de Base de datos		0,94	1,00	1,08	1,16	
Complejidad	0,70	0,85	1,00	1,15	1,30	1,65
ATRIBUTOS DE HARDWARE						
Restricciones de tiempo de ejecución			1,00	1,11	1,30	1,66
Restricciones de memoria virtual			1,00	1,06	1,21	1,56
Volatilidad de la máquina virtual		0,87	1,00	1,15	1,30	
Tiempo de respuesta		0,87	1,00	1,07	1,15	

ATRIBUTOS DE PERSONAL					
Capacidad de análisis	1,46	1,19	1,00	0,86	0,71
Experiencia en la aplicación	1,29	1,13	1,00	0,91	0,82
Calidad de los programadores	1,42	1,17	1,00	0,86	0,70
Experiencia en la máquina virtual	1,21	1,10	1,00	0,90	
Experiencia en el lenguaje	1,14	1,07	1,00	0,95	
ATRIBUTOS DEL PROYECTO					
Técnicas actualizadas de programación	1,24	1,10	1,00	0,91	0,82
Utilización de herramientas de software	1,24	1,10	1,00	0,91	0,83
Restricciones de tiempo de desarrollo	1,22	1,08	1,00	1,04	1,10

Nota. Tabla que muestra los datos cuantitativos de atributos del proyecto

Por tanto, nuestro Factor de ajuste será

$$FAE = 1.15 * 1.00 * 1.15 * 1.00 * 1.00 * 0.87 * 1.00 * 0.86 * 0.82 * 0.86 * 0.90 * 0.95 * 0.82 * 0.91 * 1.00 = 0.4452$$

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

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

$$E = 3 * 8,504^{1.05} * 0.4452 \text{ (persona/mes)}$$

$$E = 12.69 \text{ (persona/mes)}$$

Cálculo del Tiempo

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

$$T = 2.5 * 12.69^{0.35}(\text{meses})$$

$$T = 6.08 \text{ (meses) esto equivale 6 meses}$$

Cálculo del personal requerido

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

$$NP = \frac{12.69}{6.08} \text{ personas}$$

$$NP = 2.08 \cong 2 \text{ persona}$$

Cálculo de Costo Total

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

$$CT = 359,2 * 2 * 6$$

$$CT = 4310,4 \$us \cong 30000,4 Bs$$

En resumen, se necesitan dos personas para estimar un trabajo de seis meses, con un costo total de 4310,4 dólares, lo que equivale a 30000,4 bolivianos.

4.3. SEGURIDAD

Uno de los elementos fundamentales a considerar en este proyecto es la aplicación de medidas de seguridad. La norma ISO 27001 se encarga de evaluar y corregir el cumplimiento de dichas medidas, además de promover la mejora constante mediante un conjunto de controles destinados a mitigar el riesgo de incidentes de seguridad en el funcionamiento del sistema dentro de la institución. Para lograrlo, se consideran diversos tipos de seguridad, entre los cuales se incluyen.

4.3.1. Seguridad lógica

Se han tomado varias medidas para garantizar la seguridad del sistema, que incluyen:

Copias de seguridad automatizadas: Se realizan respaldos automáticos de las bases de datos para asegurar la disponibilidad de la información.

Autenticación segura: Se utiliza un sistema de autenticación que requiere nombres de usuario y contraseñas encriptadas para acceder al sistema, con diferentes niveles de acceso según los privilegios asignados.

Control de sesiones: Se implementan controles para verificar y gestionar las sesiones de usuario, garantizando la seguridad de la información mientras se utiliza el sistema.

Diversos niveles de usuario: Se establecen tres niveles de usuario con distintos niveles de privilegios para controlar el acceso a la información.

Encriptación de datos: Se emplea encriptación para proteger las rutas y contraseñas de los usuarios utilizando funciones de hash.

Resguardo de la base de datos: Se realizan desconexiones automáticas de la base de datos después de las acciones del usuario para evitar problemas con terceros, y se realizan respaldos periódicos de la base de datos para garantizar su integridad.

La seguridad de la información es una prioridad, por lo que se han automatizado procesos como la conexión y desconexión de la base de datos para asegurar su resguardo.

4.3.2. Seguridad Física

Seguridad física y del entorno: Los equipos que almacenan los datos y el sistema se mantienen en lugares seguros, con las condiciones necesarias para operar con el mínimo de riesgos y con controles de acceso adecuados.

Equipamiento: Los servidores que albergan los sistemas de la empresa cuentan con protección física adecuada y se les realiza mantenimiento regular.

Acceso físico: El acceso a las áreas críticas está restringido a personal autorizado, lo que minimiza el riesgo de incidentes fraudulentos y robos de contraseñas.

4.4. PRUEBAS AL SOFTWARE

4.4.1. Prueba de Caja Blanca

En este tipo de prueba, se examina todo el código con el objetivo de evaluar el comportamiento del sistema a través de cada uno de los casos presentados por los algoritmos utilizados en la codificación. Es decir, se trata de casos de prueba aplicados directamente al código fuente. En las pruebas de caja blanca se consideran los siguientes aspectos:

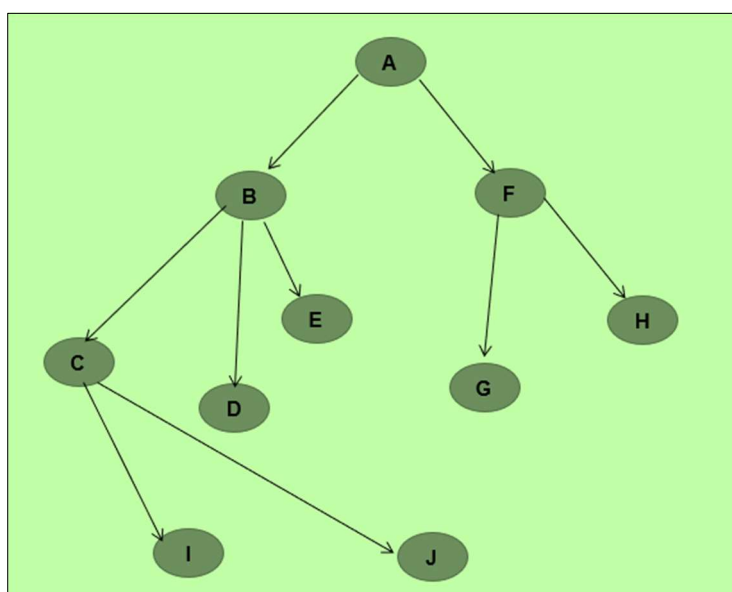
Conocimiento interno: Las pruebas se realizan con un conocimiento detallado del funcionamiento interno del código.

Realización por programadores: Solo los programadores pueden llevar a cabo estas pruebas, dado que requieren un entendimiento profundo del código fuente.

Para aplicar las pruebas de caja blanca, se utiliza la métrica de complejidad ciclométrica, que proporciona una estimación del número de casos de prueba necesarios para el código fuente. Esto asegura una cobertura adecuada de las posibles rutas que puede seguir el programa, permitiendo identificar y corregir errores de manera eficiente

Figura 27

Prueba de caja blanca



Donde:

A	=	Inicio del sistema	F	=	Módulo de reportes
B	=	Módulo de administración de usuarios	G	=	Listado de proyectos
C	=	Creación de usuarios	H	=	Reporte general
D	=	Cambiar de estado	I	=	Reporte específico
E	=	Listado de usuarios	J	=	Corroboración de proyectos

Después de crear el gráfico, el siguiente paso es calcular la complejidad dicromática del grafo. Este cálculo se realiza aplicando una fórmula específica, la cual nos permite determinar el número de rutas independientes a través del código.

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

Donde:

A	=	8 aristas
B	=	7 nodos

Por tanto:

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

$$V(G) = 3$$

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

Camino 1: A – B – C – I

Camino 3: A – B – D

Camino 2: A – B – C – J

Camino 4: A – B – E

Camino 5: A – F – H

Camino 6: A – F – G

Presentar los casos de prueba para garantizar la ejecución de cada camino. Esta condición asegura que se ejecuten ciertos caminos específicos:

- Camino 1: El usuario puede ingresar a la página principal, revisar los módulos de usuarios, ver los reportes e imprimir formularios.
- Camino 2: El usuario puede ver los reportes y verificar el estado de los proyectos.
- Camino 3: El usuario puede ingresar, administrar usuarios y cambiar sus estados.
- Camino 4: El usuario puede ingresar, administrar y listar a los usuarios.
- Camino 5: El usuario puede ingresar y administrar, generando reportes generales de los proyectos.
- Camino 6: El usuario puede ingresar y generar reportes específicos.

4.4.2. Prueba de caja negra

Para realizar las pruebas de caja negra, se utiliza la técnica de partición de equivalencia, una de las más efectivas para analizar tanto los valores válidos como los inválidos de las siguientes interfaces:

Tabla 34

Prueba de acceso al sistema.

CASO DE PRUEBA DE INGRESO AL SISTEMA

Código

Caso de Prueba 01

Descripción de la Prueba	Ingresar al sistema utilizando un nombre de usuario y una contraseña, tras la validación correspondiente.
Condiciones de ejecución	El usuario debe estar previamente registrado.
Pasos de Ejecución	El sistema presenta la pantalla de autenticación, solicitando el nombre de usuario y la contraseña para acceder.
Resultado esperado	El sistema verifica los datos proporcionados y permite al usuario acceder según su rol asignado.
Evaluación de prueba	El acceso al sistema se realiza con una verificación previa; si esta falla, el usuario es redirigido a la página de inicio.

. *Nota.* Aplicación de las métricas de calidad

Tabla 35

Caso de uso de prueba de registro de usuario

CASO DE PRUEBA DE REGISTRO DE USUARIO	
Código	Caso de Prueba 01
Descripción de la prueba	Registro de usuarios
Condiciones de ejecución	El usuario debe haber accedido al sistema mediante su sesión iniciada.
Pasos de Ejecución	<p>El sistema despliega la pantalla destinada a la búsqueda de personas inscritas en la universidad.</p> <p>El usuario localiza a la persona y selecciona la opción de asignar usuario.</p> <p>El sistema presenta un formulario para la creación de un nuevo usuario.</p>

	Luego de validar los datos ingresados, el sistema procede a crear al nuevo usuario.
Resultado esperado	El sistema verifica la información y registra al nuevo usuario.
Evaluación de prueba	Se llevó a cabo la creación de usuario exitosamente.

Nota. Aplicación de las métricas de calidad

4.4.3. Prueba de estrés

Las pruebas de estrés tienen como propósito exponer la plataforma a escenarios que sobrepasen su capacidad habitual de funcionamiento, con la finalidad de detectar posibles vulnerabilidades, defectos o fallos en el sistema.

Estas pruebas permiten evaluar si la plataforma es capaz de manejar niveles altos de usuarios simultáneos, cargas de trabajo intensivas o incrementos repentinos en el tráfico.

En este estudio específico, se empleó un servidor con ciertas modificaciones en el Sistema Operativo para realizar la prueba, los resultados obtenidos reflejan un desempeño óptimo de la plataforma bajo las condiciones desafiantes que se le presentaron.

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

CAPÍTULO V

5. CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

Después de completar el proyecto de grado titulado "SISTEMA DE INFORMACIÓN WEB DE CONTROL Y SEGUIMIENTO PARA LA ADMINISTRACIÓN DE SERVICIOS DE INTERNET Y TV CABLE" para la empresa FENIX, se lograron los objetivos establecidos, cumpliendo con las necesidades requeridas.

A partir de los objetivos planteados, se pueden extraer las siguientes conclusiones:

- Se centralizo una base de datos relacional para evitar la duplicidad de datos y mejorar la organización.
- Se gestiono toda la información de los servicios, áreas de cobertura, vendedores y clientes, optimizando así el servicio ofrecido.
- Se creo perfiles para administradores y clientes, permitiendo a estos últimos ver, modificar o solicitar la cancelación de su servicio.
- Se diseño una página web con una interfaz amigable e intuitiva que muestre toda la información de los servicios y ubicaciones al cliente, lo que podría aumentar el número de clientes potenciales.
- Acelerar la búsqueda de registros y realizar copias de seguridad diarias para tener respaldos y acceso a la información las 24 horas.
- Se genera reportes estadísticos necesarios para que la empresa mejore la atención y optimice el servicio ofrecido a los clientes.

5.2. RECOMENDACIONES

A partir de este trabajo, se proponen las siguientes recomendaciones para mejorar el sistema:

- Realizar copias de seguridad periódicas, garantizará una copia de respaldo disponible en caso de pérdida de datos o fallos del sistema.
- Mantener acceso continuo a internet, es crucial contar con una conexión a internet estable y de alta disponibilidad.
- Mantenimientos periódicos del sistema, es recomendable realizar mantenimientos periódicos al menos cada seis meses.
- La conexión a la facturación electrónica, de acuerdo a los parámetros del Servicio Nacional de Impuestos.

BIBLIOGRAFÍA

- acensTechnologies*. (2016). <https://www.acens.com/wp-content/images/2016/10/bootstrap-framework-acens-wp.pdf>
- Alicante. (18 de octubre de 2021). *Servicio de informatica*. <https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>
- Anonimo. (2011). *Capacitacion y guia para el desarrollo de Software*. [http://materias.fi.uba.ar/7548/PruebasSoftware.pdf]
- Arango Cordoba , J. F. (2016). *Laravel framework para la visualización de variables de entorno en el sector agrícola*. CALI .
- Arimetrics*. (12 de agosto de 2021). <https://www.arimetrics.com/glosario-digital/javascript>
- Asana. (19 de enero de 2013). <https://asana.com/es/resources/project-controls>
- B., G. (11 de ENERO de 2023). *¿QUÉ ES AJAX Y CÓMO FUNCIONA? HOSTINGER TUTORIALES:* <https://www.hostinger.es/tutoriales/que-es-javascript-introduccion-basica/>
- Baena, G. R., Joel Coronado, E., & Mendoza M., R. (05 de 06 de 2019). *Enumed.Net*. Enumed: <http://www.eumed.net/rev/ce/2019/norma-iso-eic.html>
- Baker. (2009).
- Barrera, A. (2021). *next U*. <https://www.nextu.com/blog/que-es-json/>
- Bustos, G. (29 de abril de 2021). *Hostinger Tutoriales*. *¿Qué es Apache? Descripción completa del servidor web Apache:* <https://www.hostinger.es/tutoriales/que-es-apache/>
- Cardenas, O. (26 de mayo de 2016). *google sites*. Ingeniería WEB: <https://sites.google.com/site/ingenieriaweb23/home>

Chavez. (2012). *Ingenieria Web*. 30-50.

Chavez Quispe, J. (2014). *Sistema web de gestión de proyectos*.

COCOMO. (28 de enero de 2013). <https://es.wikipedia.org/wiki/COCOMO>

Contreras Rivas, D. A., & Lara Ávila, B. A. (2015). *Control de presupuesto de obras civiles basado en diseño web orientado a las empresas constructoras*. Guayaquil.

Da Silva, F., & Nuñez, G. (2018). En *La era de las plataformas digitales y el desarrollo de los mercados de datos* (pág. 9). Naciones Unidas: CEPAL.

Desarrollo web. (28 de julio de 2020). <https://desarrolloweb.com/articulos/que-es-mvc.html>

EcuRed. (s.f.). <https://www.ecured.cu/MagicDraw>

Estrada, J. G. (03 de agosto de 2021). *Desarrollo Web con PHP y MySQL*.
<https://cupdf.com/document/desarrollo-web-con-php-y-mysql-52.html>

Flores Fernandez, H. A. (2010). *ARQUITECTURA MULTICAPA MEDIANTE AJAX Y ORM MULTILAYER ARCHITECTURE DRIVEN AJAX AND ORM*. BOGOTA: VINCULOS.

Gardey, J. P. (16 de 05 de 2021). *Definicion.de*. <https://definicion.de/web-2-0/>

Harb Hoecker, R. A. (s.f.). *Mysql my struct Query Lenguage*. Chile.

Instituto de Tecnologia de comumbia Britanica. (19 de septiembre de 2019).
<https://codeigniter.com/userguide3/general/welcome.html>

IONOS. (16 de marzo de 2020). <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/codeigniter-framework-php-rapido-y-versatil/>

ISO, E. (2021). <https://danieltrabajoiso25000.blogspot.com/2020/11/ventajas-y-desventajas-norma-iso-25-000.html>

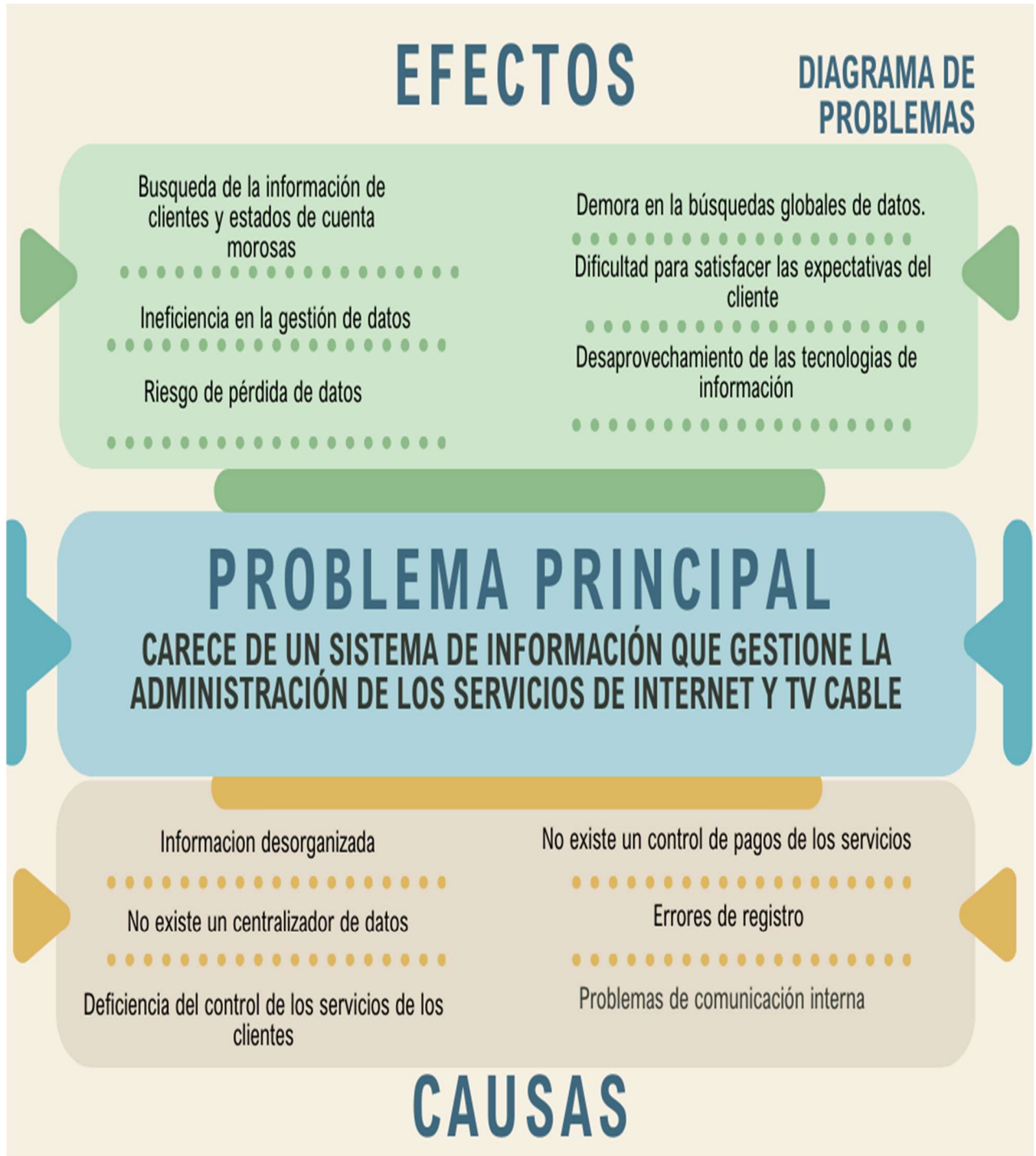
Lopez, J. (2010). *Ingenieria WEB. Ingenieria de software*.

- Lozada, A. (18 de agosto de 2020). *horizonte*. <https://soyhorizonte.com/blog/ventajas-y-desventajas-de-usar-bootstrap/>
- Ludwig, M. (2014). *UWE*. UWE: <https://uwe.pst.ifi.lmu.de/teachingTutorialProcessSpanish.html>
- Magicdraw*. (31 de octubre de 2021). https://www.magicdraw.com/main.php?ts=download_demo&cmd_go_to_login=1&menu=download_demo&back_cmd=cmd_show
- McFarland. (29 de agosto de 2014). <https://www.um.es/docencia/barzana/DAWEB/Lenguaje-de-programacion-JavaScript-1.pdf>
- Miss Juarez , F. G. (2020). *Método Cocomo II*. GUATEMALA.
- MySql.com*. (13 de abril de 2021). <https://www.mysql.com/>
- Neve. (2 de noviembre de 2019). <https://www.losmejoresrecursos.online/proyecto-construccion/>
- Parada, M. (31 de octubre de 2019). *OpenWebinars*. <https://openwebinars.net/blog/que-es-jquery/>
- Platzi*. (9 de septiembre de 2018). <https://platzi.com/blog/pruebas-esenciales-para-evaluar-el-rendimiento-de-software/>
- Porto, J. P., & Gardey, A. (2019). <https://definicion.de/json/>
- Porto, J. P., & Gardey, A. (5 de noviembre de 2019). <https://definicion.de/jquery/>
- Pressman, R. S. (2010). *Ingeniería del Software, Un enfoque práctico*. Mc Graw Hill.
- Ramos Porto, F. (2016). *Sistema web de administración de la ejecución de proyectos de construcción civil para la empresa constructora "Mulrahu SRL"*. La Paz.
- Salguera, H. (2012). Metodología UWE.

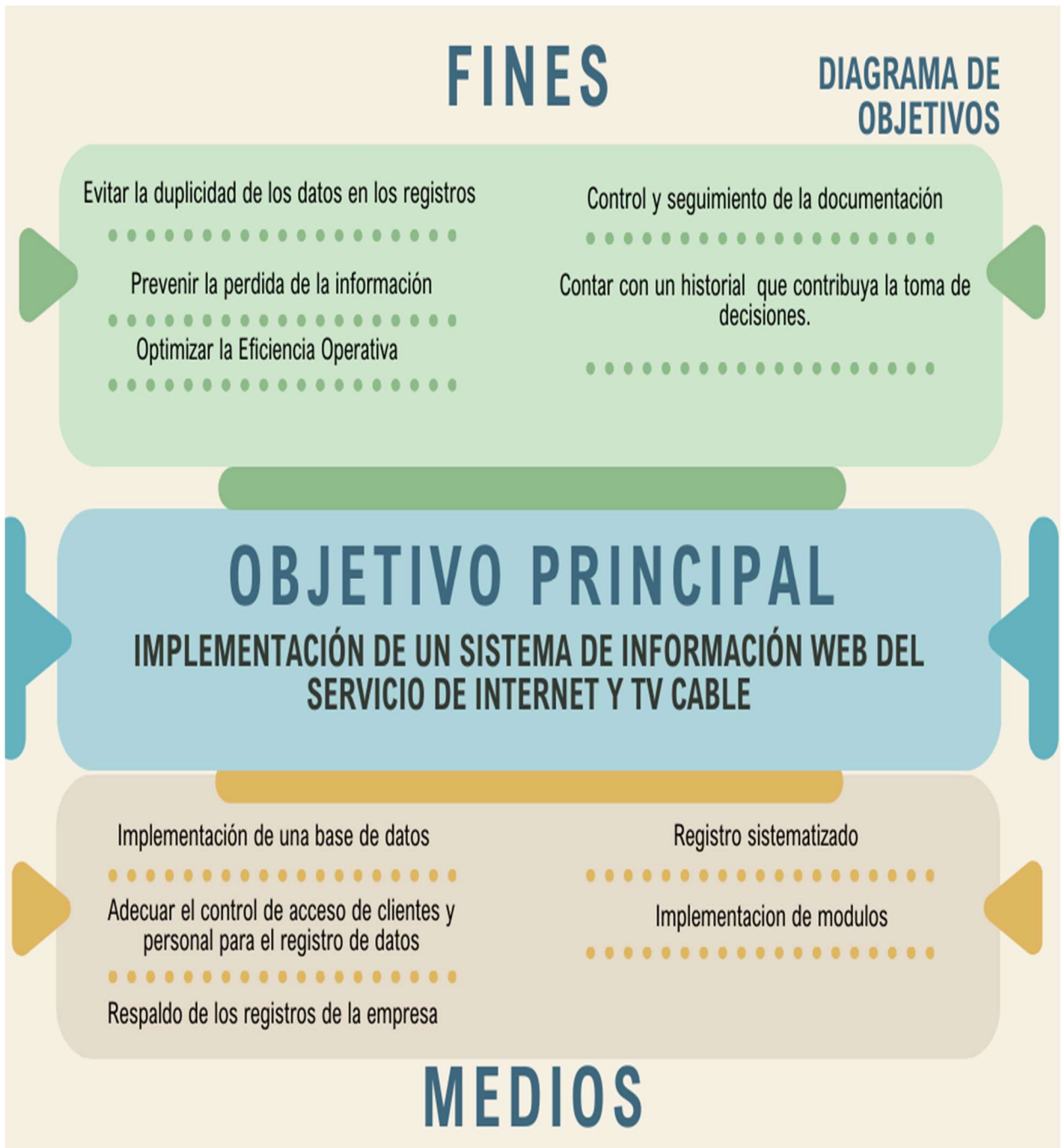
- Sanchez, G. (7 de febrero de 2017). *slideshare*.
<https://es.slideshare.net/GermnSnchezDomnguez/metodologa-uwe-umlbased-web-engineering>
- Sedera. (2016).
https://repositorio.cepal.org/bitstream/handle/11362/47540/1/S2100764_es.pdf
- SlideShare. (06 de junio de 2015). *Metodologia UWE*. <https://es.slideshare.net/omar1023/uwe-49070035>
- Thewolf, D. (25 de junio de 2015). *Metodologia UWE*. <https://metodologiauwe.wordpress.com/>
- Troncoso Contreras, A. P. (2017). *Desarrollo de una aplicación web para la gestión de empresas constructoras*. Valpariso.
- Tunarosa Muñoz, M. Y., & Ávila Bonilla, E. M. (2020). *Sistema de información web para el control y supervisión de obras civiles en el carmen de Apicalá, para la empresa instalaciones hidrosanitarias y de gas Jats S.A.S "Plumbingsoft"*. Colombia.
- UWE, L. –L.-M.-U. (2016). *UWE – UML-based Web Engineering*. LMU – Ludwig-Maximilians-Universität München:
<https://uwe.pst.ifi.lmu.de/teachingTutorialRequirementsSpanish.html>
- Valencia Pavón, E. (2012-2013). *Sistema Académico web utilizando software libre*. QUITO.
- Vargas, M. S. (25 de junio de 2017). *Ingeniería basada en modelos*.
<http://marcelosalasvargas.blogspot.com/2017>
- Wikipedia*. (31 de octubre de 2021). https://es.wikipedia.org/wiki/Web_2.0

ANEXOS

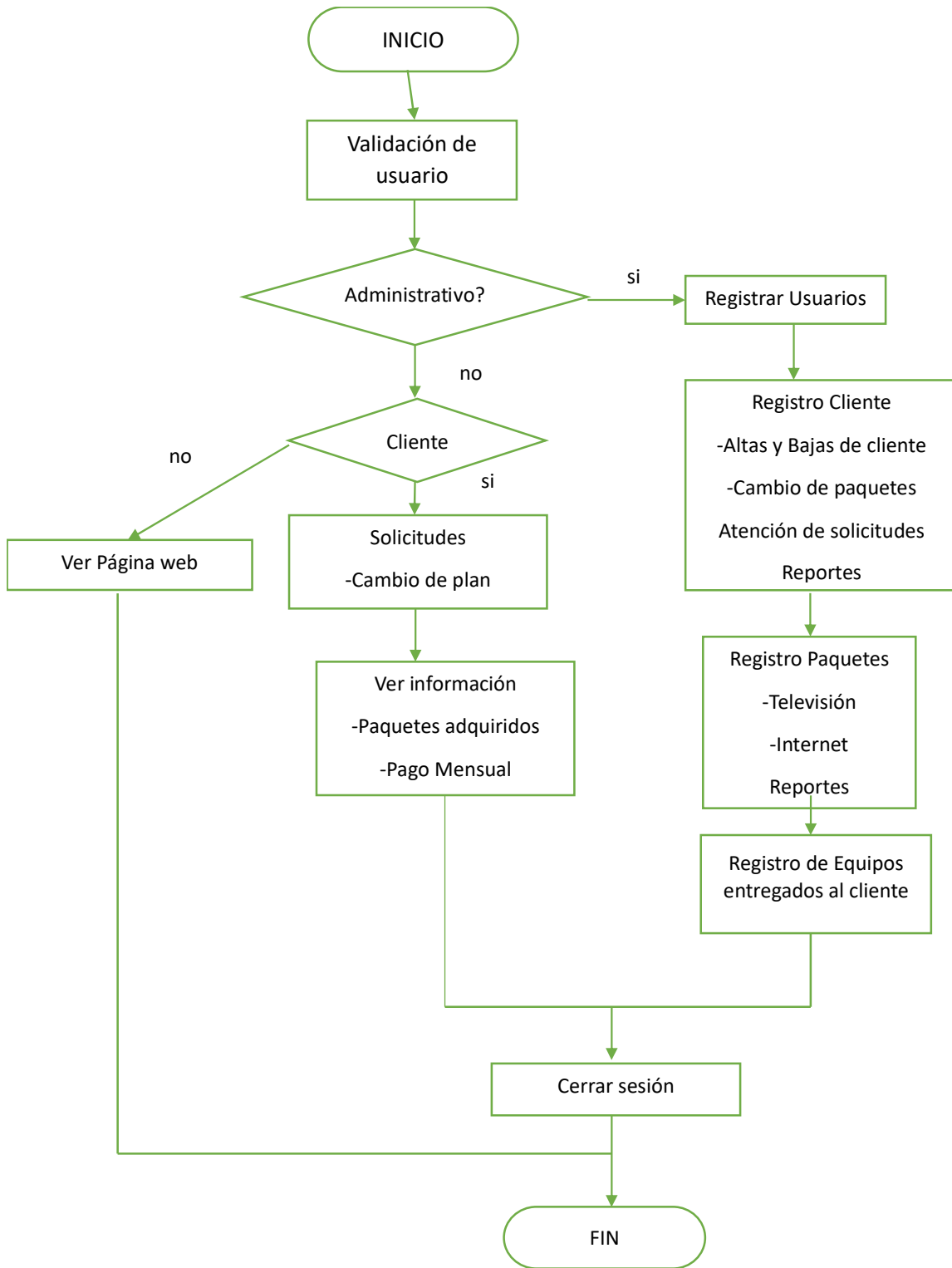
ANEXO A. ARBOL DE PROBLEMAS




ARBOL B. ARBOL DE OBJETIVOS



ANEXO C. DIAGRAMA DE FLUJO



A dark blue vertical bar is on the left side of the page. A blue arrow points from the bar towards the right, containing the title text.

*“SISTEMA DE INFORMACIÓN WEB
DE CONTROL Y SEGUIMIENTO
PARA LA ADMINISTRACIÓN DE
SERVICIOS DE INTERNET Y TV
CABLE”*

MANUAL DE USUARIO

Several thin, curved lines in shades of blue and grey originate from the bottom left corner and extend upwards and to the right.

JHONNY CHOQUE CONDORI

1. Introducción

El presente documento pretende servir de guía para el uso adecuado del sistema, ya que con esta plataforma se pretende realizar el registro adecuado de los proyectos.

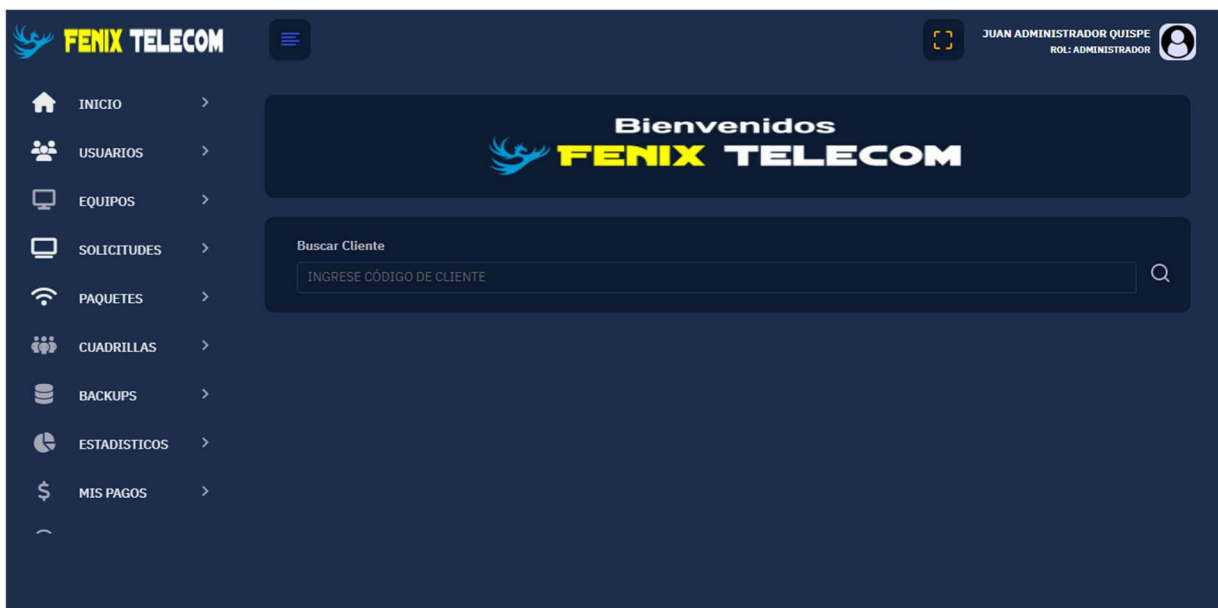
2. Objetivo del sistema

El objetivo del manual, es permitir al usuario final, que pueda tener la administración adecuada del SISTEMA DE INFORMACIÓN WEB DE CONTROL Y SEGUIMIENTO PARA LA ADMINISTRACIÓN DE SERVICIOS DE INTERNET Y TV CABLE.

3. Guía del sistema

4. Inicio

Después de haber realizado el inicio de sesión nos muestra la página de inicio para los roles de administrador y plataforma, donde se puede observar el menú del sistema y el buscador de clientes correspondiente.



Modulo Usuarios

En el menú de listado de administrativos se podrá visualizar la lista de los administrativos su foto el cargo su cuenta de correo y el estado en el que se encuentra como se ve en la siguiente imagen:

The screenshot shows the 'LISTA DE ADMINISTRATIVOS' page in the Fenix Telecom system. The interface includes a sidebar with navigation options like 'INICIO', 'USUARIOS', 'EQUIPOS', 'SOLICITUDES', 'PAQUETES', 'CUADRILLAS', 'BACKUPS', 'ESTADISTICOS', and 'PAGOS CLIENTES'. The main content area displays a table of administrators with columns for '#', 'FOTO', 'ADMINISTRATIVO', 'CARGO', 'ROL', 'ESTADO', and 'OPCIONES'. Three records are visible, each with an 'Actualizar' button and a lock icon. The user 'JUAN ADMINISTRADOR QUISPE' is logged in as 'ROL: ADMINISTRADOR'.

#	FOTO	ADMINISTRATIVO	CARGO	ROL	ESTADO	OPCIONES
1		FIDEL FLORES FLORES	SUPERVISOR PLATAFORMA	ADMINISTRADOR	ACTIVO	Actualizar
2		DELIA QUISPE ALI	JEFE PLATAFORMA	PLATAFORMA	ACTIVO	Actualizar
3		JUAN CARLOS GUTIERREZ MELENDRIZ	TÉCNICO 2	ADMINISTRADOR	ACTIVO	Actualizar

Para poder editar los datos de los administrativos dónde se realiza las siguientes acciones:

The 'Editar Administrativo' form is a modal window for editing user data. It features a profile picture placeholder and a file selection button. The form contains several input fields for personal and professional information, including names, C.I., role, email, cargo, and phone number. There are also fields for 'Apellido Paterno', 'Apellido Materno', 'Expedido', 'Complemento', 'Corporativo', and 'Hoja de vida'. The form is ready to be submitted by clicking the 'Registrar' button.

El agregar un nuevo administrativo nos despliega la siguiente ventana donde se llenan los siguientes campos

Agregar administrativo

Seleccionar archivo Ningún archivo seleccionado

Nombres * **Apellido Paterno *** **Apellido Materno ***

C.I. * **Expedido *** **Complemento**

Rol **Email *** **Corporativo**

Telefono * **Cargo *** **Hoja de vida (15MB max.)**

Seleccionar archivo Ningún ...ccionado

Cerrar X **Registrar**

Listados de técnicos: El listado de profesionales nos mostrará con la siguiente pantalla donde se visualizará los siguientes campos:

FENIX TELECOM JUAN ADMINISTRADOR QUISPE ROL: ADMINISTRADOR

LISTA DE TÉCNICOS DE CUADRILLA

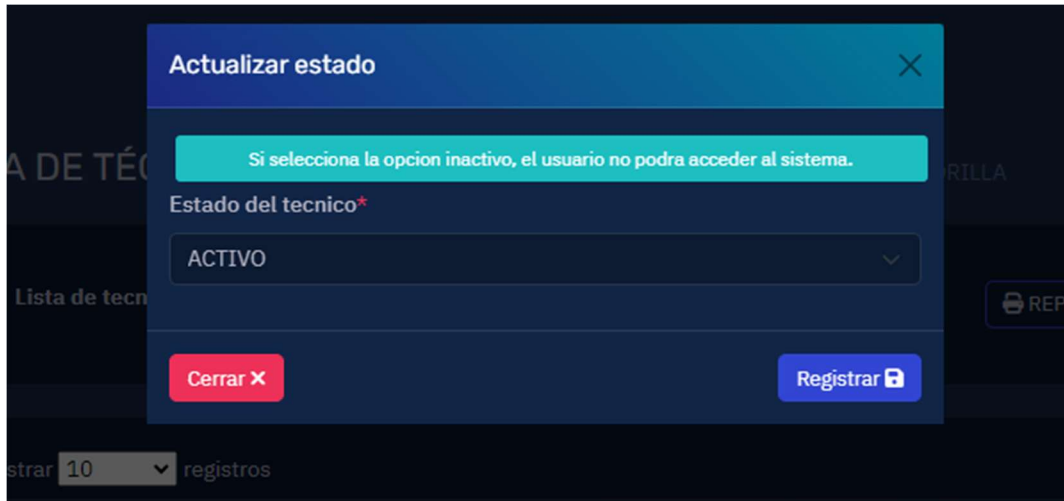
Lista de técnicos **REPORTE PDF** **+ REGISTRAR TECNICO**

Mostrar 10 registros Buscar:

#	FOTO	TÉCNICO	CARGO	ROL	ESTADO	OPCIONES
1		TANIA CHAMBI MAYTA	TÉCNICO CUADRILLA	CUADRILLA	ACTIVO	Actualizar
2		DELMER FLORES CUSI	TÉCNICO CUADRILLA	CUADRILLA	ACTIVO	Actualizar

Mostrando 1 a 2 de 2 registros Anterior **1** Siguiente

En la siguiente pantalla mostrará la ventana de actualizar el estado del profesional de la siguiente manera:



Opción de cambiar el estado entre activo e inactivo de un técnico, para dar de baja de alta o baja del sistema.



El rol de responsable puede visualizar las solicitudes asignadas por plataforma, de acuerdo a esto se realiza la instalación y a través del sistema, el responsable podrá cambiar el estado a solicitud atendida.

Mis solicitudes [Mis solicitudes](#)

Lista de mis solicitudes REPORTE PDF

Mostrar 10 registros Buscar:

#	CLIENTE	CIUDAD	DIRECCIÓN	ESTADO	OPCIONES
1	FIDEL CLIENTE QUISPE	Abel Iturralde	Zona: VILLA BOLIVAR Avenida: HACIA EL MAR No. 8788	PENDIENTE	Actualizar

Mostrando 1 a 1 de 1 registros Anterior 1 Siguiente

A través del rol administrador, se podrá crear las cuadrillas y agregar técnicos y designar un responsable de cuadrillas.

LISTA DE CUADRILLAS [LISTA DE CUADRILLAS](#)

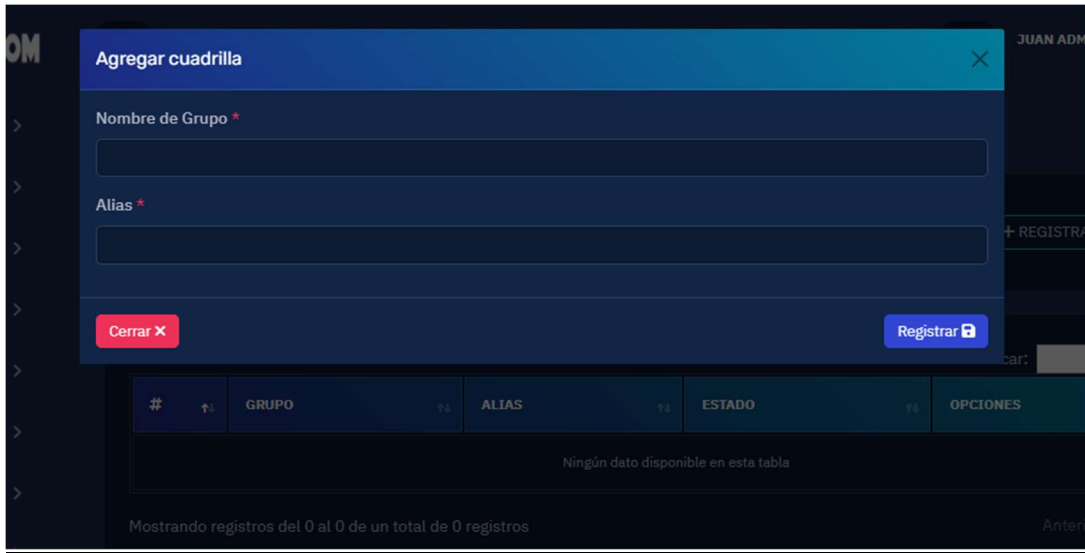
Lista de cuadrillas REPORTE PDF + REGISTRAR CUADRILLA

Mostrar 10 registros Buscar:

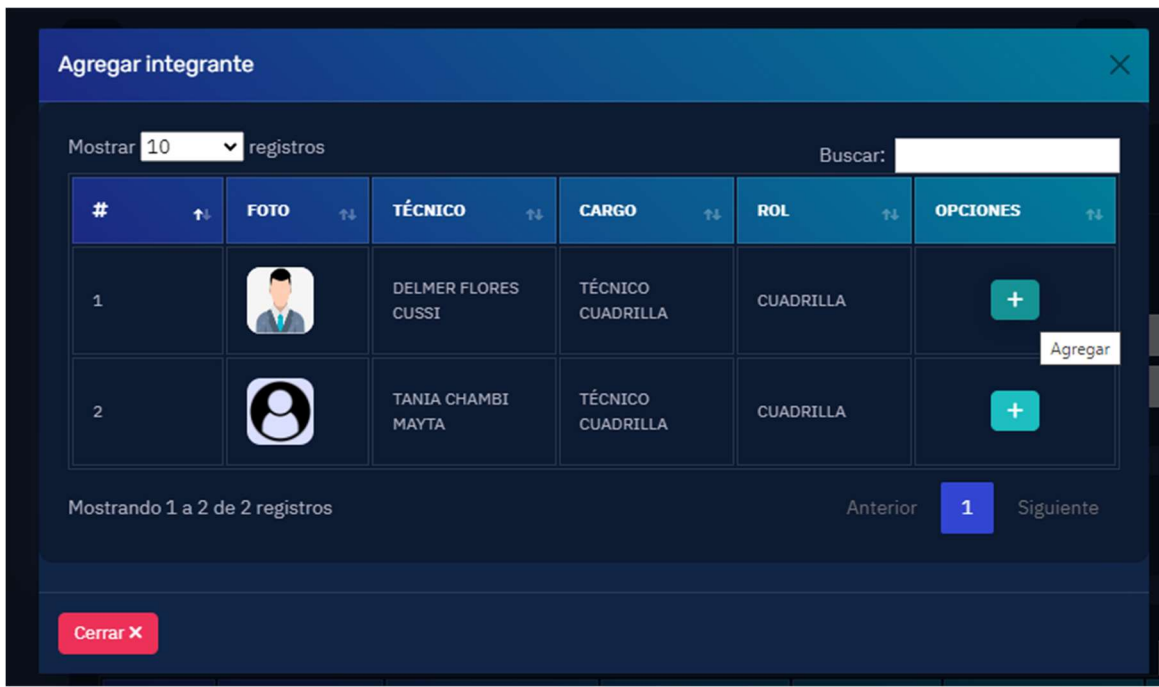
#	GRUPO	ALIAS	ESTADO	OPCIONES
1	CUADRILLA 1	ALFA	ACTIVO	Actualizar

Mostrando 1 a 1 de 1 registros Anterior 1 Siguiente

A continuación, se muestra en la siguiente imagen, el registro de una cuadrilla, y el llenado de los campos correspondientes.



Una vez creada la cuadrilla como administrador, se podrá agregar técnicos a la cuadrilla correspondiente.



Lista de Integrantes + NUEVO INTEGRANTE

Mostrar 10 registros Buscar:

#	FOTO	TÉCNICO	CARGO	ROL	ESTADO	OPCIONES
1		TANIA CHAMBI MAYTA	TÉCNICO CUADRILLA	CUADRILLA	ACTIVO	
2		DELMER FLORES CUSSI	TÉCNICO CUADRILLA	CUADRILLA	ACTIVO	

Mostrando 1 a 2 de 2 registros Anterior **1** Siguiente

Ya agregados los técnicos, podemos asignar un responsable y desasignar para cambiar a otro responsable.

Lista de Integrantes + NUEVO INTEGRANTE

Mostrar 10 registros Buscar:

#	FOTO	TÉCNICO	CARGO	ROL	ESTADO	OPCIONES
1		TANIA CHAMBI MAYTA	TÉCNICO CUADRILLA	RESPONSABLE	ACTIVO	
2		DELMER FLORES CUSSI	TÉCNICO CUADRILLA	CUADRILLA	ACTIVO	

Mostrando 1 a 2 de 2 registros Anterior **1** Siguiente

5. Clientes

En el menú usuarios también nos mostrará la lista de los clientes con los siguientes datos:

LISTA DE CLIENTES - LISTA DE CLIENTES

Lista de Clientes REPORTE PDF

Mostrar 10 registros Buscar:

#	CODIGO	CLIENTE	CORREO	TELEFONO	ESTADO	OPCIONES
1	FEN-000018	DELIA FLORES MELENDEZ	delia@gmail.com	76666466	ACTIVO	Actualizar

Mostrando 1 a 1 de 1 registros Anterior **1** Siguiente

Registro de las solicitudes de instalación, campos muy necesarios de los que posteriormente realizada la instalación, serán clientes.

The screenshot shows a web application interface for 'REGISTRO DE SOLICITUD'. The user is logged in as 'JUAN ADMINISTRADOR QUISPE' with the role 'ROL: ADMINISTRADOR'. The page title is 'REGISTRO DE SOLICITUD' and the breadcrumb is 'REGISTRO DE SOLICITUD'. The main heading is 'Solicitud' with the instruction 'Registre los datos que solicite el formulario.' Below this, there are five steps: '1 Datos Personales', '2 Datos de Contacto', '3 Ubicación', '4 Instalación', and '5 Verificación'. The '1 Datos Personales' step is active. The form fields are: 'Nombres *' (BRISA), 'Apellido Paterno *' (FERNANDEZ), 'Apellido Materno *' (MITA), 'C.I. *' (675464566), 'Expedido *' (LP), 'Complemento', and 'Sexo *' (FEMENINO). At the bottom right, there are 'Anterior' and 'Siguiente' buttons.

Se realiza la acción de editar cliente en los cuales los siguientes campos se podrán editar:

The screenshot shows the 'Solicitud' form in an 'EDITAR' view. The user is logged in as 'JUAN ADMINISTRADOR QUISPE' with the role 'ROL: ADMINISTRADOR'. The page title is 'Solicitud' and the breadcrumb is 'REGISTRO DE SOLICITUD'. The main heading is 'Solicitud' with the instruction 'Registre los datos que solicite el formulario.' Below this, there are five steps: '1 Datos Personales', '2 Datos de Contacto', '3 Ubicación', '4 Instalación', and '5 Verificación'. The '1 Datos Personales' step is active. The form fields are: 'Nombres y Apellidos *' (BRISA FERNANDEZ MITA), 'C.I.' (675464566 LP), 'Sexo *' (FEMENINO), 'Celular *' (76565465), 'Email *' (nico@gmail.com), 'Contacto referencia *' (JHONNY CHOQUE), 'Número Contacto *' (76565443), 'Parentesco *' (AMIGO), 'Departamento *' (La Paz), 'Provincia' (Murillo), 'Localidad *' (EL ALTO), 'Dirección' (Zona RIO SECO, Calle HACIA EL MAR, N°), and 'Paquete *' (TV E INTERNET - COMBO PLUS - Tarifa: 25.7 Bs.). At the bottom right, there are 'Anterior' and 'Finalizar' buttons.

También se tendrán las listas de solicitudes, visualizados por el rol de plataforma y administrador.

Lista de Solicitudes

Mostrar 10 registros

Buscar:

#	CLIENTE	CIUDAD	DIRECCIÓN	ESTADO	OPCIONES
1	FIDEL CLIENTE QUISPE	Abel Iturralde	Zona: VILLA BOLIVAR Avenida: HACIA EL MAR No. 8788	PENDIENTE	

Mostrando 1 a 1 de 1 registros

Anterior 1 Siguiente

Visualización de los datos del cliente.

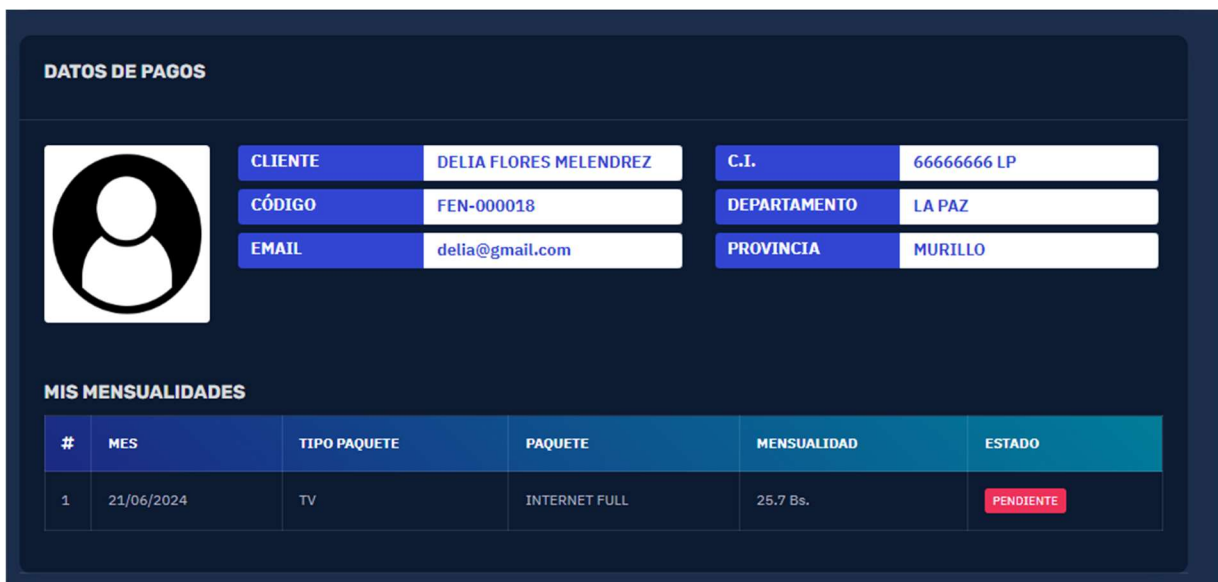
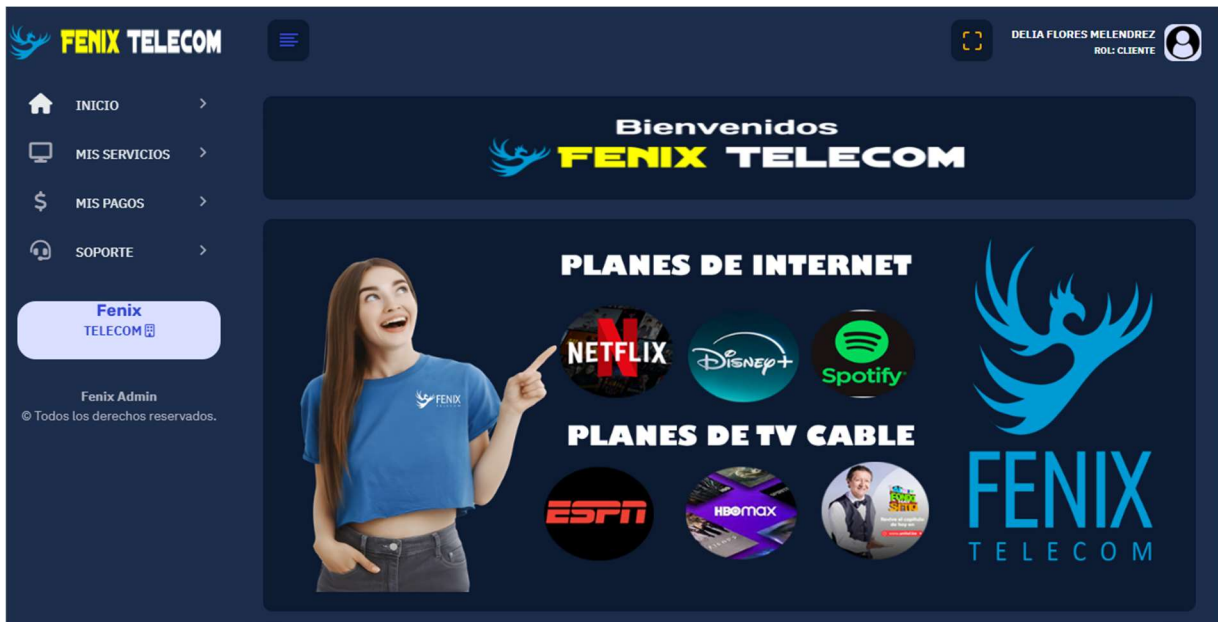
Datos del Cliente

	CLIENTE	DELIA FLORES MELENDREZ	C.I.	66666666 LP
	CÓDIGO	FEN-000018	DEPARTAMENTO	LA PAZ
	SEXO	FEMENINO	PROVINCIA	MURILLO
	LOCALIDAD	EL ALTO	ZONA	RIO SECO
	AVENIDA/CALLE	AZURDUY	# CASA	647
	EMAIL	delia@gmail.com		

PAQUETE ADQUIRIDOS

#	TIPO PAQUETE	PAQUETE	MENSUALIDAD	ESTADO
1	INTERNET	INTERNET FULL	25.7	ACTIVO

El cliente una vez dado de alta, podrá ingresar al sistema con su correo electrónico como usuario y como password el ci, recomendándose cambiar la contraseña.







6. Equipos

En esta sección del sistema, el usuario podrá ver la lista de todas los equipos que un cliente puede tener, además que se visualiza las opciones de crear, editar y eliminar.

LISTA DE EQUIPOS [-](#) LISTA DE EQUIPOS

Lista de equipos REPORTE PDF + REGISTRAR EQUIPO

Mostrar 10 registros Buscar:

#	EQUIPO	TIPO DE EQUIPO	ESTADO DE EQUIPO	ESTADO	OPCIONES
1	ROUTER MARCA: HUITRON MODELO: Y-8888-00-E SERIE: 67564	INTERNET	SIN ASIGNAR	ACTIVO	Actualizar  
2	DECODIFICADOR MARCA: HUIION MODELO: RTH-899 SERIE: 545454	TV CABLE	SIN ASIGNAR	ACTIVO	Actualizar  

Mostrando 1 a 2 de 2 registros Anterior 1 Siguiente

En la opción de editar, tenemos la opción de actualizar los datos de equipo de acuerdo al dato seleccionado.

Editar equipo ✕

Nombre *
ROUTER

TIPO DE EQUIPO *
Internet

MARCA *
HUITRON

MODELO *
Y-8888-00-E

SERIE *
67564

OBSERVACION *
S/O

Cerrar ✕ Registrar 

7. Soporte

El cliente puede realizar la petición de soporte técnico, que será derivado por plataforma a una caudrilla.

Lista de Asistencias Técnicas + REGISTRAR ASISTENCIA TÉCNICA

Mostrar 10 registros Buscar:

#	FECHA	TIPO	DESCRIPCIÓN	SOLUCIÓN	ESTADO
1	2024-06-21	TV CABLE	MI EQUIPO DE INTERNET NO MANDA LOS DATOS PARA NAVEGAR CORRECTAMENTE.	NO ATENDIDO	SOLICITUD <small>Fecha: 2024-06-21</small>

Mostrando 1 a 1 de 1 registros Anterior 1 Siguiente

Una vez designado por plataforma se dará la atención correspondiente a la solicitud de soporte.

LISTA DE SOLICITUDES DE ASISTENCIAS TÉCNICAS 🏠 - LISTA DE SOLICITUDES DE ASISTENCIAS TÉCNICAS

Lista de Solicitudes Asistencias Técnicas

Mostrar 10 registros Buscar:

#	FECHA	CLIENTE	CUADRILLA	TIPO	DESCRIPCIÓN	SOLUCIÓN	ESTADO	OPCIONES
1	2024-06-21	FEN-000018	SIN ASIGNAR	TV CABLE	MI EQUIPO DE INTERNET NO MANDA LOS DATOS PARA NAVEGAR CORRECTAMENTE.	NO ATENDIDO	SOLICITUD <small>Fecha: 2024-06-21</small>	Asignar

Mostrando 1 a 1 de 1 registros Anterior 1 Siguiente

Actualizar los estados, una vez atendidas las solicitudes.

LISTA DE SOLICITUDES DE ASISTENCIAS TECNICAS [-](#) LISTA DE SOLICITUDES DE ASISTENCIAS TECNICAS

Lista de Solicitudes Asistencias Tecnicas

Mostrar 10 registros Buscar:

#	FECHA	CLIENTE	CUADRILLA	TIPO	DESCRIPCIÓN	SOLUCIÓN	ESTADO	OPCIONES
1	2024-06-21	FEN-000018	CUADRILLA 1	TV CABLE	MI EQUIPO DE INTERNET NO MANDA LOS DATOS PARA NAVEGAR CORRECTAMENTE.	NO ATENDIDO	EN ESPERA Fecha: 2024-06-21	Asignar REPORTE PDF

Mostrando 1 a 1 de 1 registros Anterior **1** Siguiente

8. Backups

Visualización de las copias de base de datos del sistema.

Lista de Backups [-](#) Lista de Backups

Lista de backups [+ REALIZAR BACKUPS](#)

Mostrar 10 registros Buscar:

#	NOMBRE DE ARCHIVO	TAMAÑO	FECHA	TRANSCURRIDO	OPCIONES
1	fenix/2024-06-13-18-48-31.zip	6 KB	2024-06-13 22:48:34	hace 1 semana	
2	fenix/2024-06-12-14-56-41.zip	6 KB	2024-06-12 18:56:43	hace 1 semana	
3	fenix/2024-05-20-20-52-07.zip	5 KB	2024-05-21 00:52:08	hace 1 mes	
4	fenix/2024-05-20-20-11-00.zip	5 KB	2024-05-21 00:11:02	hace 1 mes	

Mostrando 1 a 4 de 4 registros Anterior **1** Siguiente

9. Estadísticas

A continuación, se muestra los reportes estadísticos del sistema.

Bienvenidos

FENIX TELECOM



4

ADMINISTRATIVOS



2

TÉCNICOS CUADRILLA



0

SOLICITUDES

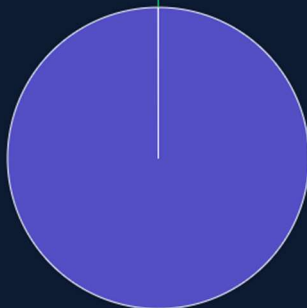


1

CLIENTES

ESTADISTICO DE PAQUETES.

COMBO TV E INTERNET 0.0% INTERNET 0.0%



TV 100.0%

Descripción

Highcharts.com



SERVICIO POR PROVINCIAS

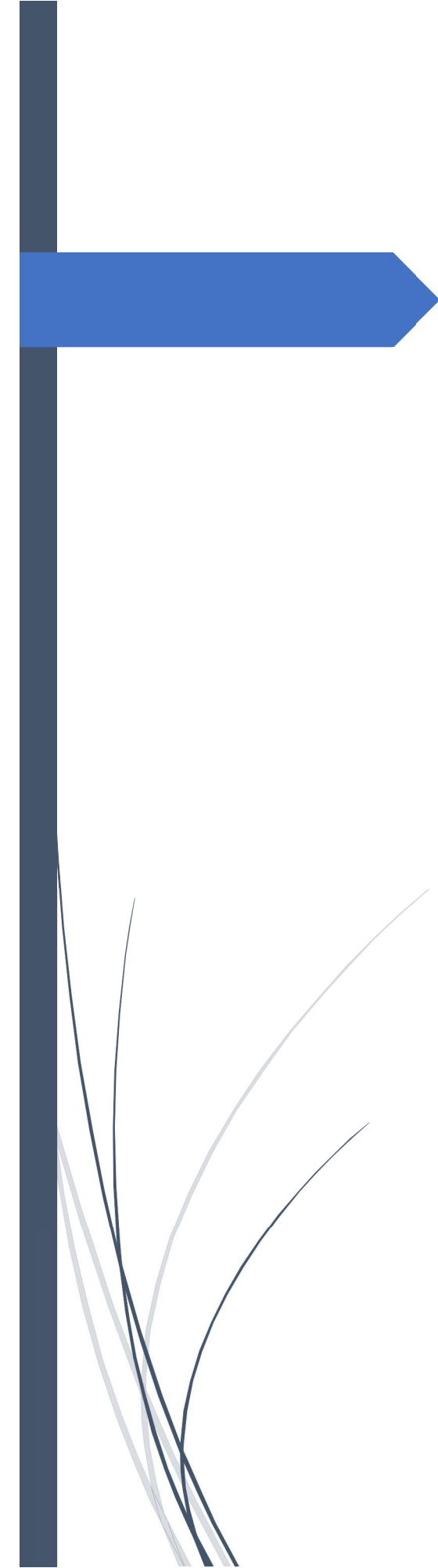
Estadístico de clientes



Column widths are proportional to GDP

Highcharts.com

Descripción



*“SISTEMA DE INFORMACIÓN WEB
DE CONTROL Y SEGUIMIENTO
PARA LA ADMINISTRACIÓN DE
SERVICIOS DE INTERNET Y TV
CABLE”*

MANUAL TÉCNICO

JHONNY CHOQUE CONDORI

10. PRINCIPALES REQUERIMIENTOS

1.1 Requerimientos Funcionales

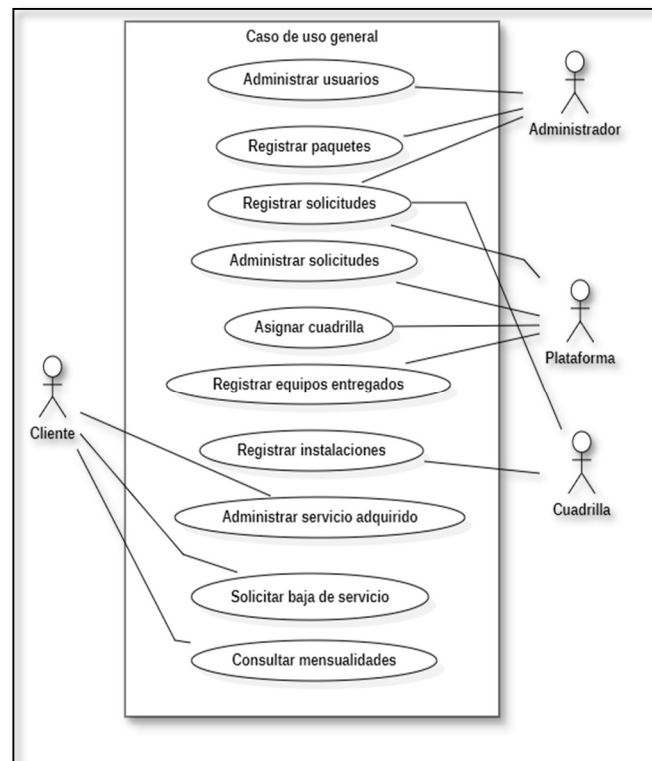
- Acceso al Sistema mediante verificación de datos de acceso.
- Administrar los roles y permisos de acuerdo a los usuarios registrados.
- Administración de Administrativos.
- Administración de Profesionales.
- Administración de Proyectos.
- Reporte Estadístico.
- Respaldos de Base de Datos.

1.2 Requerimientos No Funcionales

- Sistema Compatible con los diferentes navegadores.
- Interfaz gráfica debe responder al usuario cuando lo requiera
- Transacciones deben responder las necesidades de los usuarios.
- Soporte y Mantenimiento, para disponibilidad del sistema.

2 DISEÑO DEL SISTEMA

2.1 Caso de Uso General



3 BASE DE DATOS

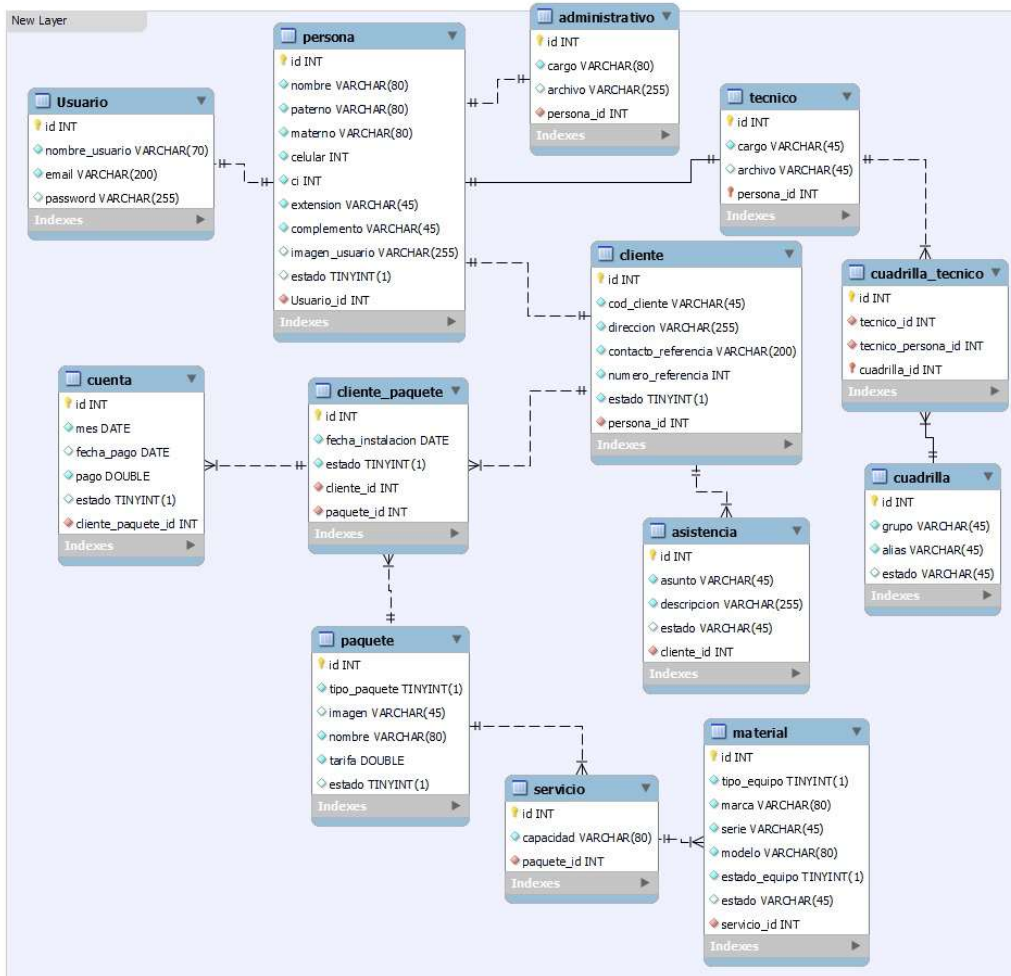
3.1 Modelo de Base de Datos Relacional

4 HERRAMIENTA DE DESARROLLO

Las herramientas para el desarrollo del sistema serán los siguientes:

➤ **Hypertext Pre.Procesor “PHP”**

PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.



➤ **MySQL**

MySQL es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual: Licencia pública general/Licencia comercial por Oracle Corporation.

➤ **Laravel**

Es un framework de código abierto para desarrollar aplicaciones y servicios web con PHP, basado en el modelo vista controlador, intenta aprovechar lo mejor de otros frameworks y aprovechar las características de las últimas versiones de PHP

5 REQUERIMIENTO E INSTALACIÓN

5.1 Requerimientos

A continuación, se lista los requerimientos del sistema para la implementación:

- **PHP versión mínima 8.1 para arriba.**
- **Servidor Apache instalado.**
- **Acceso a Terminal para ejecución de comandos.**
- **Composer.**
- **Node JS**

El sistema cuenta con un flujo considerable de archivos pdf e imágenes por lo que se recomienda tener la capacidad de acuerdo a la siguiente especificación:

- **50 GB de almacenamiento como mínimo.**