



UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE QUITO

CARRERA DE COMPUTACIÓN

**SISTEMA INFORMÁTICO WEB PARA GESTIONAR LA RECAUDACIÓN DEL
SERVICIO DE INTERNET QUE BRINDA LA EMPRESA QUICKNET**

Trabajo de Titulación previo a la obtención del
Título de Ingeniero en Ciencias de la Computación

AUTOR: BORIS MAURICIO TAPIA FUENMAYOR

TUTOR: LINA PATRICIA ZAPATA MOLINA

Quito – Ecuador

2022

**CERTIFICADO DE RESPONSABILIDAD Y AUTORIA DEL TRABAJO DE
TITULACION**

Yo, Boris Mauricio Tapia Fuenmayor con documento de identificación No 1720504560;
manifestamos que:

Soy el autor y responsable del presente trabajo; y, autorizo a que sin fines de lucro la
Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o
parcial el presente trabajo de titulación.

Quito, 14 de septiembre del año 2022

Atentamente,



Boris Mauricio Tapia Fuenmayor

1720504560

**CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE
TITULACION A LA UNIVERSIDAD POLITECNICA SALESIANA**

Yo, Tapia Fuenmayor Boris Mauricio con documento de identificación No 1720504560, expreso la voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autor del proyecto técnico : “Sistema Informático Web Para Gestionar la Recaudación del Servicio de Internet que Brinda la Empresa Quicknet”, el cual ha sido desarrollado para optar por el título de: INGENIERO EN CIENCIAS DE LA COMPUTACIÓN, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana

Quito, 14 de septiembre del año 2022

Atentamente,



Boris Mauricio Tapia Fuenmayor

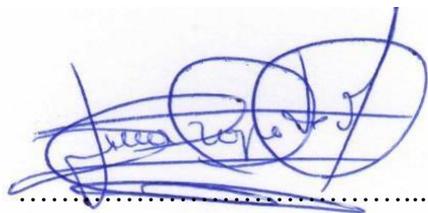
1720504560

CERTIFICADO DE DIRECCION DEL TRABAJO DE TITULACION

Yo Lina Patricia Zapata Molina con documento de identidad N° 0501877278, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: SISTEMA INFORMÁTICO WEB PARA GESTIONAR LA RECAUDACIÓN DEL SERVICIO DE INTERNET QUE BRINDA LA EMPRESA QUICKNET realizado por Boris Mauricio Tapia Fuenmayor con documento de identificación N° 1720504560, obteniendo como resultado final el trabajo de titulación bajo la opción Proyecto Técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Quito, 14 de septiembre del año 2022

Atentamente



ING. Lina Patricia Zapata Molina, PhD.

0501877278

DEDICATORIA

Este proyecto técnico dedico a mis dos abuelitos que están en el cielo a mamita Fanny y papito Oncho, fueron unas personas que me demostraron que en la vida no hay que rendirse y seguir adelante a pesar de las adversidades.

A mi papá Boris Tapia y mamá María Erminia Fuenmayor por su gran ejemplo, constancia y amor que me han demostrado para seguir adelante y a mi abuelita Cecilia Balseca por su apoyo incondicional.

A María José Robles por su gran compañía y apoyo en el transcurso de estos últimos años de mi carrera, a mis amigos y familiares que creyeron en mí y nunca dudaron que lo iba a lograr

AGRADECIMIENTOS

Agradezco a Dios por darme la oportunidad de terminar mi carrera, a mi padre Boris Vinicio Tapia por brindarme el estudio universitario y su apoyo en todo momento, a mi madre María Erminia Fuenmayor por su amor, cariño, fortaleza y ejemplo para salir adelante.

Agradezco al ingeniero Gustavo Tapia por darme la oportunidad de desarrollar este proyecto técnico para su empresa.

Agradezco a todos los profesores y amigos por haber compartido sus enseñanzas y amistad en el transcurso de estos años.

Agradezco a mis compañeros de trabajo por sus enseñanzas día a día para ser un mejor profesional.

Agradezco a José Max Tapia por su apoyo en el transcurso de mi carrera universitaria.

Contenido

RESUMEN	16
ABSTRACT.....	17
1. INTRODUCCIÓN	18
1.1. PROBLEMÁTICA	19
1.2. JUSTIFICACIÓN	20
1.3. OBJETIVOS	21
1.3.1. OBJETIVO GENERAL:.....	21
1.4. METODOLOGÍA	21
2. MARCO TEORICO.....	22
2.1. ANTECEDENTES HISTÓRICOS	22
2.2. FUNDAMENTOS TEORICOS	24
2.2.1. SISTEMA INFORMÁTICO WEB	24
2.2.2. IMPORTANCIA DE UN SISTEMA INFORMATICO WEB.....	24
2.2.3. HTML5.....	25
2.2.4. ESTRUCTURA BASICA DE UN DOCUMENTO HTML5.....	25
2.2.5. CSS.....	26
2.2.6. RESPONSIVE DESING	26
2.2.7. GESTION DE COBROS	27
2.2.8. MYSQL.....	28
2.2.9. DJANGO.....	28
2.2.10. FRAMEWORK	29
2.2.11. JAVASCRIPT	29
2.2.12. BOOTSTRAP.....	29
2.2.13. VISUAL STUDIO CODE	30
3. ANALISIS Y DISEÑO	30
3.1. SITUACIÓN ACTUAL.....	30

3.2.	REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES	30
3.2.1.	<i>Requerimientos Funcionales</i>	30
3.2.2.	REQUERIMIENTOS NO FUNCIONALES	32
3.3.	DISEÑO DEL SISTEMA	33
3.3.1.	DESCRIPCIÓN DE ACTORES.....	33
3.3.2.	DESCRIPCIÓN DE LAS TAREAS	34
3.3.3.	DIAGRAMAS DE CASO DE USO	34
3.3.3.1.	CASO DE USO SISTEMA DE COBROS.....	35
3.3.3.2.	CASO DE USO: INGRESO AL SISTEMA.....	35
3.3.3.3.	ESPECIFICACIÓN DEL CASO DE USO: INGRESO AL SISTEMA.....	36
3.3.3.4.	CASO DE USO: ADMINISTRACIÓN DE USUARIOS	36
3.3.3.5.	ESPECIFICACIÓN DEL CASO DE USO: ADMINISTRACIÓN DE USUARIOS	37
3.3.3.6.	CASO DE USO: ADMINISTRACIÓN DE CLIENTES	38
3.3.3.7.	ESPECIFICACIÓN DEL CASO DE USO: ADMINISTRACIÓN DE CLIENTES	38
3.3.3.8.	CASO DE USO: ADMINISTRACIÓN DE CONTRATOS	39
3.3.3.9.	ESPECIFICACIÓN DEL CASO DE USO: ADMINISTRACIÓN DE CONTRATO	39
3.3.3.10.	CASO DE USO: ADMINISTRACIÓN DE ORDENES DE INSTALACIÓN	40
3.3.3.11.	ESPECIFICACIÓN DEL CASO DE USO: ADMINISTRACIÓN DE ORDENES DE	
	INSTALACIÓN 40	
3.3.3.18.	CASO DE USO: ADMINISTRACIÓN DE PROVEEDOR	45
3.4.	DIAGRAMA DE SECUENCIAS	49
3.4.1.	DIAGRAMA DE SECUENCIA: INICIO DE SESIÓN	49
3.4.2.	DIAGRAMA DE SECUENCIA: CREACIÓN DE USUARIO.....	50
3.4.3.	DIAGRAMA DE SECUENCIA: CREACIÓN DE CLIENTES Y CONTRATOS	51
3.4.4.	DIAGRAMA DE SECUENCIA: MODIFICACIÓN DE CONTRATOS.....	51
3.4.5.	DIAGRAMA DE SECUENCIA: REGISTRO DE MATERIALES.....	52
3.4.6.	DIAGRAMA DE SECUENCIA: REGISTRO DE PLANES	53
3.4.7.	DIAGRAMA DE SECUENCIA: REGISTRO DE COMPRAS.....	54
3.4.8.	DIAGRAMA DE SECUENCIA: IMPRESIÓN DE RECIBO DE COBRO	54

3.4.9.	<i>DIAGRAMA DE SECUENCIA: REGISTRO DE PAGO</i>	55
3.5.	DIAGRAMA DE CLASE	56
3.6.	BASE DE DATOS DE LA APLICACIÓN.....	56
3.7.	PROTOTIPO DE LA INTERFAZ DEL SISTEMA INFORMATICO WEB.....	65
3.7.1.	<i>PROTOTIPO DEL LOGIN</i>	65
3.7.2.	PROTOTIPO DE LA PANTALLA DE INICIO.....	66
3.7.3.	<i>PROTOTIPO CREACIÓN DE CLIENTES Y CONTRATO</i>	67
3.7.4.	<i>PROTOTIPO LISTADO DE CLIENTES</i>	67
3.7.5.	<i>PROTOTIPO LISTADO DE CONTRATO</i>	68
3.7.6.	<i>PROTOTIPO DE REGISTRO DE MATERIALES</i>	69
3.7.7.	<i>PROTOTIPO DE REGISTRO DE COMPRAS</i>	69
3.7.8.	<i>PROTOTIPO DE LISTADO DE PRE-FACTURAS</i>	70
3.7.9.	<i>PROTOTIPO REGISTRO DE PAGO</i>	71
3.7.10.	<i>PROTOTIPO DE LA ORDEN DE COBRO (PDF)</i>	71
4.	CONSTRUCCION Y PRUEBAS	72
4.1.	HERRAMIENTAS Y TECNOLOGÍAS.....	73
4.1.1.	<i>Python 3.9.2</i>	73
4.1.2.	<i>Django 3.2</i>	73
	<i>Herramientas usadas:</i>	73
4.1.3.	<i>VALIDACIONES PROPIAS DE DJANGO</i>	73
4.1.4.	INTERFACES PRINCIPALES DEL SISTEMA	74
4.1.4.1.	<i>INTERFAZ LOGIN DEL SISTEMA.</i>	74
4.1.4.2.	<i>PANTALLA DE REGISTRO DE USUARIO</i>	75
4.1.4.3.	<i>Pantalla de Creación de Clientes Y Contratos</i>	77
4.1.4.4.	<i>PANTALLA ORDENES DE INSTALACIÓN</i>	80
4.1.4.5.	<i>PANTALLA DE REGISTRO DE MATERIALES</i>	80
4.1.4.6.	<i>REGISTRO DE PRODUCTOS</i>	83
4.1.4.7.	<i>PANTALLA REGISTRO DE COMPRAS</i>	85

4.1.4.8.	PANTALLA IMPRESIÓN DE COBROS	87
4.1.4.9.	PANTALLA DE REGISTRO DE PAGO.....	89
4.1.4.10.	GitHub.....	91
4.2.	PRUEBAS.....	91
4.2.1.	PRUEBAS DEL CÓDIGO	92
4.2.2.	MEDIDAS DE FIABILIDAD.....	93
4.2.3.	MEDIDAS DE SEGURIDAD.....	93
4.2.4.	DUPLICIDAD DEL CÓDIGO	93
4.2.5.	PRUEBAS DE RENDIMIENTO.....	94
4.2.6.	ANÁLISIS DE RESULTADOS.....	95
5.	LISTA DE REFERENCIAS	95

ÍNDICE DE FIGURAS

<i>Figura 1. Estructura básica de HTML5 Fuente: extraído de https://disenowebakus.net/imagenes/articulos/estructura-basica-de-una-pagina-web-en-html.jpg</i>	<i>26</i>
<i>Figura 2 Diseño Responsive Desing Fuente: extraído de https://jaestic.com/responsive-design/</i>	<i>27</i>
<i>Figura 3 Diagrama de Caso de Uso Sistema de Cobros.....</i>	<i>35</i>
<i>Figura 4 Ingreso al Sistema</i>	<i>35</i>
<i>Figura 5 Administración de Usuario</i>	<i>36</i>
<i>Figura 6 Administración de Clientes.....</i>	<i>38</i>
<i>Figura 7 Administración de Contrato</i>	<i>39</i>
<i>Figura 8 Administración de Ordenes de Instalación.....</i>	<i>40</i>
<i>Figura 9 Administración de Bodega</i>	<i>41</i>
<i>Figura 10 Administrar Productos</i>	<i>42</i>
<i>Figura 11 Administración de Categoría.....</i>	<i>43</i>
<i>Figura 12 Administrar Unidad de Medida.....</i>	<i>44</i>
<i>Figura 13 Administrar Proveedor</i>	<i>45</i>
<i>Figura 14 Administrar Compras</i>	<i>47</i>
<i>Figura 15 Gestión de Cobros</i>	<i>48</i>
<i>Figura 16 Diagrama de Secuencia: Inicio de Sesión</i>	<i>50</i>
<i>Figura 17 Diagrama de Secuencia: Creación de Usuario.....</i>	<i>51</i>
<i>Figura 18 Diagrama de Secuencia: Creación de Clientes y Contratos</i>	<i>51</i>
<i>Figura 19 Diagrama de Secuencia: Modificación de Contratos</i>	<i>52</i>
<i>Figura 20 Diagrama de Secuencia: Registro de Materiales</i>	<i>53</i>
<i>Figura 21 Diagrama de Secuencia: Registro de Planes</i>	<i>53</i>
<i>Figura 22 Diagrama de Secuencia: Registro de Compras</i>	<i>54</i>
<i>Figura 23 Diagrama de Secuencia: Impresión de Recibo de Cobro</i>	<i>55</i>
<i>Figura 24 Diagrama de Secuencia: Registro de Pago.....</i>	<i>55</i>
<i>Figura 25 Diagrama de Clases.....</i>	<i>56</i>
<i>Figura 26 Diagrama de la Base de Datos</i>	<i>56</i>
<i>Figura 27 Diseño del Login</i>	<i>66</i>

<i>Figura 28 Prototipo de la Pantalla de Inicio</i>	67
<i>Figura 29 Prototipo Creación de Clientes y Contrato</i>	67
<i>Figura 30 Prototipo Listado de Clientes</i>	68
<i>Figura 31 Prototipo Listado de Contrato</i>	68
<i>Figura 32 Prototipo Listado de Registro de Materiales</i>	69
<i>Figura 33 Prototipo de Registro de Compras</i>	70
<i>Figura 34 Prototipo de Listado de Pre-facturas</i>	70
<i>Figura 35 Prototipo de Registro de Pago</i>	71
<i>Figura 36 Prototipo de la Orden de Cobro (pdf)</i>	72
<i>Figura 37 Interfaz, login del sistema</i>	74
<i>Figura 38 Código de Validación del Login</i>	75
<i>Figura 39 Creacion de Usuario</i>	76
<i>Figura 40 Configuracion de Validaciones</i>	76
<i>Figura 41 Código de las Validaciones de las Contraseñas</i>	77
<i>Figura 42 Pantalla de Registro de Clientes y Contratos</i>	78
<i>Figura 43 Codigo inline form</i>	78
<i>Figura 44 Formulario del Cliente</i>	79
<i>Figura 45 Validación de Fecha del Contrato</i>	79
<i>Figura 46 Listado de Ordenes de instalación</i>	80
<i>Figura 47 Pantalla Registro de Materiales</i>	81
<i>Figura 48 Código para la Creación de la Primera Orden de Cobro</i>	82
<i>Figura 49 Registro de Productos</i>	83
<i>Figura 50 Código de Productos</i>	84
<i>Figura 51 Registro de Compras</i>	85
<i>Figura 52 Validaciones Para el Registro de Compras</i>	86
<i>Figura 53 Pantalla Listado de Compras</i>	86
<i>Figura 54 Listado de Facturas de los Clientes</i>	87
<i>Figura 55 Pdf de la Orden de Cobro</i>	87
<i>Figura 56 Codigo para la Creacion del Recibo de Cobro</i>	88

<i>Figura 57 Listado de Registro de Pago.....</i>	<i>89</i>
<i>Figura 58 Historial de Pagos de los Clientes.....</i>	<i>89</i>
<i>Figura 59 Formulario del Registro de Pago.....</i>	<i>90</i>
<i>Figura 60 Código de validación del Formulario de Pago.....</i>	<i>90</i>
<i>Figura 61 Control de Versiones del Proyecto.....</i>	<i>91</i>
<i>Figura 62 Resumen de las Pruebas Realizadas</i>	<i>92</i>
<i>Figura 63 Detalle de las Medidas de Fiabilidad.....</i>	<i>93</i>
<i>Figura 64 Detalle de las Vulnerabilidades.....</i>	<i>93</i>
<i>Figura 65 Duplicidad del código.....</i>	<i>94</i>
<i>Figura 66 Configuración Para las Pruebas de Rendimiento</i>	<i>94</i>
<i>Figura 67 Resumen, Reporte de Carga.....</i>	<i>95</i>

TABLA DE CONTENIDO

<i>Tabla 1 Registro de Clientes</i>	31
<i>Tabla 2 Creación de Contratos</i>	31
<i>Tabla 3 Registro de Ordenes de instalación</i>	31
<i>Tabla 4 Registro de Planes</i>	31
<i>Tabla 5 Inventario de productos</i>	32
<i>Tabla 6 Emision de Ordenes de Cobros</i>	32
<i>Tabla 7 Registro de pago mensual</i>	32
<i>Tabla 8 Escalabilidad</i>	32
<i>Tabla 9 Delimitacion de Usuario</i>	33
<i>Tabla 10 Disponibilidad</i>	33
<i>Tabla 11 Validaciones</i>	33
<i>Tabla 12 Nombre de Actores y su rol en la aplicación</i>	34
<i>Tabla 13 Roles que intervienen en la aplicación</i>	34
<i>Tabla 14 Ingresar al Sistema</i>	36
<i>Tabla 15 Administración de Usuarios</i>	37
<i>Tabla 16 Administración de Clientes</i>	39
<i>Tabla 17 Administración de Contrato</i>	40
<i>Tabla 18 Administración de Ordenes de Instalación</i>	41
<i>Tabla 19 Administración de Productos</i>	43
<i>Tabla 20 Administración de Categoría</i>	44
<i>Tabla 21 Administrar Unidad de Medida</i>	45
<i>Tabla 22 Administrar Proveedor</i>	46
<i>Tabla 23 Administración de Compras</i>	47
<i>Tabla 24 Gestión de Cobros</i>	49
<i>Tabla 25 Tabla Clientes</i>	57
<i>Tabla 26 Tabla Contrato</i>	58
<i>Tabla 27 Tabla Plan</i>	58
<i>Tabla 28 Tabla Parroquia</i>	58

<i>Tabla 29 Tabla Provincia</i>	58
<i>Tabla 30 Tabla Cantón</i>	59
<i>Tabla 31 Tabla Materiales</i>	59
<i>Tabla 32 Tabla Ordenes de instalación</i>	60
<i>Tabla 33 Tabla de la Factura de Encabezado</i>	60
<i>Tabla 34 Tabla Detalle de la Factura</i>	61
<i>Tabla 35 Tabla de los Pagos de la Factura</i>	62
<i>Tabla 36 Tabla de los Registros de Pago de los Recibos de Cobro</i>	62
<i>Tabla 37 Tabla de Productos</i>	63
<i>Tabla 38 Tabla de Categorías</i>	63
<i>Tabla 39 Tabla de Marcas</i>	63
<i>Tabla 40 Tabla de las Unidades de Medida</i>	63
<i>Tabla 41 Tabla Detalles de la Compra</i>	64
<i>Tabla 42 Tabla del Encabezado de la Compra</i>	65
<i>Tabla 43 Tabla de Proveedores</i>	65

RESUMEN

Este proyecto técnico es un sistema informático web para la gestión de cobros del servicio de internet de la empresa Quicknet. La creación del sistema informático dispone de las siguientes fases: introducción, marco teórico, análisis, diseño y construcción. la metodología usada fue RUP para el desarrollo del sistema informático. Al finalizar el proyecto técnico se obtuvo como resultado un sistema que realiza la gestión de clientes, contratos, ordenes de instalación, inventario y gestión de cobros. Cabe mencionar que el desarrollo del sistema fue realizado con Django y se utilizó GitHub para manejar el control de versiones del sistema.

ABSTRACT

This technical project is a web-based computer system for the management of the Quicknet company's Internet service charges. The creation of the computer system has the following phases: introduction, theoretical framework, analysis, design and construction. The methodology used was RUP for the development of the computer system. At the end of the technical project, the result was a system that manages clients, contracts, installation orders, inventory, and collection management. It is worth mentioning that the development of the system was done with Django and GitHub was used to manage the version control of the system.

1. INTRODUCCIÓN

La mayoría de las empresas, que no disponen de sistemas informáticos y/o procesos automatizados, y que trabajan apoyando en diferentes tipos de herramienta tanto manual como informáticas, como: manejo de hojas electrónicas, editores de textos, apuntes en libretas personales, entre otros, en muchas ocasiones se presentan problemas como: pérdida de información, inconsistencia de datos, consumo excesivo de recursos y tiempo, etc. Afectando, de esta manera al correcto desempeño y un servicio ineficiente para con los clientes en general. (Eduardo Pérez Bautista, 2012)

A nivel mundial las microempresas sufrieron muchos problemas por el tema de pandemia, ocasionando una pérdida del 30% de sus facturas, esto sucedió por varios factores, uno de ellos es la falta de la creación de un sistema de cobros que permita optimizar los procesos y el tiempo, esto evita un crecimiento exponencial de sus ventas por lo que se dedican a elaborar estrategias para dar a conocer sus productos o servicio (Alexandra et al., 2019)

En América Latina, la empresa AXS de Bolivia proveedora de internet sufrió múltiples denuncias por un mal manejo de la gestión de cobros generando disgusto y malos entendidos por parte de los clientes. (Millan, 2019)

En Ecuador y específicamente en la provincia de Imbabura el proveedor de internet “SAITEL.EC”, realizaron encuestas para saber que se debe mejorar de la empresa, las encuestas realizadas dieron que el 33.33% se debe mejorar el aspecto administrativo (gestión de cobro del servicio de internet) para poder efectuar con el objetivo establecido de la empresa (Hernández Calderón, 2019).

El propósito de este proyecto técnico es la elaboración un sistema informático web que cumpla las necesidades del departamento de cobranza de la empresa Quicknet, ayudando a mejorar la gestión de recaudación del servicio de internet de la empresa. El sistema informático

cubre los módulos de clientes, contrato, ordenes de instalación, ingreso de productos y facturación, ayudando al manejo y registro de datos para el manejo de información de la empresa

En cuanto a la elaboración del presente trabajo se dividirá los capítulos de la siguiente manera:

En el capítulo I “INTRODUCCION”, identificamos el problema a investigar, objetivos y justificaciones.

En el capítulo II “MARCO TEORICO” hablaremos sobre los antecedentes históricos, fundamentos teóricos.

En el capítulo II “ANALISIS Y DISEÑO” Aquí hablaremos sobre la situación actual de la empresa, los diagramas de actividades, diagramas de secuencia, diagramas de base de datos y el prototipo del sistema

En el capítulo III “CONSTRUCCION Y PRUEBAS” Aquí hablaremos sobre la construcción del sistema y sus respectivas pruebas realizadas

1.1. PROBLEMÁTICA

El presente proyecto técnico responderá a la siguiente problemática ¿Existe un sistema informático web para el departamento de cobranzas que gestione la recaudación del servicio de internet de la empresa Quicknet? La empresa Quicknet es una empresa nueva que inicio sus operaciones en el mercado el mes de febrero, esta empresa se dedica a facilitar el servicio de internet a la parroquia El Carmelo ubicada en la provincia del Carchi. La cual, a diferencia de otras entidades de proveedores de internet, esta empresa en su departamento de gestión de cobranzas no cuenta con una sistema informático que gestione el servicio de cobros de internet;

la gestión de cobros la realiza manualmente teniendo una ineficiencia en registrar y generar los cobros de los clientes. Con el sistema informático a implementarse, se lograrán beneficios como el registro de clientes, registro de ordenes de instalación del servicio de internet, control y registro de los pagos de los clientes.

Por ello, el presente trabajo técnico pretende, desarrollar un sistema informático web para el departamento de cobranzas que gestione la recaudación del servicio de internet de la empresa Quicknet y tener un eficiente manejo de caja de la empresa.

1.2. JUSTIFICACIÓN

Los sistemas informáticos se han convertido en una de las herramientas más viables en el mercado del desarrollo, y la aplicación web creada para Quicknet brindará a los clientes una mejor gestión de los servicios de recolección, mayor eficiencia y un control más ordenado del procesamiento de datos.(Mateo Taboada, 2018)

La aplicación brindará un mejor manejo de información de los clientes de la empresa, proporcionando almacenamiento de datos y registro de órdenes de pago de cada cliente que adquirirá el servicio, ofreciendo un control claro, específico y a tiempo.

Para la empresa Quicknet creada a principios del mes de febrero del año actual, la aplicación web resolverá la gestión de cobranzas del servicio de internet, manejo de caja e información de los clientes.

El incentivo para la creación de la aplicación web se debía a la falta de sistematización del control de la gestión de cobro del servicio de internet y la pérdida de registros de clientes de las planillas de pago.

El presente proyecto directamente beneficiaria a la empresa Quicknet y al departamento de gestión de cobros, así como indirectamente a desarrolladores y estudiantes relacionadas al área de la informática que pertenezcan o formen parte de la empresa.

1.3. OBJETIVOS

1.3.1. OBJETIVO GENERAL:

Desarrollar un sistema informático web que gestione la recaudación del servicio de internet en la empresa Quicknet.

1.3.2. OBJETIVO ESPECIFICO:

- Mejorar el proceso de recaudo por el servicio de internet que provee la empresa Quicknet
- Desarrollar un sistema web que automatice tanto el manejo de clientes y el proceso de cobro del servicio de internet proporcionado a dichos clientes.
- Realizar la entrega de la aplicación web de calidad y amigable al usuario, cuyo proceso se valida a través del uso métricas de evaluación de la eficiencia y la confiabilidad

1.4. METODOLOGÍA

La metodología de trabajo para el desarrollo estará basada en la metodología RUP considerando las siguientes etapas. (Rodriguez Arias & Yopez Jordan, 2021)

1.4.1. Fase de inicio: (se define el alcance del proyecto)

- Definir el alcance del proyecto
- Identificar los riesgos asociados al proyecto
- Elabora el plan de las fases y el de la iteración posterior
- Detallar la arquitectura del software.

1.4.2. Fase de definición del problema, análisis y diseño

- Diseñar solución preliminar del problema planteado

- Diseñar casos de uso, diagrama de actividades, diagrama de secuencias y diagrama de clases.
- Diagrama de la base de datos

1.4.3. Fase construcción e implementación

- Codificar el sistema web propuesto
- Evaluación en forma conjunta con los usuarios
- Realizar mejoras a la aplicación web según las recomendaciones por el usuario final

1.4.4. Fase de transición (finalización del proyecto y pruebas):

- Realización de pruebas de las mejoras realizadas al software.
- Corrección de error y defectos encontrados en las pruebas
- Fase de capacitación al usuario
- Soporte necesario en caso de fallo
- Entrega del producto final

2. MARCO TEORICO

2.1. ANTECEDENTES HISTÓRICOS

(Solis & Lizardo, 2019), Su estudio estuvo enfocado en la optimización de los procesos de cobranzas, a través de la implementación de un sistema informático para el área de créditos en la empresa Créditos Sebastián. Este software utiliza la Metodología UML–UWE para la parte de documentación y la metodología RUP para la programación o codificación del software. El resultado obtenido fue una reducción del nivel de Morosidad del 35.48% al 24.40%, lo que representa una reducción de 11.05%. En conclusión, la implementación del software ha cumplido con los objetivos planteados, mejorando procesos, reduciendo la morosidad y controlando el incremento de la cartera vencida.

(Alvarado & Nicols, 2020) en su trabajo plantea como objetivo el llegar a determinar el nivel de influencia que ejerce un sistema web sobre el proceso de cobranza en la Institución Educativa Virgen de Fátima. El autor utiliza la metodología SCRUM para el desarrollo del sistema web, porque ofrece ventajas en el desarrollo de las etapas como es el de fácil adaptación y un crecimiento iterativo. En los resultados obtenidos el nivel de endeudamiento se redujo del 47.32% al 26.44% con el uso del sistema web implementado. En conclusión, el uso del sistema web redujo el nivel de endeudamiento en un 20.88%, en el proceso de cobranza.

(CHAVEZ & JACK, 2018) Su estudio tuvo como objetivo general Implementar un sistema web para optimizar el proceso de gestión de cobranza para la empresa Service Collection S.A.C. La metodología aplicada fue RUP ya que permitió un mejor análisis y diseño del sistema. En los resultados, se manifestó que la hipótesis 1, demostró que el nivel de significancia es menor a 0.05, aceptando la hipótesis alterna por lo que, la implementación de un sistema web mejoraría el proceso de atención al cliente. En la hipótesis 2 se acepta la alterna por lo tanto la implementación del sistema reduciría el índice de morosidad para la empresa. En conclusión, se logró optimizar el proceso de cobranza de la empresa, así también teniendo una aceptación del personal de la empresa evitando la pérdida de cartera y morosidad.

(Mateo Taboada, 2018) Este estudio busca determinar el nivel de influencia que ejerce un Sistema web, en el proceso de cobranzas del área de operaciones de la empresa Lerma S.A.C. Se utilizó la metodología SCRUM, el tipo de investigación es preexperimental y con enfoque cuantitativo. En los resultados de investigación demostraron que, la implantación de un sistema web, incrementó el porcentaje de captación de contactos exitosos del 27.69% a 35.85%, de forma similar ocurrió con el porcentaje de cobro que se incrementó del 10.68% al 16.03%. Llegando a la conclusión de que el aplicativo o sistema informático mejorara el proceso de cobranzas en el área de operación de la empresa.

(Rodríguez Arias & Yepez Jordan, 2021) Este estudio pretende Establecer un sistema web por medio de programación orientada a objetos y herramientas de desarrollo libres que automaticen la información de los procesos de negocios comerciales de la empresa BELLNOVAINSER S.A. La metodología utilizada es RUP por ser un modelo ágil facilitando la creación de los procesos del sistema. Para ello se utilizan diversas técnicas o instrumentos de recolección de información como encuestas, entrevistas. Como resultado se manifestó que las tecnologías de la información ayudo a mejorar a la empresa ecuatoriana BELLNOVAINSER S.A para tener un mejor manejo de información. Llegando a la conclusión que usando técnica de investigación han permitido el desarrollo del sistema web para la automatización y gestión de la organización.

2.2. FUNDAMENTOS TEORICOS

2.2.1. SISTEMA INFORMÁTICO WEB

Un sistema informático web es un tipo de software que se encuentra codificado en un algún tipo de lenguaje de programación cuya ejecución se lo realiza mediante un navegador web, también se los conoce como aplicaciones web, una de las principales ventajas es el uso de una base de datos la que nos permitirá guardar grandes volúmenes de datos. Se encuentra montado en un servidor web conocido también como hosting brindando seguridad, estabilidad y alojamiento de la aplicación web.

2.2.2. IMPORTANCIA DE UN SISTEMA INFORMATICO WEB

Dado la adopción masiva y acelerada, a nivel global, de Internet en las últimas dos décadas, las aplicaciones web se han convertido en una parte esencial de la actividad humana revolucionando el manejo de información tanto para las empresas como para las personas.

Dependiendo las necesidades de la empresa, es posible que se requieran aplicaciones web que nos ayude a gestionar las propias operaciones de la organización, como la contabilidad,

la gestión financiera, la fabricación, el mantenimiento, la gestión de inventario y la gestión de clientes. Cada organización debe encontrar la mejor manera de optimizar sus procesos y satisfacer las crecientes necesidades de los clientes, y muchas veces la mejor manera es utilizar un software personalizado.

2.2.3. HTML5

Según (Vega & Van Der Henst S, 2011) HTML5 significa (Lenguaje de marcado de hipertexto), es una nueva versión (aún en desarrollo) que, junto con CSS, define nuevos estándares de desarrollo web, rediseña el código para abordar problemas y, por lo tanto, lo actualiza para cumplir con los nuevos requisitos.

Según (Vega & Van Der Henst S, 2011) HTML5 (Lenguaje de marcado de hipertexto), es una nueva versión que junto con CSS, define nuevos estándares en el diseño y desarrollo web. Rediseña el código para abordar problemas y, por lo tanto, lo actualiza para cumplir con los nuevos requisitos. No se limita a crear nuevas etiquetas o atributos, sino que incluye muchas características nuevas y proporciona una plataforma para desarrollar aplicaciones web complejas (a través de API).

2.2.4. ESTRUCTURA BASICA DE UN DOCUMENTO HTML5

Se divide en 3 etiquetas principales, la etiqueta <html> es el encargado de envolver todo el contenido de la página web, la etiqueta <head> designa la información sobre la página web, como el título, palabras claves de búsqueda y sus hojas de estilo y la etiqueta <body> alberga todo el contenido a mostrar en el navegador.

Estructura básica



Figura 1. Estructura básica de HTML5

2.2.5. CSS

CSS es un lenguaje de hojas de estilo en cascada utilizado para controlar la apariencia, el estilo o presentación de documentos web definidos en HTML y XHTML siendo uno de los pilares más importante para el diseño de aplicación web y páginas web.

2.2.6. RESPONSIVE DESING

Responsive desing o diseño web adaptable, es la adaptación de un diseño web ofreciéndole al usuario la adaptabilidad y manejo de la página web sin importar el tamaño de los dispositivos de navegación.

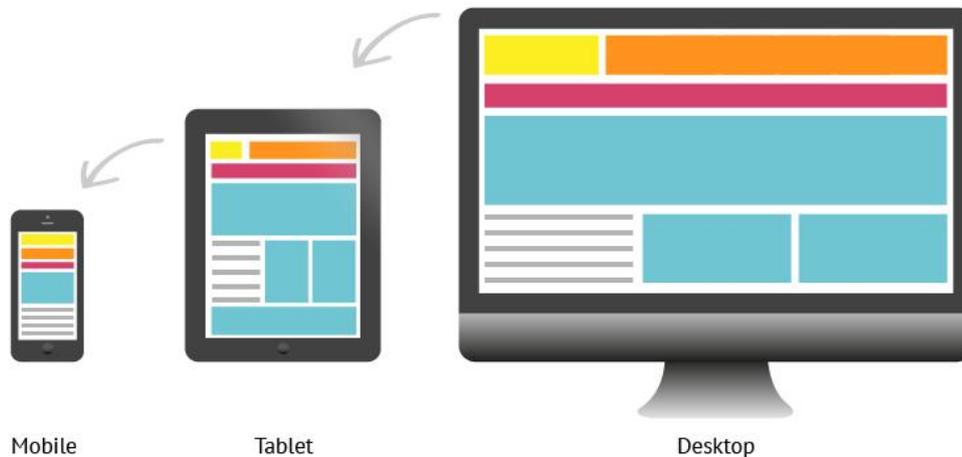


Figura 2 Diseño Responsive Desing
Fuente: extraído de <https://jaestic.com/responsive-design/>

2.2.7. GESTION DE COBROS

Desde 1792 hasta 1750 d.C., Mesopotamia promulgó el Código de Hammurabi. Es una cláusula judicial elaborada en el litigio entre el deudor y el acreedor, y tiene como objetivo brindar cierta comodidad a los agricultores para saldar sus deudas. En Mesopotamia, “el préstamo es un comportamiento muy común, se basa en uno de los dos productos básicos que se utilizan como moneda: la cebada o la plata”. También explicó que todo negocio crediticio es gratuito, si está interesado debe ser formalmente confirmado por un contrato escrito. (Morales & Morales, 2014)

El área de crédito y cobranza de toda empresa es una parte clave, que antes de vender debe decidir el objetivo de ventas, la cantidad y el período de ventas. Seguido a la venta, el área de cobranza debe asegurar que se cumplen las condiciones de pago, caso contrario debe tomar las medidas necesarias para garantizar el pago lo más pronto posible. Esto está relacionado con los ingresos de la empresa y directamente relacionado con el capital de trabajo, y se refleja en sus estados financieros. (Morales & Morales, 2014)

El primer Montepío nació en Perugia en 1462 y el segundo nació en Gubbio en 1463. Posteriormente se extendieron por Italia y, con el apoyo de los franciscanos, las autoridades clásicas ayudaron a los pobres y protegieron a los cristianos de la usura. Los primeros Montes de Piedad eran muy frágiles y muchos de ellos desaparecieron por diversos motivos, entre ellos: falta de liquidez, funcionarios inútiles, empleados que malversaron fondos, defraudaron a los clientes, saquearon la ciudad, la codicia del príncipe, y le dieron préstamos a la ciudad que ya había sido anunciada. Quiebras, etc. (Morales & Morales, 2014)

Para una empresa la gestión de cobros es fundamental, y esto se logra mediante una gestión de capital actualizada. Por tanto, se debe cumplir con las obligaciones y condiciones de pago y fundamentalmente hacer valer nuestro derecho a recibir el pago. Por gestión de cobros entendemos el control de las diferentes facturas que una empresa recibe de los clientes a través de estrategias, tareas administrativas y acciones financieras, representando así el monto económico a recibir.

2.2.8. MYSQL

MySQL es un sistema gestor de base de datos relacionales, su licencia es de código abierto, utiliza una arquitectura cliente-servidor y se ha convertido en uno de las mejores bases de datos a nivel mundial por que permite el manejo de grandes volúmenes de datos y su gran adaptabilidad para poder usarse con cualquier tipo de lenguaje de programación. (MySQL, 2021)

2.2.9. DJANGO

Es un Framework de aplicaciones web y de licencia open source escrito en Python, que respeta el patrón de diseño modelo-vista-template (MVT). Este framework facilita la creación de aplicaciones web de manera sencilla ayudando a la escalabilidad de proyectos medianos o grandes. (Django, 2005)

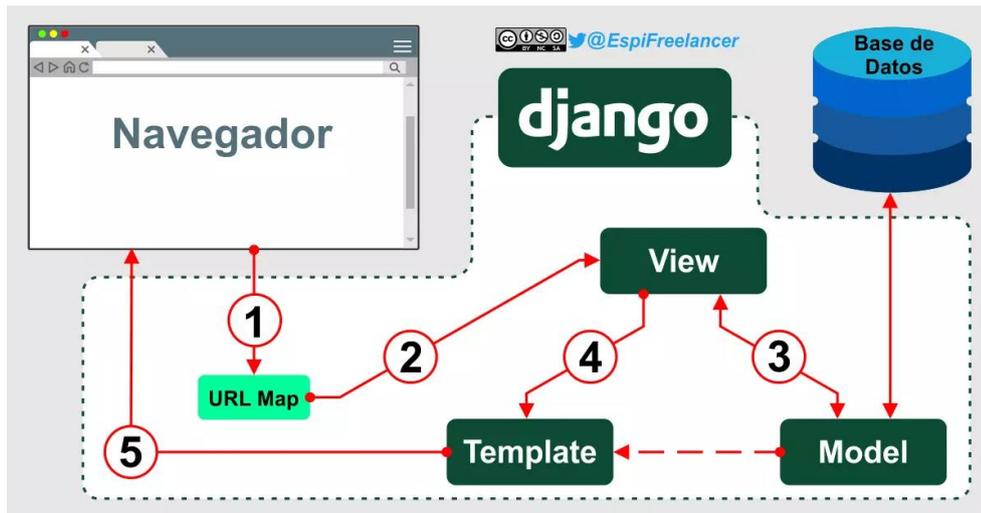


Figura 3 Modelo, Vista, Template Django
 Fuente: https://espifreelancer.com/images/Django_mtv.webp

2.2.10. FRAMEWORK

Es una estructura jerárquica que muestra qué tipos de programas pueden o deben construirse y cómo se relacionan entre sí. Algunos marcos de sistemas informáticos también incluyen programas reales, especifican interfaces de programación o proporcionan herramientas de programación para usar los marcos.(techtarget, n.d.)

2.2.11. JAVASCRIPT

Según (Ramos, 2021) JavaScript es el lenguaje de programación encargado de dotar de mayor interactividad y dinamismo a las páginas web. Cuando JavaScript se ejecuta en el navegador, no requiere un compilador. El navegador lee el código directamente, sin necesidad de instalar ninguna aplicación. Como tal, es reconocido como uno de los tres lenguajes nativos de la web, junto con HTML (contenido y su estructura) y CSS (diseño de contenido y su estructura).

2.2.12. BOOTSTRAP

Es un framework de diseño que se utiliza para el desarrollo frontend, se especializa en diseñar páginas web de una manera rápida y sencilla, teniendo como objetivo principal la creación de sitios web responsive.

2.2.13. VISUAL STUDIO CODE

Es un editor de código multiplataforma que puede ser instalado en cualquier sistema operativo, ya sea Linux, Windows o Mac. Es un editor de código muy potente enfocado a aumentar el rendimiento de los programadores.

3. ANALISIS Y DISEÑO

3.1. SITUACIÓN ACTUAL

La empresa Quicknet Ecuador proporciona servicio de internet en la provincia del Carchi cantón El Carmelo. El departamento de cobranza no cuenta con un manejo de control de información para sus clientes y mucho menos del registro de sus aportaciones económicas mes por mes por los pagos del servicio de internet, por lo que se decide la implementación de un sistema informático web que facilite al departamento de cobranza.

El sistema informático abarca los siguientes procesos:

- Gestión de clientes
- Gestión de contrato
- Gestión de ordenes de instalación
- Gestión de planes
- Gestión de productos
- Consulta y generación de ordenes de cobro
- Registro de pagos

Mas adelante se describe cada uno

3.2. REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES

3.2.1. Requerimientos Funcionales

Los requerimientos funcionales son los objetivos principales que se tienen que cumplir en la creación de cualquier sistema informático a desarrollar.

A continuación, hablaremos de los requisitos funcionales que el software debe tener para cumplir con la necesidad del departamento de cobranza:

ID:	RF-001	Relación:	
Descripción:	Registro de Clientes	Autor:	Boris Tapia
<ul style="list-style-type: none"> El técnico podrá registrar los datos de los clientes como el nombre, la cedula los teléfonos, la provincia entre otra información del usuario 			

Tabla 1 Registro de Clientes

ID:	RF-002	Relación:	
Descripción:	Registro del contrato	Autor:	Boris Tapia
<ul style="list-style-type: none"> El técnico registrará el contrato del servicio de internet del cliente, se registrará el número de contrato, fecha del contrato, el tipo de plan y la dirección de la instalación del contrato 			

Tabla 2 Creación de Contratos

ID:	RF-003	Relación:	
Descripción:	Registro de la orden de instalación	Autor:	Boris Tapia
<ul style="list-style-type: none"> El técnico registrara los materiales de la orden de instalación con la fecha en la que se despliego el servicio de internet 			

Tabla 3 Registro de Ordenes de instalación

ID:	RF-004	Relación:	
Descripción:	Registro de Planes	Autor:	Boris Tapia
<ul style="list-style-type: none"> El Administrador podrá registrar los planes establecidos por la empresa 			

Tabla 4 Registro de Planes

ID:	RF-005	Relación:	
Descripción:	Inventario de Productos	Autor:	Boris Tapia
<ul style="list-style-type: none"> El usuario podrá registrar los productos, la categoría, las marcas, la unidad de medida, el proveedor y registrar la factura de compra de los productos para el tema de stock 			

Tabla 5 Inventario de productos

ID:	RF-006	Relación:	
Descripción:	Emisión de Ordenes de cobros	Autor:	Boris Tapia
<ul style="list-style-type: none"> El departamento de cobranza genera automáticamente las ordenes de cobro para los clientes. 			

Tabla 6 Emisión de Ordenes de Cobros

ID:	RF-007	Relación:	
Descripción:	Registro de pago mensuales	Autor:	Boris Tapia
<ul style="list-style-type: none"> El departamento de cobranza registrara los pagos mensuales de los clientes 			

Tabla 7 Registro de pago mensual

3.2.2. REQUERIMIENTOS NO FUNCIONALES

ID:	RNF-001	Relación:	
Descripción:	Escalabilidad	Autor:	Boris Tapia
<ul style="list-style-type: none"> El aplicativo web tiene la posibilidad de usar debidamente los recursos de base de datos, diseño, datos para el crecimiento interno de la aplicación en la empresa 			

Tabla 8 Escalabilidad

ID:	RNF-002	Relación:	
-----	---------	-----------	--

Descripción:	Delimitación de Usuarios	Autor:	Boris Tapia
<ul style="list-style-type: none"> El sistema contempla la delimitación de sesiones para cada usuario 			

Tabla 9 Delimitación de Usuario

ID:	RNF-003	Relación:	
Descripción:	Disponibilidad	Autor:	Boris Tapia
<ul style="list-style-type: none"> La disponibilidad de la aplicación es para todos los usuarios los 7 días de la semana y las 24 horas 			

Tabla 10 Disponibilidad

ID:	RNF-004	Relación:	
Descripción:	Validaciones	Autor:	Boris Tapia
<ul style="list-style-type: none"> El aplicativo web dispone de validaciones para que el uso sea de una manera correcta 			

Tabla 11 Validaciones

3.3. DISEÑO DEL SISTEMA

3.3.1. DESCRIPCIÓN DE ACTORES

En el sistema informático se identifica 4 actores con diferentes funciones en el sistema, describirá a continuación en la tabla

PERFIL	DESCRIPCIÓN
Administrador	Usuario que administra el sistema, es capaz de modificar, crear nuevos usuarios, permite gestionar los permisos de cada usuario, ingresar, modifica y elimina planes, puede modificar información de los clientes, además gestiona el mantenimiento de la aplicación
Técnico	Usuario capaz de ingresar, clientes, contratos y ordenes de instalación

Bodeguero	Usuario encargado del ingreso de productos, marca, modelo, unidades de medida, proveedores, categoría y las compras
Cobrador	Usuario encargado del área de registro de cobro y de la generación de ordenes de cobro.

Tabla 12 Nombre de Actores y su rol en la aplicación

3.3.2. DESCRIPCIÓN DE LAS TAREAS

PERFIL	NOMBRE DEL SERVICIO
Administrador	<ul style="list-style-type: none"> • Gestiona los usuarios de la aplicación y los permisos • Creación, modificación y eliminación de clientes • Gestiona los planes del servicio • Gestiona el mantenimiento de la aplicación • Gestión de los datos sobre las actividades de los técnicos, bodegueros, y cobrador
Técnico	<ul style="list-style-type: none"> • Gestiona la creación de nuevos clientes • Consulta el listado de clientes • Gestiona la creación de contratos del servicio de internet • Consulta el listado de contratos • Gestiona la creación de ordenes de instalación • Consulta el listado de materiales usados
Bodeguero	<ul style="list-style-type: none"> • Gestiona el ingreso de productos • Gestiona el ingreso de categoría • Gestiona el ingreso de marca • Gestiona el ingreso de unidades de medida • Gestiona el ingreso de proveedores • Gestiona las compras de productos
Cobrador	<ul style="list-style-type: none"> • Genera ordenes de cobros para los clientes • Registra los pagos de los clientes

Tabla 13 Roles que intervienen en la aplicación

3.3.3. DIAGRAMAS DE CASO DE USO

En esta sección detallaremos los diagramas de caso de uso del sistema que identificamos en nuestro sistema con sus especificaciones en cada uno de los diagramas desarrollados.

3.3.3.1. CASO DE USO SISTEMA DE COBROS

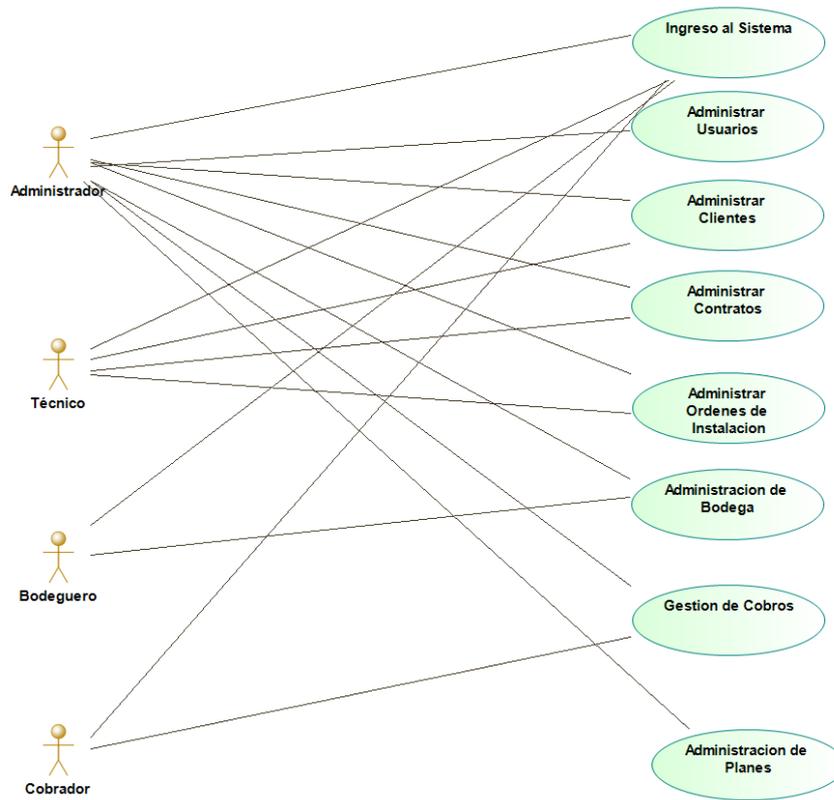


Figura 4 Diagrama de Caso de Uso Sistema de Cobros

3.3.3.2. CASO DE USO: INGRESO AL SISTEMA

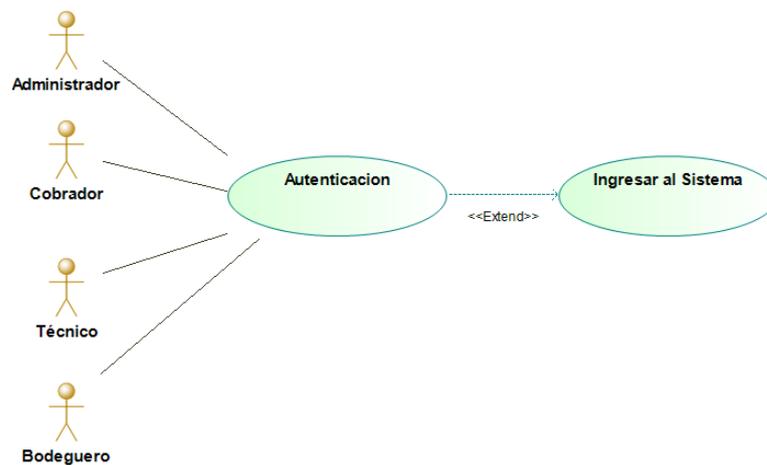


Figura 5 Ingreso al Sistema

3.3.3.3. ESPECIFICACIÓN DEL CASO DE USO: INGRESO AL SISTEMA

	Caso de Uso: Autenticación - Ingresar al Sistema
Actores:	Usuario: Administrador Usuario: Técnico Usuario: Bodeguero Usuario: Cobrador
Descripción	Verifica un usuario que se encuentra registrado para ingreso al sistema
Precondiciones:	Los usuarios previamente deben estar registrados en el sistema
Post Condiciones:	La validación se realiza correctamente
Flujo normal de Eventos	Validar <ol style="list-style-type: none"> 1. El usuario ingresa en un formulario el usuario y la contraseña 2. El sistema valida que la información sea correcta 3. Si el sistema valida que las credenciales sean correctas, el sistema muestra la pantalla principal con el menú de opciones pertinentes para el usuario
Flujo Alternativo	Usuario no registrado <ol style="list-style-type: none"> 1. En el paso 2 del flujo normal, si el usuario no existe en la base de datos se solicita al administrador la creación del usuario con el permiso requerido

Tabla 14 Ingresar al Sistema

3.3.3.4. CASO DE USO: ADMINISTRACIÓN DE USUARIOS

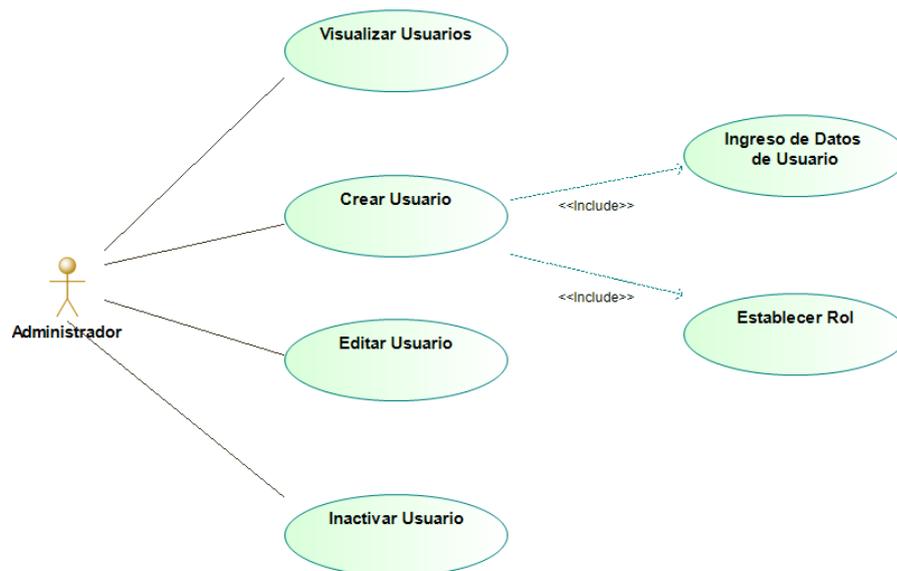


Figura 6 Administración de Usuario

3.3.3.5. **ESPECIFICACIÓN DEL CASO DE USO: ADMINISTRACIÓN DE USUARIOS**

	Caso de Uso: Administración de Usuarios
Actores:	Usuario: Administrador
Descripción	Registra, modifica, inactiva usuarios para el uso del sistema
Precondiciones:	El usuario debe estar registrado como administrador
Post Condiciones:	Resultado almacenado con éxito
Flujo normal de Eventos	<p>Nuevo</p> <ol style="list-style-type: none"> 1. El usuario selecciona la opción de administrar usuarios 2. El sistema despliega una lista de todos los usuarios 3. El usuario selecciona crear usuario nuevo 4. El sistema muestra el formulario para la creación de usuario 5. El usuario selecciona el rol a que pertenece 6. El sistema verifica que la información sea correcta y guarda la información
Flujo Alternativo	<p>Modificar</p> <ol style="list-style-type: none"> 1. En el paso 3 del flujo normal, si el usuario selecciona editar usuario 2. El sistema muestra la información del usuario a modificar en un formulario 3. El usuario realiza los cambios pertinentes de la modificación 4. El sistema verifica que la información sea correcta y actualiza la información <p>Inactivar</p> <ol style="list-style-type: none"> 1. En el paso 3 del flujo normal, si el usuario selecciona inactivar usuario 2. El sistema muestra un mensaje indicando si está seguro inactivar el usuario 3. Si el usuario indica que si 4. El sistema inactivo al usuario lo que le inhabilita para ingresar al sistema

Tabla 15 Administración de Usuarios

3.3.3.6. CASO DE USO: ADMINISTRACIÓN DE CLIENTES

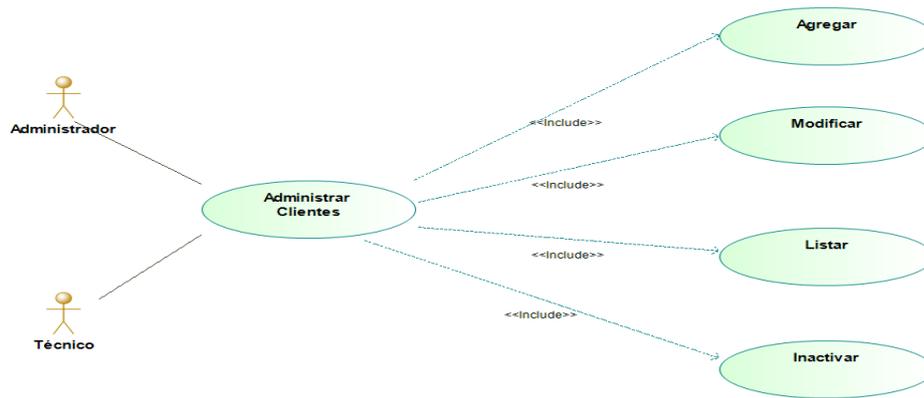


Figura 7 Administración de Clientes

3.3.3.7. ESPECIFICACIÓN DEL CASO DE USO: ADMINISTRACIÓN DE CLIENTES

	Caso de Uso: Administración de Clientes
Actores:	Usuario: Administrador Usuario: Técnico
Descripción	Registra, modifica, inactiva Clientes
Precondiciones:	El usuario debe tener los permisos de administrador o técnico
Post Condiciones:	Resultado almacenado con éxito
Flujo normal de Eventos	Nuevo <ol style="list-style-type: none"> 1. El usuario selecciona del menú la opción cliente 2. El usuario selecciona del submenú la opción registro del cliente 3. El sistema muestra el formulario para la creación de nuevo cliente 4. El usuario registra la información del cliente 5. El sistema verifica que la información sea correcta y guarda la información
Flujo Alternativo	Modificar <ol style="list-style-type: none"> 1. En el paso 2 del flujo normal, si el usuario selecciona listado de clientes

	<ol style="list-style-type: none"> 2. El sistema muestra el listado de clientes 3. El usuario visualiza el detalle del cliente seleccionando el botón detalle 4. El usuario selecciona el botón editar 5. El usuario visualiza la información del usuario a modificar 6. El usuario realiza los cambios pertinentes de la modificación 7. El sistema verifica que la información sea correcta y actualiza la información <p>Inactivar</p> <ol style="list-style-type: none"> 1. En el paso 2 del flujo normal, si el usuario selecciona listado de clientes 2. El sistema muestra el listado de clientes 3. El usuario visualiza el detalle del cliente seleccionando el botón detalle 4. El usuario selecciona la opción inactivar cliente 5. El sistema muestra un mensaje indicando si está seguro inactivar el cliente 6. Si el usuario confirma que desea desactivar 7. El sistema inactivo al cliente lo que le inhabilita para los siguientes módulos
--	---

Tabla 16 Administración de Clientes

3.3.3.8. CASO DE USO: ADMINISTRACIÓN DE CONTRATOS

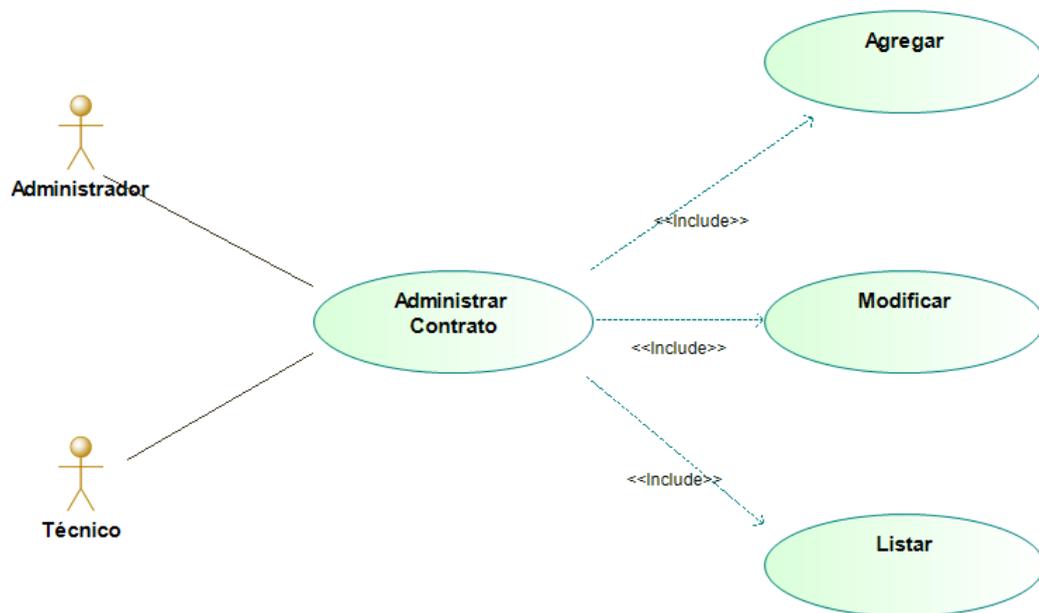


Figura 8 Administración de Contrato

3.3.3.9. ESPECIFICACIÓN DEL CASO DE USO: ADMINISTRACIÓN DE CONTRATO

	Caso de Uso: Administración de Contrato
Actores:	Usuario: Administrador Usuario: Técnico

Descripción	Registra y modifica contratos para el uso del sistema
Precondiciones:	El cliente debe estar registrado en el sistema
Post Condiciones:	Resultado almacenado con éxito
Flujo normal de Eventos	<p>Agregar</p> <ol style="list-style-type: none"> 1. El usuario selecciona en el menú la opción contrato 2. El usuario selecciona en el submenú la opción de registro de contrato 3. El sistema muestra el formulario para la creación de un contrato 4. El usuario selecciona un cliente 5. El usuario registra la información del contrato 6. El sistema verifica que la información sea correcta y guarda la información
Flujo Alternativo	<p>Modificar</p> <ol style="list-style-type: none"> 1. En el paso 2 del flujo normal, si el usuario selecciona lista de contrato. 2. El sistema muestra la información de los contratos creados. 3. El usuario selecciona el botón de visualizar los detalles cliente. 4. El sistema despliega la información del cliente seleccionado. 5. El usuario selecciona el botón de editar información del cliente. 6. El usuario visualiza la información del usuario a modificar. 7. El usuario realiza los cambios pertinentes de la modificación. 8. El sistema verifica que la información sea correcta y actualiza la información.

Tabla 17 Administración de Contrato

3.3.3.10. CASO DE USO: ADMINISTRACIÓN DE ORDENES DE INSTALACIÓN

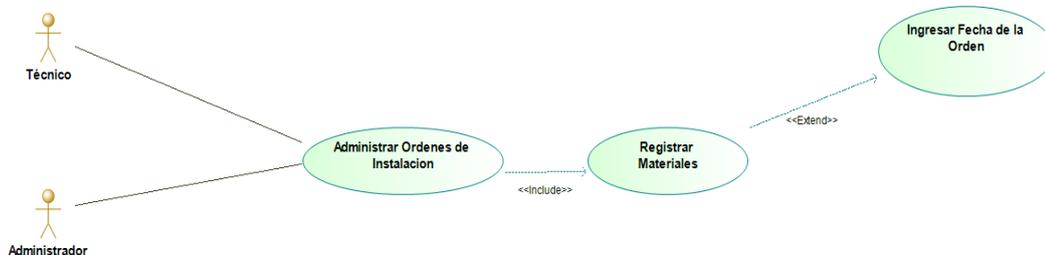


Figura 9 Administración de Ordenes de Instalación

3.3.3.11. ESPECIFICACIÓN DEL CASO DE USO: ADMINISTRACIÓN DE ORDENES DE INSTALACIÓN

	Caso de Uso: Administración de Ordenes de Instalación
Actores:	Usuario: Administrador Usuario: Técnico
Descripción	Registro de materiales y Lista general de Ordenes de instalación
Precondiciones:	<ul style="list-style-type: none"> - El cliente debe tener un contrato creado previamente - Al crear un contrato automáticamente se crea una orden de instalación del cliente

Post Condiciones:	Resultado almacenado con éxito
Flujo normal de Eventos	<p>Registro de Materiales</p> <ol style="list-style-type: none"> 1. El usuario selecciona en el menú la opción Ordenes de instalación 2. El usuario selecciona en el submenú la opción de listado de ordenes de instalación 3. El sistema muestra las ordenes de instalación de los clientes 4. El usuario selecciona registro de materiales 5. El usuario registra la información de los materiales 6. El usuario registra la fecha de la instalación del servicio de internet 7. El sistema verifica que la información sea correcta y guarda la información
Flujo Alterno	<p>Listar</p> <ol style="list-style-type: none"> 9. En el paso 2 del flujo normal, si el usuario selecciona listado de materiales. 10. El sistema muestra la información en general de los materiales registrados.

Tabla 18 Administración de Ordenes de Instalación

3.3.3.12. CASO DE USO: ADMINISTRACIÓN DE BODEGA

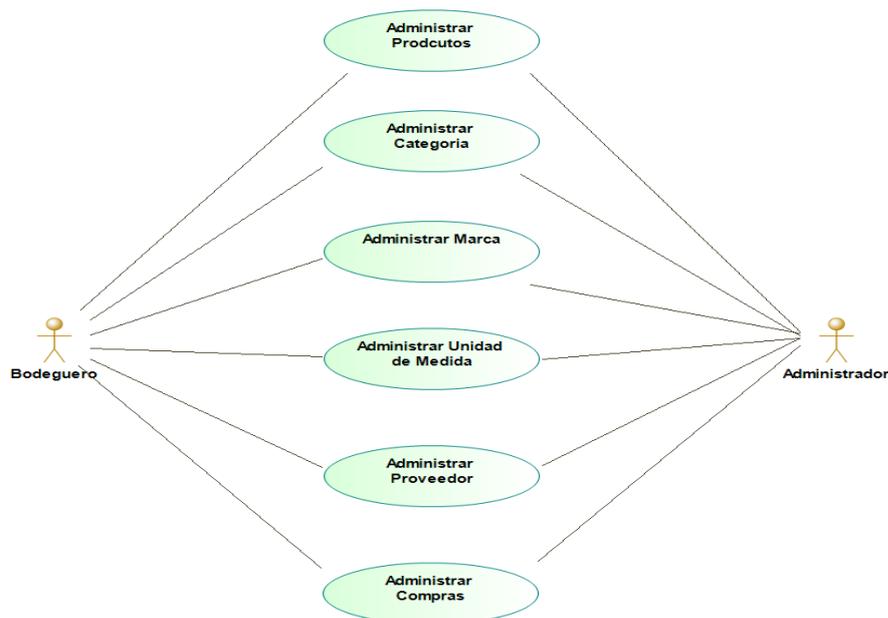


Figura 10 Administración de Bodega

3.3.3.13. CASO DE USO: ADMINISTRACIÓN DE BODEGA

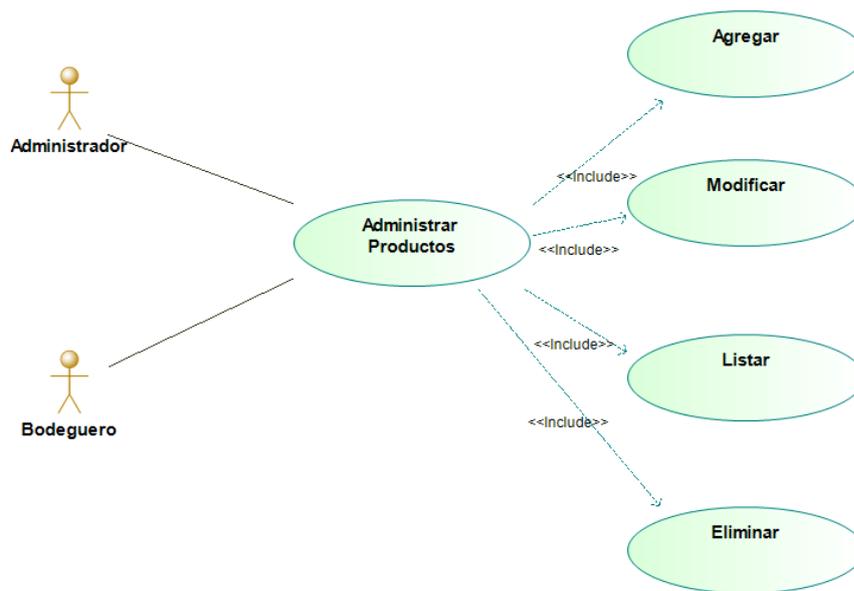


Figura 11 Administrar Productos

3.3.3.14. ESPECIFICACIÓN DEL CASO DE USO: ADMINISTRACIÓN DE PRODUCTOS

	Caso de Uso: Administración de Productos
Actores:	Usuario: Administrador Usuario: Bodeguero
Descripción	Agregar, Listar, Modificar y Eliminar Productos
Precondiciones:	<ul style="list-style-type: none"> - El usuario debe tener los permisos de administrador o bodeguero - Los productos deben tener previamente creada una marca, categoría, unidad de medida
Post Condiciones:	Resultado almacenado con éxito
Flujo normal de Eventos	<p>Agregar</p> <ol style="list-style-type: none"> 1. El usuario selecciona en el menú la opción Ingresos 2. El usuario selecciona en el submenú la opción de productos 3. El sistema muestra los productos 4. El usuario selecciona la opción crear nuevo producto 5. El sistema muestra el formulario de creación de nuevo producto 6. El usuario registra un nuevo producto 7. El sistema verifica que la información sea correcta y guarda la información
Flujo Alternativo	Modificar

	<ol style="list-style-type: none"> 1. En el paso 4 del flujo normal, si el usuario selecciona editar. 2. El sistema despliega la información del producto seleccionado. 3. El usuario realiza los cambios pertinentes de la modificación. 4. El sistema verifica que la información sea correcta y actualiza la información. <p>Eliminar</p> <ol style="list-style-type: none"> 1. En el paso 4 del flujo normal, si el usuario selecciona eliminar. 2. El sistema muestra un mensaje indicando si está seguro eliminar el producto 3. Si el usuario confirma que desea eliminar 4. El sistema eliminara el producto lo que le inhabilita para los siguientes módulos
--	---

Tabla 19 Administración de Productos

Caso de uso: Administración de Categoría

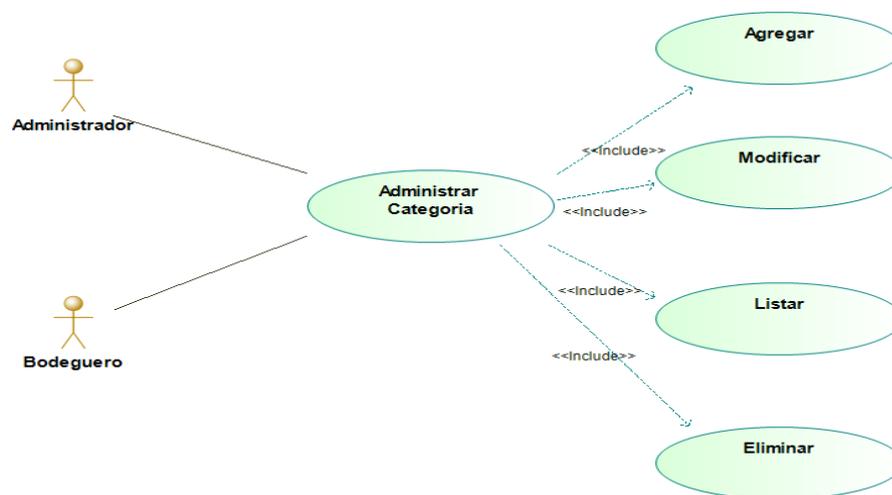


Figura 12 Administración de Categoría

3.3.3.15. ESPECIFICACIÓN DEL CASO DE USO: ADMINISTRACIÓN DE CATEGORÍA

	Caso de Uso: Administración de Categoría
Actores:	Usuario: Administrador Usuario: Bodeguero
Descripción	Agregar, Listar, Modificar y Eliminar Categoría
Precondiciones:	El usuario debe tener los permisos de administrador o bodeguero
Post Condiciones:	Resultado almacenado con éxito
Flujo normal de Eventos	<p>Registrar</p> <ol style="list-style-type: none"> 1. El usuario selecciona en el menú la opción Ingresos 2. El usuario selecciona en el submenú la opción de marca 3. El sistema muestra las marcas creadas 4. El usuario selecciona la opción crear nueva marca 5. El sistema muestra el formulario de creación de nueva marca

	6. El usuario registra una nueva marca 7. El sistema verifica que la información sea correcta y guarda la información
Flujo Alternativo	<p>Modificar</p> <ol style="list-style-type: none"> 1. En el paso 4 del flujo normal, si el usuario selecciona editar. 2. El sistema despliega la información de la marca seleccionada. 3. El usuario realiza los cambios pertinentes de la modificación. 4. El sistema verifica que la información sea correcta y actualiza la información. <p>Eliminar</p> <ol style="list-style-type: none"> 1. En el paso 4 del flujo normal, si el usuario selecciona eliminar. 2. El sistema muestra un mensaje indicando si está seguro de eliminar la marca 3. Si el usuario confirma que desea eliminar 4. El sistema eliminará la marca lo que le inhabilita para los siguientes módulos

Tabla 20 Administración de Categoría

3.3.3.16. CASO DE USO: ADMINISTRACIÓN DE UNIDAD DE MEDIDA

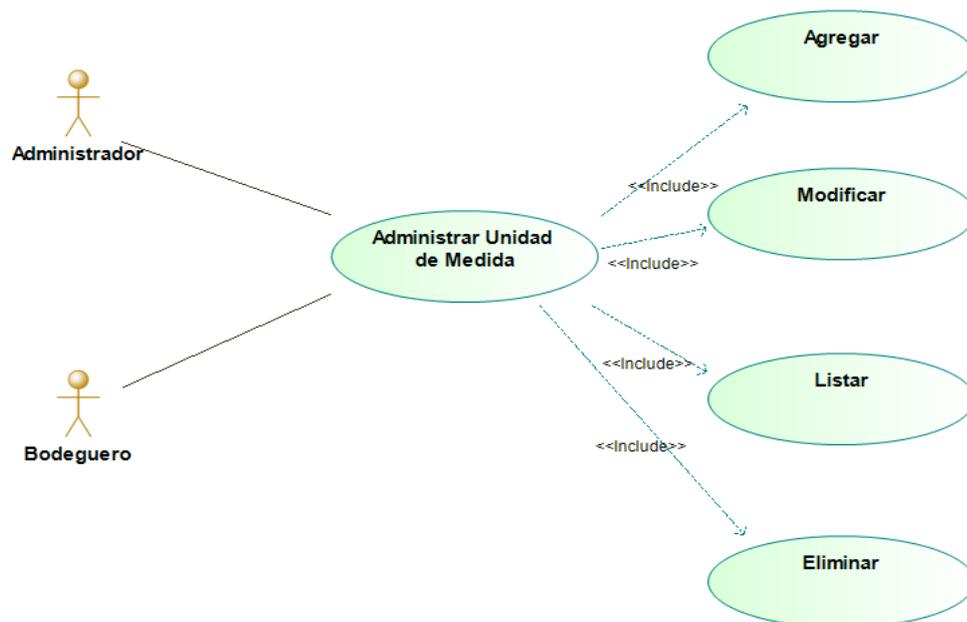


Figura 13 Administrar Unidad de Medida

3.3.3.17. ESPECIFICACIÓN DEL CASO DE USO: ADMINISTRACIÓN DE UNIDAD DE MEDIDA

	Caso de Uso: Administración de Unidad de Medida
Actores:	Usuario: Administrador Usuario: Bodeguero
Descripción	Agregar, Listar, Modificar y Eliminar unidad de medida
Precondiciones:	El usuario debe tener los permisos de administrador o bodeguero

Post Condiciones:	Resultado almacenado con éxito
Flujo normal de Eventos	<p>Registrar</p> <ol style="list-style-type: none"> 1. El usuario selecciona en el menú la opción Ingresos 2. El usuario selecciona en el submenú la opción unidad de medida 3. El sistema muestra las unidades de medidas creadas 4. El usuario selecciona la opción crear nueva unidad de medida 5. El sistema muestra el formulario de crear nueva unidad de medida 6. El usuario registra una nueva unidad de medida 7. El sistema verifica que la información sea correcta y guarda la información
Flujo Alternativo	<p>Modificar</p> <ol style="list-style-type: none"> 1. En el paso 4 del flujo normal, si el usuario selecciona editar. 2. El sistema despliega la información de la unidad de medida seleccionado. 3. El usuario realiza los cambios pertinentes de la modificación. 4. El sistema verifica que la información sea correcta y actualiza la información. <p>Eliminar</p> <ol style="list-style-type: none"> 1. En el paso 4 del flujo normal, si el usuario selecciona eliminar. 2. El sistema muestra un mensaje indicando si está seguro eliminar la marca 3. Si el usuario confirma que desea eliminar 4. El sistema eliminara la unidad de medida lo que le inhabilita para los siguientes módulos

Tabla 21 Administrar Unidad de Medida

3.3.3.18. CASO DE USO: ADMINISTRACIÓN DE PROVEEDOR

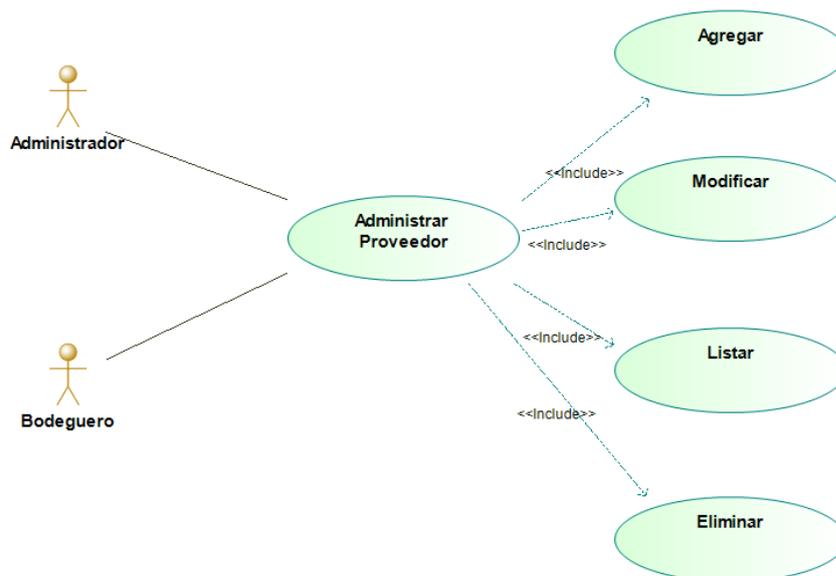


Figura 14 Administrar Proveedor

3.3.3.19. ESPECIFICACIÓN DEL CASO DE USO: ADMINISTRACIÓN DE UNIDAD DE MEDIDA

	Caso de Uso: Administración de Proveedores
Actores:	Usuario: Administrador Usuario: Bodeguero
Descripción	Agregar, Listar, Modificar y Eliminar Proveedores
Precondiciones:	El usuario debe tener los permisos de administrador o bodeguero
Post Condiciones:	Resultado almacenado con éxito
Flujo normal de Eventos	<p>Registrar</p> <ol style="list-style-type: none"> 1. El usuario selecciona en el menú la opción ingresos 2. El usuario selecciona en el submenú la opción proveedor 3. El sistema muestra los proveedores creados 4. El usuario selecciona la opción crear nuevo proveedor 5. El sistema muestra el formulario de crear nuevo proveedor 6. El usuario registra un nuevo proveedor 7. El sistema verifica que la información sea correcta y guarda la información
Flujo Alternativo	<p>Modificar</p> <ol style="list-style-type: none"> 1. En el paso 4 del flujo normal, si el usuario selecciona editar. 2. El sistema despliega la información del proveedor seleccionado. 3. El usuario realiza los cambios pertinentes de la modificación. 4. El sistema verifica que la información sea correcta y actualiza la información. <p>Eliminar</p> <ol style="list-style-type: none"> 1. En el paso 4 del flujo normal, si el usuario selecciona eliminar. 2. El sistema muestra un mensaje indicando si está seguro eliminar la marca 3. Si el usuario confirma que desea eliminar 4. El sistema eliminara el proveedor lo que inhabilita para los siguientes módulos

Tabla 22 Administrar Proveedor

3.3.3.20. CASO DE USO: ADMINISTRACIÓN DE COMPRAS

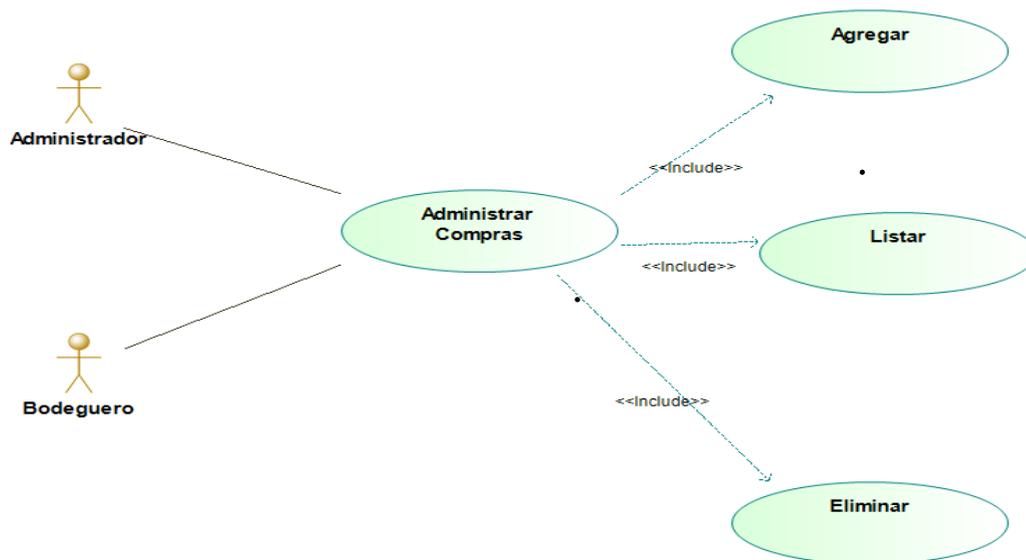


Figura 15 Administrar Compras

3.3.3.21. ESPECIFICACIÓN DEL CASO DE USO: ADMINISTRACIÓN DE COMPRAS

	Caso de Uso: Administración de Compras
Actores:	Usuario: Administrador Usuario: Bodeguero
Descripción	Agregar, Listar, Modificar Compras
Precondiciones:	El usuario debe tener los permisos de administrador o bodeguero Previamente debe tener registros los módulos proveedores, productos, categoría, marca y unidad de medida
Post Condiciones:	Resultado almacenado con éxito
Flujo normal de Eventos	Registrar <ol style="list-style-type: none"> 1. El usuario selecciona en el menú la opción ingresos 2. El usuario selecciona en el submenú la opción compras 3. El sistema muestra las compras realizadas 4. El usuario selecciona la opción crear nueva compra 5. El sistema muestra el formulario de crear nueva compra 6. El usuario registra una nueva compra 7. El sistema verifica que la información sea correcta y guarda la información
Flujo Alternativo	Modificar <ol style="list-style-type: none"> 1. En el paso 4 del flujo normal, si el usuario selecciona editar. 2. El sistema despliega la información de la compra seleccionada. 3. El usuario realiza los cambios pertinentes de la modificación. 4. El sistema verifica que la información sea correcta y actualiza la información.

Tabla 23 Administración de Compras

3.3.3.22. CASO DE USO: GESTIÓN DE COBROS

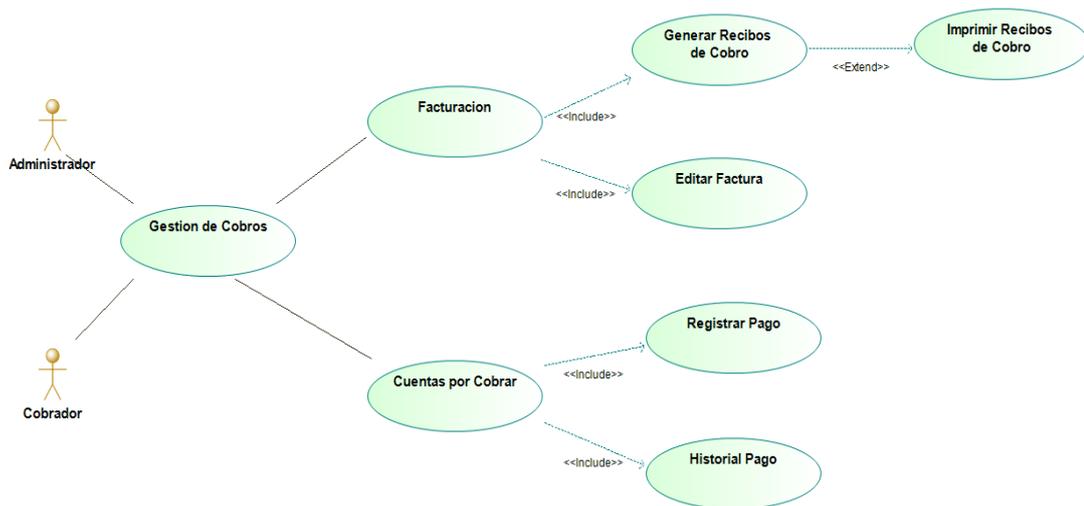


Figura 16 Gestión de Cobros

3.3.3.23. ESPECIFICACIÓN DEL CASO DE USO: GESTIÓN DE COBROS

	Caso de Uso: Gestión de Cobros
Actores:	Usuario: Administrador Usuario: Cobrador
Descripción	El usuario podrá generar recibos de cobro, editar facturas, imprimir recibos de cobro, registrar pago y observar el historial de pagos
Precondiciones:	El usuario debe tener los permisos de administrador o cobrador Previamente debe tener un contrato y una orden de instalación creada
Post Condiciones:	Resultado almacenado con éxito Impresión de recibos de cobro con éxito
Flujo normal de Eventos	Generación de recibos de cobro <ol style="list-style-type: none"> 1. El usuario selecciona en el menú la opción facturación 2. El usuario selecciona en el submenú la opción facturación 3. El sistema muestra los clientes adeudados y pagados 4. El usuario selecciona la opción imprimir orden de cobro 5. El sistema imprime la orden de cobro del cliente
Flujo Alternativo	Editar Factura

	<ol style="list-style-type: none"> 1. En el paso 3 del flujo normal, si el usuario selecciona editar. 2. El sistema despliega un formulario para agregar nuevos productos al recibo. 3. El usuario realiza los cambios pertinentes en la factura. 4. El sistema verifica que la información sea correcta y actualiza la información. <p>Registro de Pago</p> <ol style="list-style-type: none"> 1. En el paso 1 del flujo normal, si el usuario selecciona cuentas por cobrar. 2. El sistema despliega un listado de los clientes que deben valores. 3. El usuario selecciona registrar pago. 4. El sistema despliega un formulario para registrar el pago 5. El usuario selecciona la forma de pago a realizar y registra los valores 6. El sistema verifica que la información sea correcta y actualiza la información. <p>Historial de Pago</p> <ol style="list-style-type: none"> 1. En el paso 1 del flujo normal, si el usuario selecciona cuentas por cobrar. 2. El sistema despliega un listado de los clientes que deben valores. 3. El usuario selecciona historial de pago realizado. 4. El sistema despliega un historial de pago del cliente
--	---

Tabla 24 Gestión de Cobros

3.4. DIAGRAMA DE SECUENCIAS

Se realizo 9 diagramas de secuencias relacionados con los requerimientos que solicita la empresa Quicknet para el desarrollo de la aplicación

3.4.1. DIAGRAMA DE SECUENCIA: INICIO DE SESIÓN

En este diagrama el usuario visualiza la pantalla de inicio de sesión, el usuario ingresara su usuario y contraseña, el controller valida los datos y envía a la base de datos si el formulario es correcto, la base de datos verificara los datos y permitirá el ingreso a la aplicación.

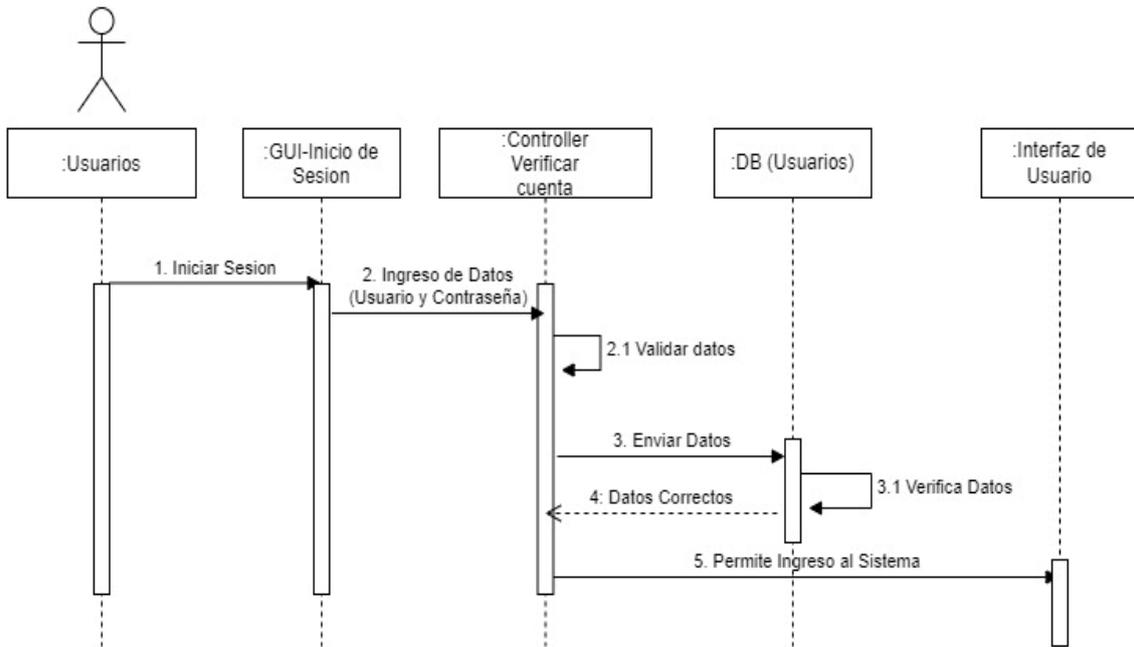


Figura 17 Diagrama de Secuencia: Inicio de Sesión

3.4.2. DIAGRAMA DE SECUENCIA: CREACIÓN DE USUARIO

En este diagrama el usuario realiza la creación de usuarios nuevos, el usuario visualiza el listado de usuarios creados, el usuario selecciona crear nuevo usuario permitiéndole registrar a los usuarios nuevos con sus respectivos permisos para el uso de la aplicación.

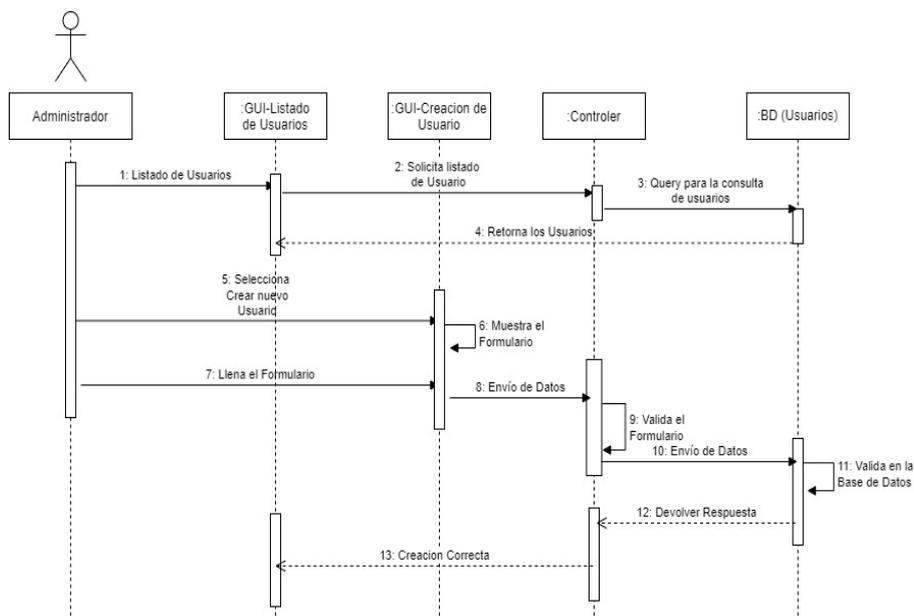


Figura 18 Diagrama de Secuencia: Creación de Usuario

3.4.3. DIAGRAMA DE SECUENCIA: CREACIÓN DE CLIENTES Y CONTRATOS

Este diagrama muestra cómo se registra clientes y contratos, el usuario seleccionará la opción creación de clientes, se desplegará el formulario permitiéndole ingresar un nuevo cliente y contrato.

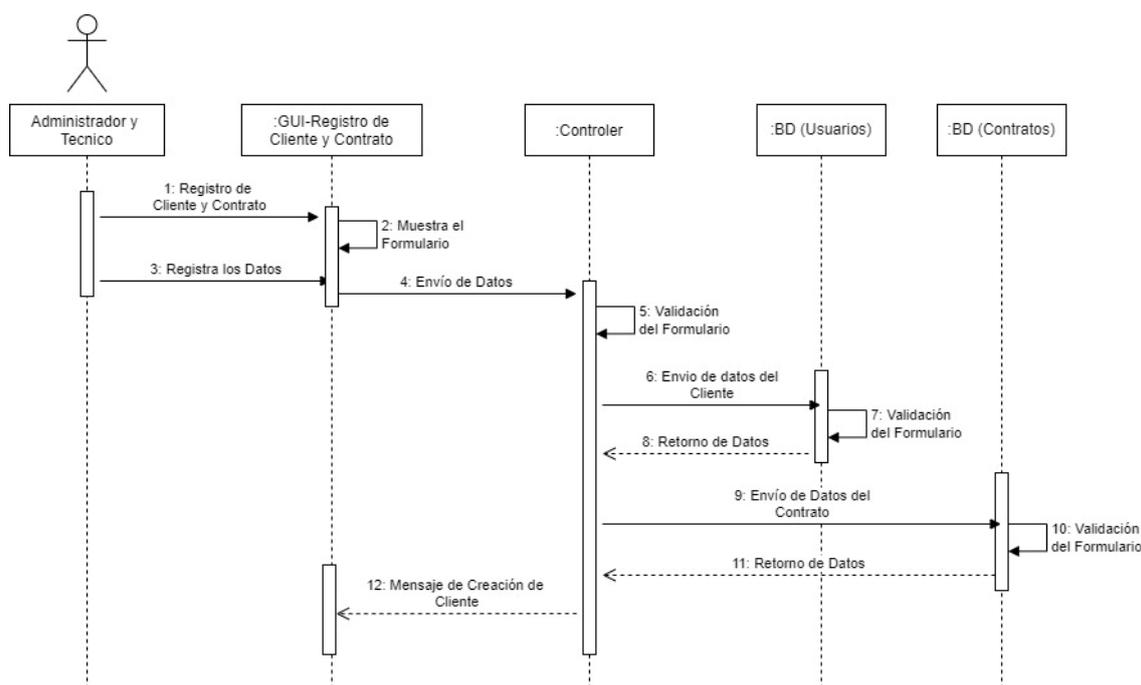


Figura 19 Diagrama de Secuencia: Creación de Clientes y Contratos

3.4.4. DIAGRAMA DE SECUENCIA: MODIFICACIÓN DE CONTRATOS

Este diagrama permite la modificación de contratos, el usuario podrá editar la información del contrato seleccionado, modificando la información que devuelve la base de datos mediante el formulario del detalle del contrato

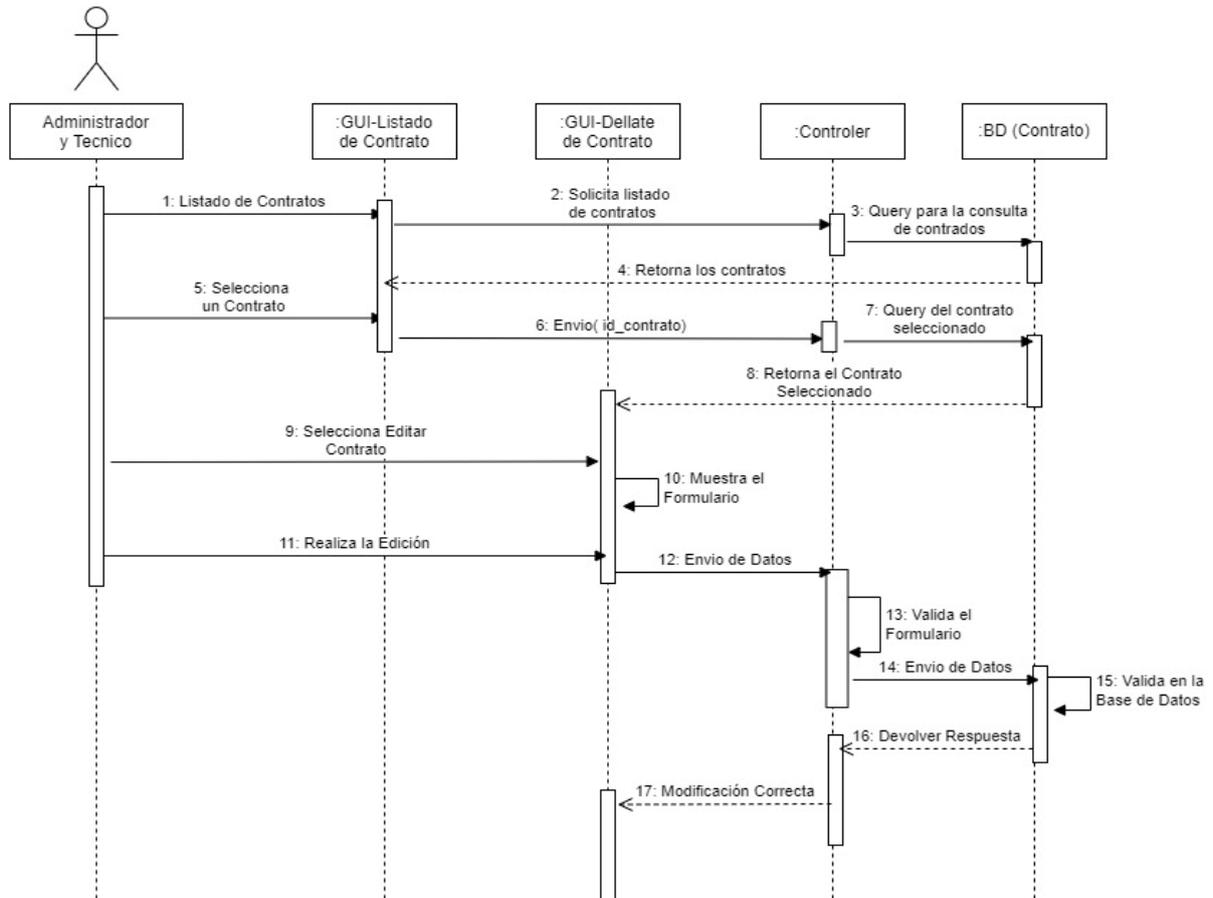


Figura 20 Diagrama de Secuencia: Modificación de Contratos

3.4.5. DIAGRAMA DE SECUENCIA: REGISTRO DE MATERIALES

Este diagrama muestra cómo se registra nuevos materiales de las ordenes de instalación, previamente el usuario tendrá que a ver registrado un cliente nuevo con su contrato, automáticamente por cada cliente nuevo se registrará una orden de instalación permitiéndole al usuario el ingreso de materiales utilizados en las instalaciones del servicio de internet.

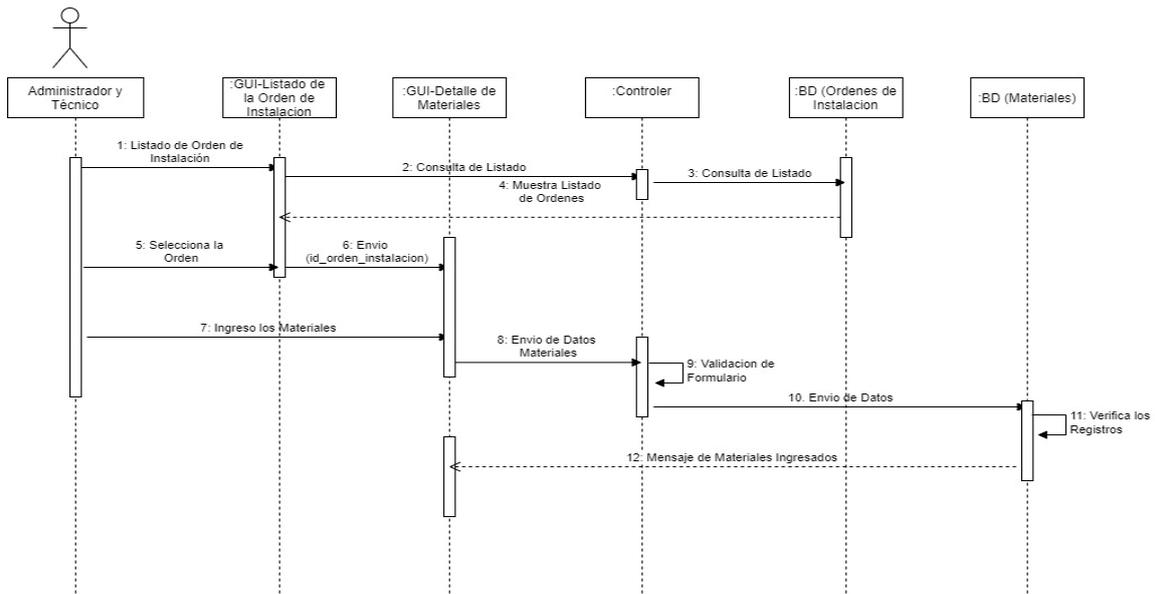


Figura 21 Diagrama de Secuencia: Registro de Materiales

3.4.6. DIAGRAMA DE SECUENCIA: REGISTRO DE PLANES

En este diagrama el usuario muestra cómo se crearía un nuevo plan, teniendo en cuenta que los planes son establecidos por la empresa Quicknet.

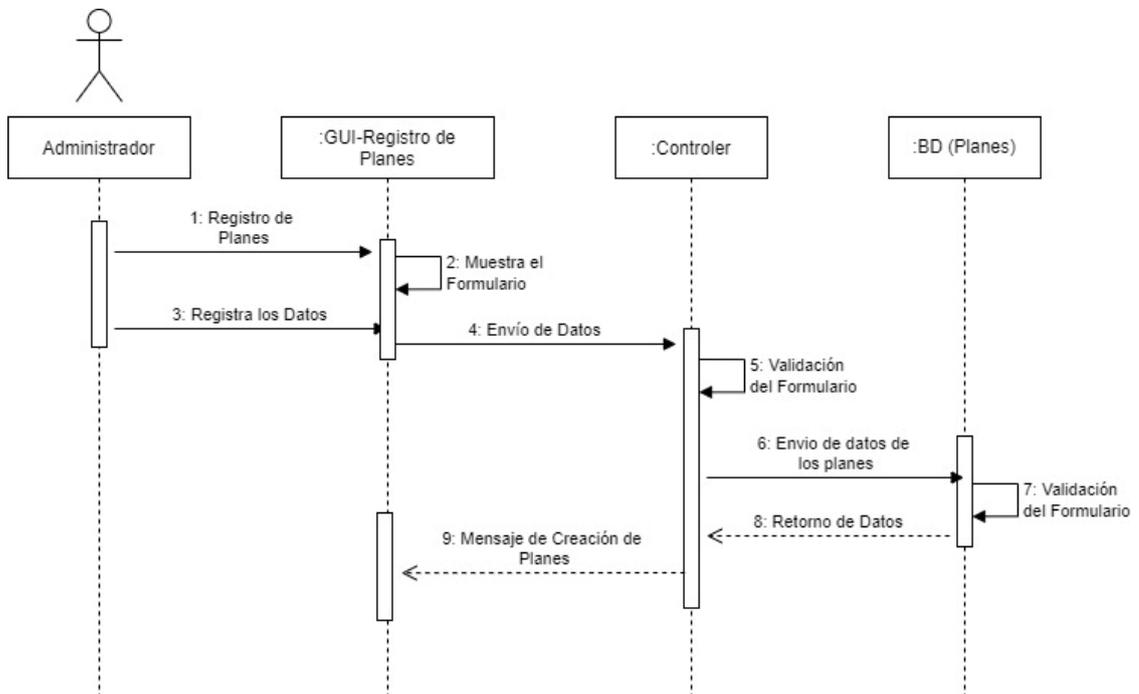


Figura 22 Diagrama de Secuencia: Registro de Planes

3.4.7. DIAGRAMA DE SECUENCIA: REGISTRO DE COMPRAS

En este diagrama muestra la secuencia de cómo se realiza el registro de una compra, el usuario en la primera pantalla visualiza el listado de compras realizadas, selecciona la opción creación de compras lo que al usuario se desplegará un formulario para el ingreso de la factura pertinente a la compra, previamente se deberá tener registrado los productos, las marcas, los proveedores para poder registrar una compra correctamente

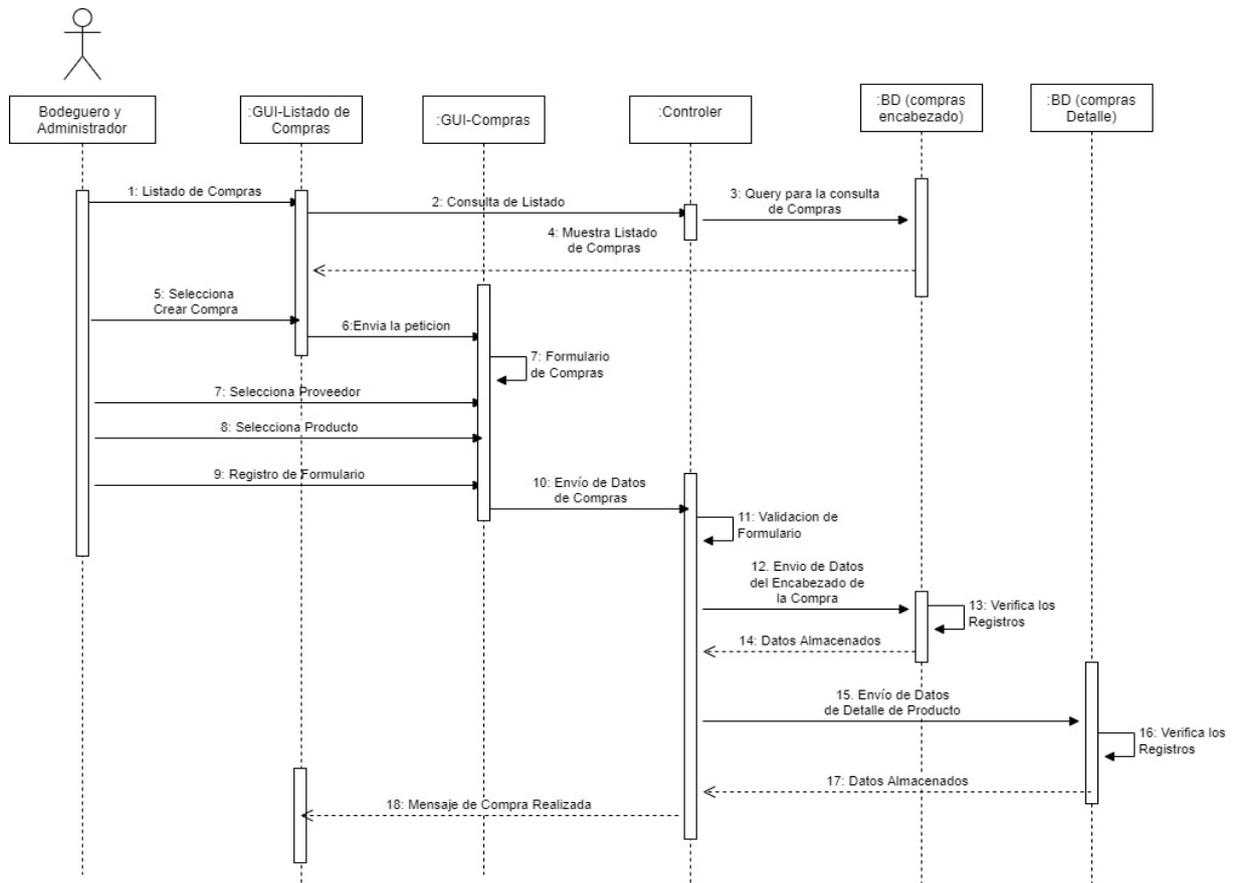


Figura 23 Diagrama de Secuencia: Registro de Compras

3.4.8. DIAGRAMA DE SECUENCIA: IMPRESIÓN DE RECIBO DE COBRO

En este diagrama de secuencia, el usuario visualiza todas las personas que tienen saldos pendientes permitiendo descargar los recibos de cobros en formato pdf.

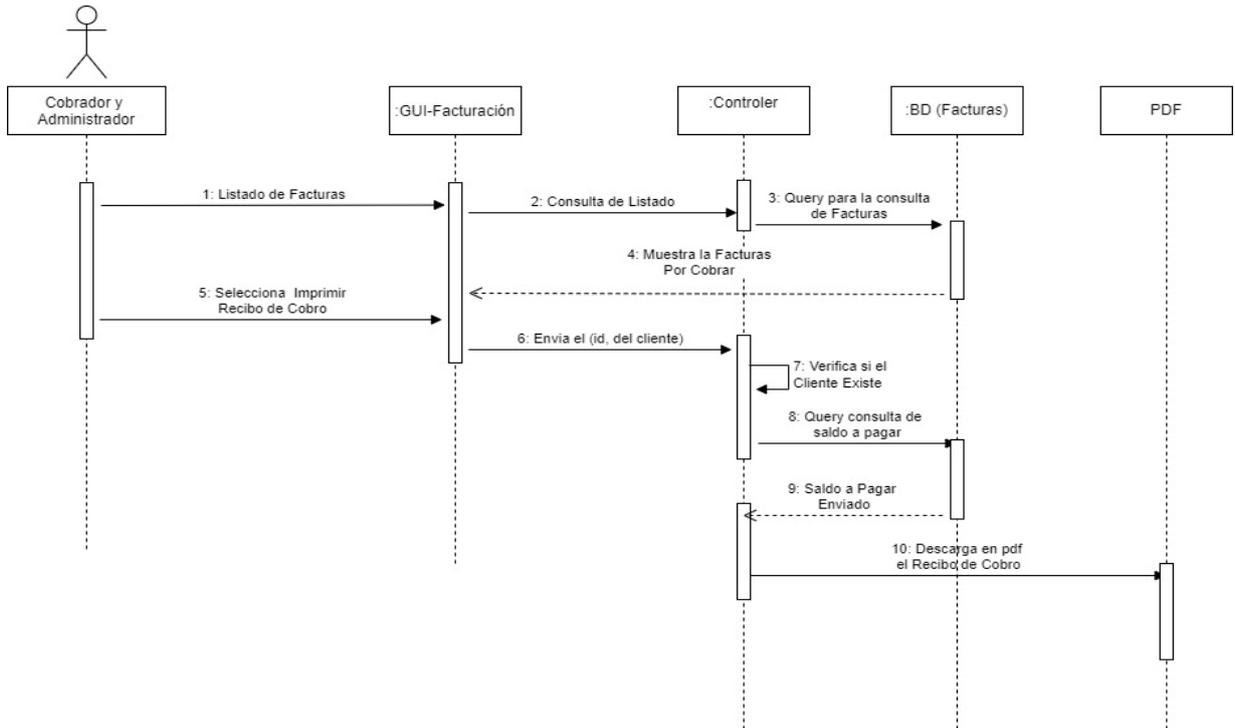


Figura 24 Diagrama de Secuencia: Impresión de Recibo de Cobro

3.4.9. DIAGRAMA DE SECUENCIA: REGISTRO DE PAGO

En este diagrama de secuencia el usuario mediante un formulario realiza los registros de pago de los clientes para que automáticamente se genere el recibo de cobro del siguiente mes una vez registrado el pago

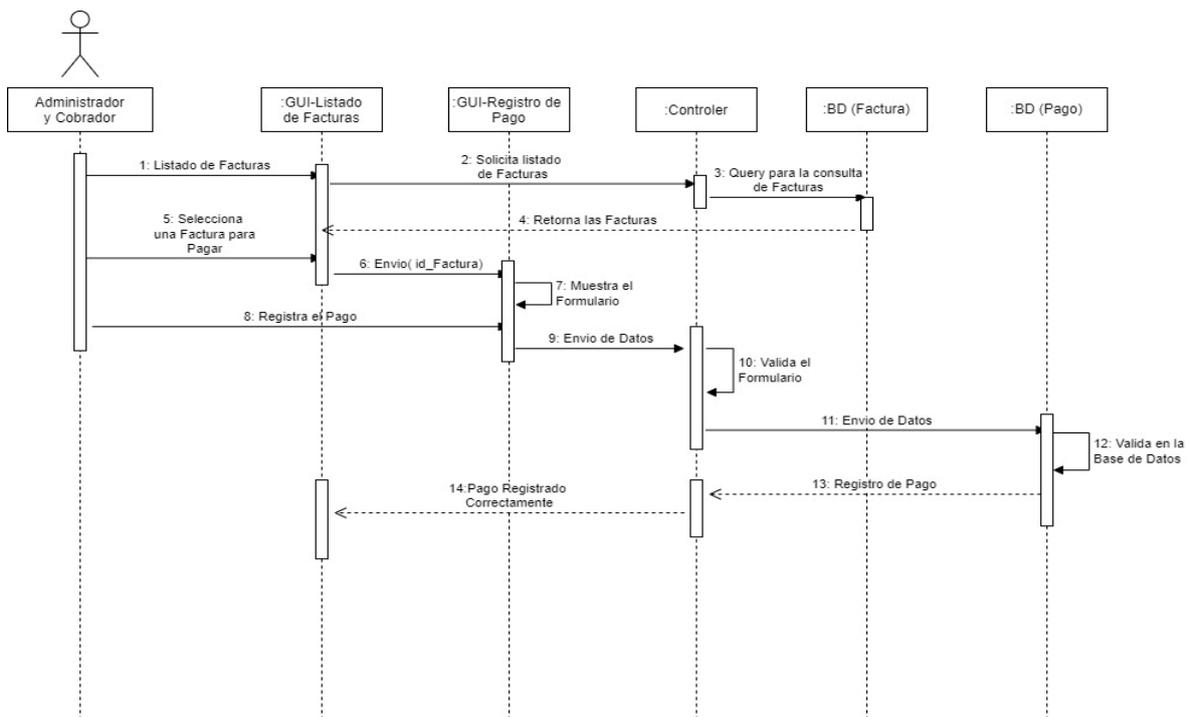


Figura 25 Diagrama de Secuencia: Registro de Pago

3.5. DIAGRAMA DE CLASE

En este diagrama describimos la estructura del sistema con sus clases que se utilizara, muestra como está desarrollado el aplicativo informático

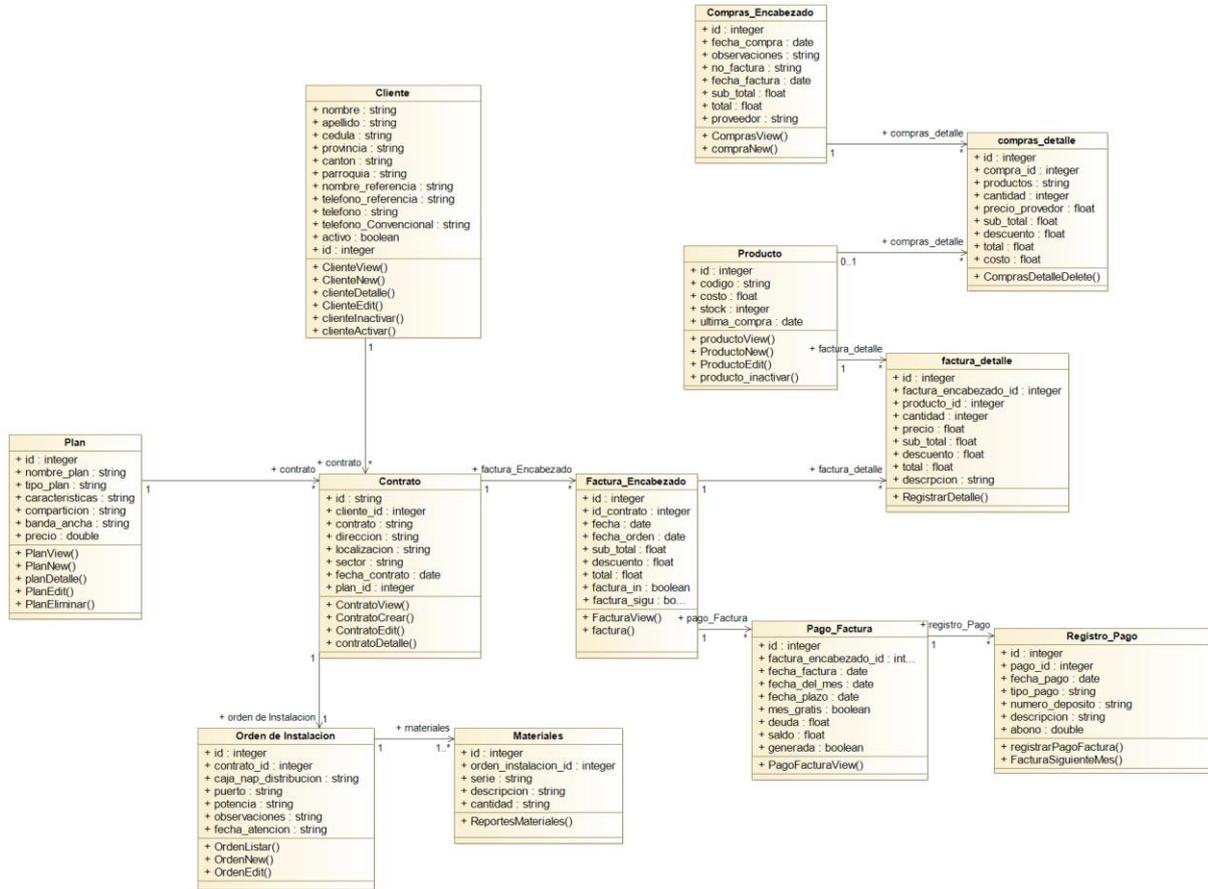


Figura 26 Diagrama de Clases

3.6. BASE DE DATOS DE LA APLICACIÓN

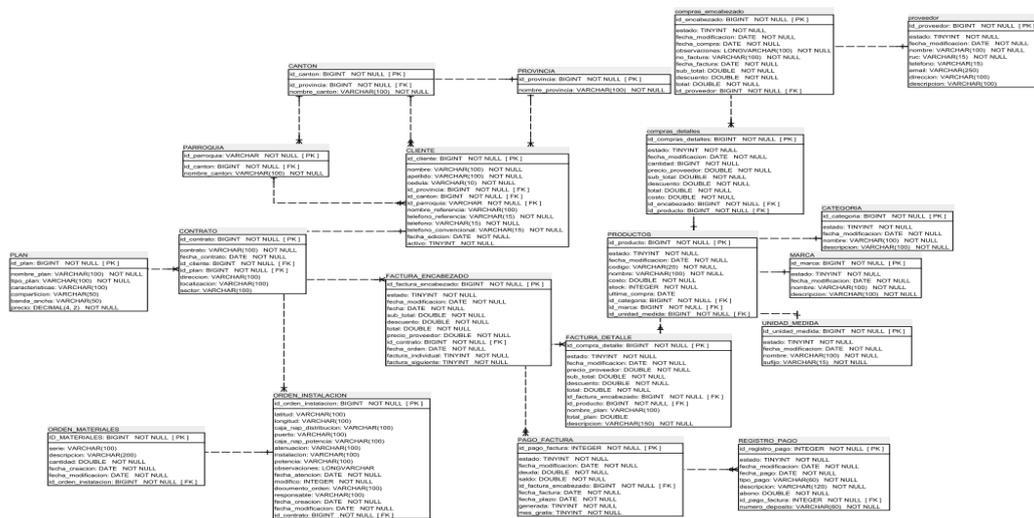


Figura 27 Diagrama de la Base de Datos

A continuación, se detalla las tablas, claves primarias y foráneas, el tipos de datos que conforman la base de datos donde se almacenará toda la información del sistema de cobros de la empresa Quicknet.

Tabla clientes: Esta Tabla almacena la información de los clientes que disponen del servicio de internet.

Campo	Primary Key	Forense Key	Tipo Dato	Descripción
id	X		integer	Id de la tabla clientes
Nombre			Varchar	Nombre del cliente
Apellido			Varchar	Apellido del cliente
Cedula			Varchar	Cedula del cliente
Provincia_id			integer	Provincia donde vive el cliente
Cantón_id			integer	Cantón donde vive el cliente
Parroquia_id			integer	Parroquia donde vive el cliente
Nombre_referencia			Varchar	Nombre de una persona de referencia cercana del cliente
Apellido_referencia			Varchar	Apellido de la persona de referencia del cliente
Teléfono_referencia			Varchar	Teléfono de la persona de referencia del cliente
Teléfono			Varchar	Teléfono celular del cliente
Teléfono_convencional			Varchar	Teléfono convencional del cliente
Fecha_edicion			date	Fecha en la que se edita el cliente
Activo			bool	Estado del cliente si se encuentra activo o inactivo

Tabla 25 Tabla Clientes

Tabla Contrato: Esta tabla almacena la información del contrato de cada cliente

Campo	Primary Key	Forense Key	Tipo Dato	Descripción
id	X		integer	Id de la tabla contrato
Fecha_contrato			Date	Fecha en la que se firmó el contrato
Cliente_id		X	Integer	Cliente al que pertenece el contrato
Plan_id		X	Integer	Plan que pertenece al contrato
Dirección			Varchar	Dirección donde se instala el servicio de internet
Localización			Varchar	Lugar donde reside el cliente
Sector			Varchar	Sector donde vive el cliente
Contrato			Varchar	Numero de contrato único de cada cliente

Tabla 26 Tabla Contrato

Tabla Plan: Esta tabla Almacena la información de los planes.

Campo	Primary Key	Forense Key	Tipo Dato	Descripción
id	X		integer	Id de la tabla plan
Nombre_plan			Varchar	Nombre del plan
Tipo_plan			Varchar	Tipo de plan
Características			Varchar	Características del plan
Compartición			Varchar	Tipo de compartición del plan
Banda_ancha			Varchar	Banda ancha del plan
Precio			decimal	Valor al público del plan

Tabla 27 Tabla Plan

Tabla Parroquia: Esta tabla Almacena la información de las parroquias.

Campo	Primary Key	Forense Key	Tipo Dato	Descripción
id	X		integer	Id de la tabla parroquia
Nombre_parroquia			Varchar	Nombre de la parroquia
Cantón_id		X	integer	Cantón

Tabla 28 Tabla Parroquia

Tabla Provincia: Esta tabla almacena la información de las provincias

Campo	Primary Key	Forense Key	Tipo Dato	Descripción
id	X		integer	Id de la tabla provincia
nombre_provincia			Varchar	Nombre de la provincia

Tabla 29 Tabla Provincia

Tabla Cantón: Esta tabla Almacena la información de los cantones.

Campo	Primary Key	Forense Key	Tipo Dato	Descripción
id	X		integer	Id de la tabla cantón
nombre_canton			Varchar	Nombre del cantón
provincia_id		X	integer	Provincia

Tabla 30 Tabla Cantón

Tabla de Materiales: Esta tabla Almacena la información de los materiales usados en la instalación.

Campo	Primary Key	Forense Key	Tipo Dato	Descripción
id	X		integer	Id de la tabla materiales
Serie			Varchar	Serie o Mac del equipo instalado
Descripción			Varchar	Descripción del equipo instalado
Fecha_creacion			date	Fecha de creación del registro
Fecha_modificacion			date	Fecha en la que se modifica el registro
Orden_instalacion_id			integer	Relación con la tabla orden de instalación
Cantidad			integer	Cantidad de los materiales instalados

Tabla 31 Tabla Materiales

Tabla Orden de Instalación: Esta tabla almacena la información de las ordenes de instalación de los clientes.

Campo	Primary Key	Forense Key	Tipo Dato	Descripción
id	X		integer	Id de la tabla orden de instalación
Latitud			Varchar	Latitud del lugar de instalación del cliente
Longitud			Varchar	Longitud del lugar de instalación del cliente
Caja_nap_distribucion			Varchar	Número que se registra a una caja nao
Puerto			Varchar	Numero de puerto de la instalación de internet
Caja_nap_potencia			Varchar	Caja nap de potencia
Atenuación			Varchar	Atenuación del equipo

Instalación			Varchar	Tipo de instalación realizada
Potencia			Varchar	Potencia de la red
Observaciones			Varchar	Registro donde se pone las observaciones de la instalación
Fecha_atencion			date	Fecha donde el servicio de internet fue instalado
Modifico			integer	Valor que indica si el registro fue modificado
Responsable			Varchar	Nombre de responsable de la instalación
Fecha_creacion			date	Fecha de la creación del registro
Fecha_modificacion			date	Fecha de la modificación del registro
Contrato_id		X	integer	Relación con el contrato del cliente

Tabla 32 Tabla Ordenes de instalación

Tabla Factura de Encabezado: Esta tabla Almacena los encabezados de las facturas de los clientes.

Campo	Primary Key	Forense Key	Tipo Dato	Descripción
id	X		integer	Id de la tabla clientes
Estado			bool	Estado del encabezado
Fecha_modificacion			Date	Fecha de modificación del registro
Fecha			Date	Fecha en el que se creó el registro
Sub_total			Float	Subtotal de factura del cliente
Descuento			Float	Descuento de la factura
Total			Float	Total, de la suma del subtotal + el descuento
Contrato_id		X	Integer	Relación con la tabla contrato
Fecha_orden			Date	Fecha para visualizar el mes
Factura_individual			Bool	Verifica si la factura es individual
Factura_siguiente			bool	Verifica si la factura del próximo mes ya fue creada

Tabla 33 Tabla de la Factura de Encabezado

Tabla Factura del Detalle: Esta tabla Almacena la información del detalle de las facturas de los clientes.

Campo	Primary Key	Forense Key	Tipo de Dato	Descripción
id	X		integer	Id de la tabla clientes
Estado			Bool	Estado de la tabla
Fecha_modificacion			Date	Fecha en la que se modificó el contrato
Cantidad			Integer	Cantidad de productos comprados
Precio			Float	Valor individual de cada producto
Sub_total			Float	Subtotal individual de cada producto
Descuento			Float	Descuento individual de cada producto
Total			Float	Total, de cada producto
Factura_id			Integer	Relación a la factura que pertenece
Producto_id		X	Integer	Productos que se selecciona a la tabla producto
Nombre_plan			Varchar	En caso de ser solo plan se mostrará el nombre del plan correspondiente
Descripción			Varchar	Descripción de cada ítem

Tabla 34 Tabla Detalle de la Factura

Tabla Pago de la Factura: Esta tabla almacena la información de los pagos de la factura de los clientes.

Campo	Primary Key	Forense Key	Tipo de Datos	Descripción
id	X		integer	Id de la tabla pago factura
Estado			Bool	Estado de la tabla
Fecha_modificacion			Date	Fecha en la que se realizó la modificación
Deuda			Float	Deuda del valor a pagar
Saldo			Float	Saldo que tiene que pagar el servicio de internet
Factura_id		X	Integer	Relación con el encabezado de la factura

Fecha_factura			Date	Fecha en la que se imprimirá el orden de cobro
Fecha_plazo			Date	Fecha plazo a pagar la orden de cobro
Fecha_del_mes			Date	Fecha del mes que se realizara el corte
Generada			Bool	Valor que indica si el recibo fue generado
Mes_gratis			Bool	Opción para indicar si el cliente tiene mes gratis

Tabla 35 Tabla de los Pagos de la Factura

Tabla Registro Pago: Esta tabla almacena el historial de pagos de los clientes

Campo	Primary Key	Forense Key	Tipo de Datos	Descripción
Id			integer	Id de la tabla clientes
Estado			Bool	Estado de la tabla
Fecha_modificacion		X	Date	Fecha en que se modificó la tabla
Fecha_pago			date	Fecha del pago
Descripción			Varchar	Descripción del pago
Abono			Float	Valor abonado
Pago_id		X	Integer	Relación con la tabla Pago
Numero_deposito			Varchar	Numero de deposito
Tipo_pago			Varchar	Tipo de pago a realizar

Tabla 36 Tabla de los Registros de Pago de los Recibos de Cobro

Tabla Productos: Esta tabla almacena la información de los productos adquiridos

Campo	Primary Key	Forense Key	Tipo de Datos	Descripción
id	X		integer	Id de la tabla productos
Estado			Bool	Estado de la tabla
Fecha_modificacion			Date	Fecha en la que modifica la tabla
Codigo			Varchar	Código del producto específico
Nombre			Varchar	Nombre del producto
Costo			Float	Costo del que se adquirió el producto
Stock			Integer	Stock del producto
Ultima_compra			Date	Fecha de la última compra
Categoría_id		X	Int	Relación con la tabla categoría

Marca_id		X	Int	Relación con la tabla marca
Unidad_medida_id		X	Int	Relación con la tabla unidad de medida

Tabla 37 Tabla de Productos

Tabla Categoría: Esta tabla almacena las categorías para usarlos en los productos

Campo	Primary Key	Forense Key	Tipo de Datos	Descripción
id	X		integer	Id de la tabla categoría
Estado			Bool	Estado de la tabla
Fecha_modificacion		X	Date	Fecha en la que se modifica
Nombre			Varchar	Nombre de la categoría
descripción			Varchar	Descripción de la categoría

Tabla 38 Tabla de Categorías

Tabla Marca: Esta tabla almacena las marcas para usarlos en los productos

Campo	Primary Key	Forense Key	Tipo de Datos	Descripción
id	X		integer	Id de la tabla Marca
Estado			Bool	Estado de la tabla
Fecha_modificacion		X	Date	Fecha en la que se modificó la marca
Nombre			Varchar	Nombre de la marca
Descripción			Varchar	Descripción de la marca

Tabla 39 Tabla de Marcas

Tabla Unidad de Medida: Esta tabla almacena las unidades de medida para usarlos en los productos

Campo	Primary Key	Forense Key	Tipo de Datos	Descripción
id	X		integer	Id de la tabla unidad de medida
Estado			Bool	Estado de la tabla
Fecha_modificacion		X	Date	Fecha en la que se modifíco
Nombre			Varchar	Nombre de la unidad de medida
sufijo			Varchar	Sufijo de la unidad de medida

Tabla 40 Tabla de las Unidades de Medida

Tabla Detalles de la Compra: Esta tabla almacena los detalles de los productos de la compra

Campo	Primary Key	Forense Key	Tipo de Datos	Descripción
Id	X		integer	Id de la tabla clientes
Estado			Bool	Estado de la tabla
Fecha_modificacion			Date	Fecha en la que se modifica el registro
Cantidad			Int	Cantidad de productos comprados
Precio_proveedor			Float	Precio que nos vende el proveedor
Sub_total			Float	Subtotal de la compra
Descuento			Float	Descuento de la compra
Total			Float	Total, de la compra
Costo			Float	Costo de venta de la compra
Compra_id		X	Integer	Relación con la tabla compras
Productos_id		X	integer	Relación con la tabla productos

Tabla 41 Tabla Detalles de la Compra

Tabla Encabezado de Compras: Esta tabla almacena los datos de la factura de la compra

Campo	Primary Key	Forense Key	Tipo de Datos	Descripción
id	X		integer	Id de la tabla encabezado compra
estado			Bool	Estado de la tabla
Fecha_modificacion		X	Date	Fecha en la que se realiza alguna modificación
Fecha_compra			Date	Fecha en la que se realiza la compra
Observaciones			Varchar	Observaciones de la compra
No_factura			Varchar	Numero de factura de la compra
Fecha_factura			Date	Fecha de la factura de la compra
Sub_total			Float	Subtotal de la compra
Descuento			Float	Descuento de la compra

Total			Float	Valor total de la compra realizada
Proveedor_id		X	Integer	Proveedor al que se realiza la compra

Tabla 42 Tabla del Encabezado de la Compra

Tabla Proveedores: Esta Tabla registra los proveedores a lo que nosotros realizamos la compra de materiales.

Campo	Primary Key	Forense Key	Tipo de Datos	Descripción
id	X		integer	Id de la tabla proveedor
Estado			Bool	Estado de la tabla
Fecha_modificacion		X	Date	Fecha de la modificación
Nombre			Varchar	Nombre del proveedor
Ruc			Varchar	Ruc del proveedor
Teléfono			Varchar	Teléfono del proveedor
Email			Varchar	Correo electrónico del proveedor
Dirección			Varchar	Dirección del proveedor
descripción			Varchar	Descripción del proveedor

Tabla 43 Tabla de Proveedores

3.7. PROTOTIPO DE LA INTERFAZ DEL SISTEMA INFORMATICO WEB

El diseño del prototipo de la aplicación nos ayuda a tener una idea más clara de cómo será nuestro sistema al finalizarlo, teniendo en cuenta la facilidad de manejo y la funcionalidad. Para el diseño del prototipo se usó una aplicación llamada Draw.io, esta aplicación nos proporciona diferentes herramientas para la creación de prototipos de interfaz, diagramas, entre otras utilidades.

3.7.1. PROTOTIPO DEL LOGIN

Al abrir nuestro sistema informático web, se contemplará la pantalla de inicio de sesión, esta pantalla permitirá al usuario ingresar sus credenciales para poder continuar con la pantalla principal

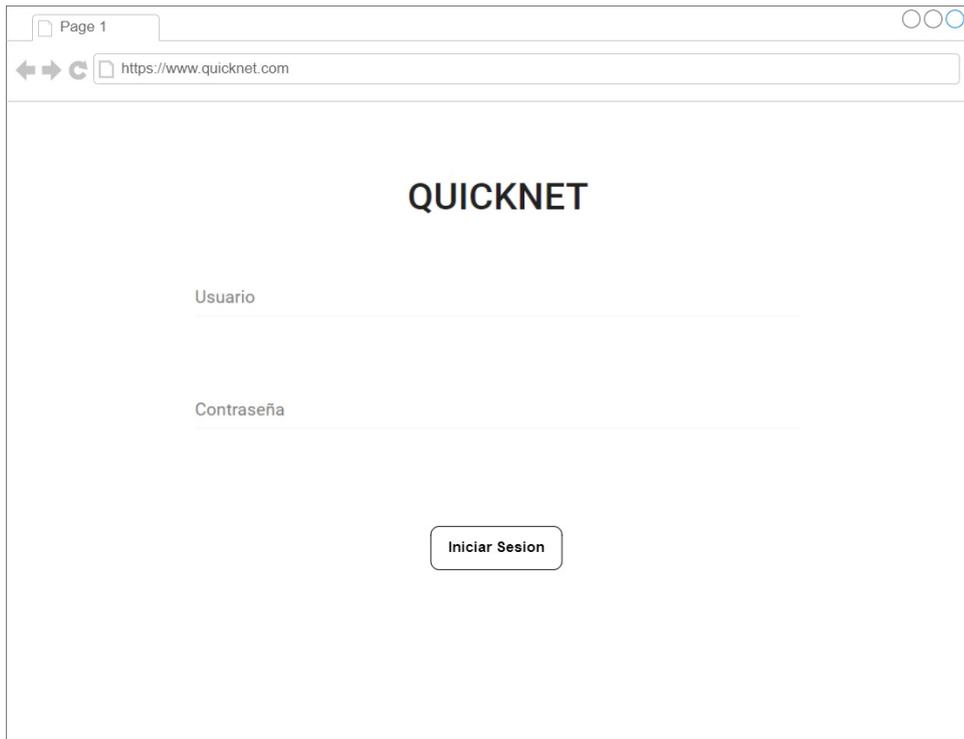


Figura 28 Diseño del Login

3.7.2. PROTOTIPO DE LA PANTALLA DE INICIO

Al iniciar sesión, se muestra la pantalla principal, esta se encuentra conformada por una barra lateral en el lado izquierdo y un contenedor principal. En la barra lateral izquierda se encuentra ubicado el menú de la aplicación y en el contenedor mostrara el total de los clientes por provincia y el total de clientes en general.

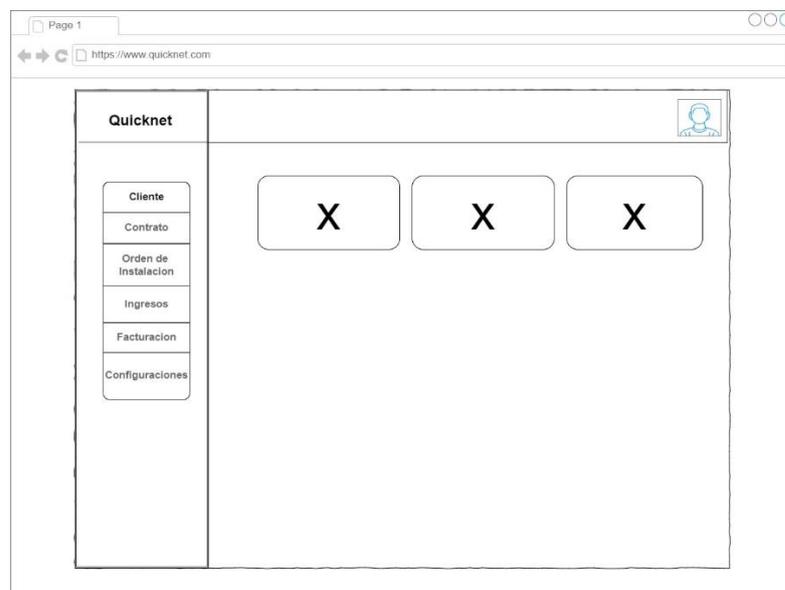


Figura 29 Prototipo de la Pantalla de Inicio

3.7.3. PROTOTIPO CREACIÓN DE CLIENTES Y CONTRATO

Dentro de la interfaz creación de clientes, el usuario llenara el formula con los datos necesarios del cliente, en este apartado también se registra un nuevo contrato pertinente al cliente registrado.

El prototipo muestra una interfaz web con un navegador que muestra 'Page 1' y 'https://www.quicknet.com'. A la izquierda hay un menú con las opciones: Cliente, Contrato, Orden de Instalacion, Ingresos, Facturacion y Configuraciones. El contenido principal está dividido en dos secciones: 'Registrar Cliente' y 'Registrar Contrato'. La sección 'Registrar Cliente' incluye campos para Nombre, Apellido, Cedula, Telefono Celular, Telefono Convencional, Provincia, Canton, Parroquia, Nombre de Referencia, Apellido de Referencia y Celular de Referencia. La sección 'Registrar Contrato' incluye campos para Numero Contrato, Fecha Contrato, Plan y un botón 'Guardar'. Todos los campos de texto tienen 'Messina Cake' como valor de ejemplo, excepto 'Fecha Contrato' que muestra 'Your name' con un ícono de calendario y 'Plan' que es un menú desplegable con 'Option 1'.

Figura 30 Prototipo Creación de Clientes y Contrato

3.7.4. PROTOTIPO LISTADO DE CLIENTES

Dentro del menú clientes, el usuario podrá observa el listado de clientes con los detalles de los clientes creados.

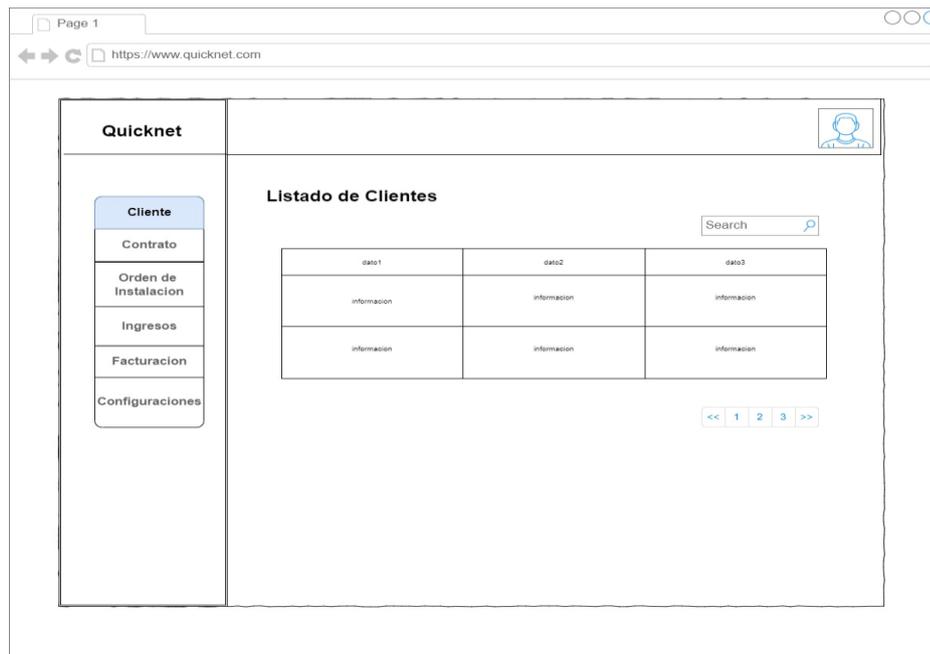


Figura 31 Prototipo Listado de Clientes

3.7.5. PROTOTIPO LISTADO DE CONTRATO

Dentro del menú de contrato, el usuario observara el listado de contratos que almacena la aplicación.

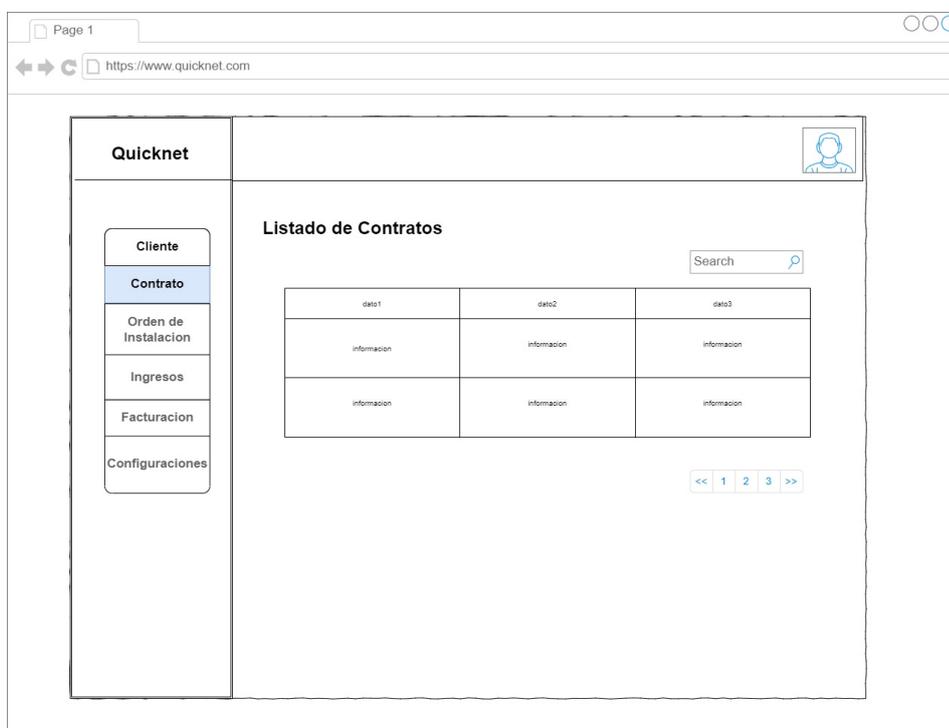


Figura 32 Prototipo Listado de Contrato

3.7.6. PROTOTIPO DE REGISTRO DE MATERIALES

El usuario después de seleccionar el ingreso de materiales de un cliente, el usuario visualizara el formulario donde registrara los materiales que usó el técnico para la instalación del servicio de internet.

El prototipo muestra una interfaz web con un navegador que muestra 'Page 1' y 'https://www.quicknet.com'. A la izquierda hay un menú de navegación con los siguientes ítems: 'Cliente', 'Contrato', 'Orden de Instalacion' (destacado en azul), 'Ingresos', 'Facturacion' y 'Configuraciones'. El contenido principal se divide en dos secciones: 'Quicknet' y 'Registro de Materiales'. En la sección 'Registro de Materiales', hay un campo de texto para 'Datos del Cliente'. Debajo de esto, hay dos campos de entrada para 'Latitud' y 'Longitud'. A continuación, hay una sección 'Materiales' con tres campos de entrada etiquetados como 'Serie', 'Descripcion' y 'Materiales'. En la parte inferior izquierda, hay un campo para 'Fecha de Instalacion' con un ícono de calendario. En la parte inferior derecha, hay un botón 'Guardar'.

Figura 33 Prototipo Listado de Registro de Materiales

3.7.7. PROTOTIPO DE REGISTRO DE COMPRAS

Dentro de esta interfaz se seleccionará el proveedor, el usuario ingresará la fecha en la que se realizó la compra, ingresa el numero de la factura, se ingresa observaciones en caso de existir, se seleccionará los productos de la compra y para finalizar se ingresará la cantidad, descuento y el precio de la compra

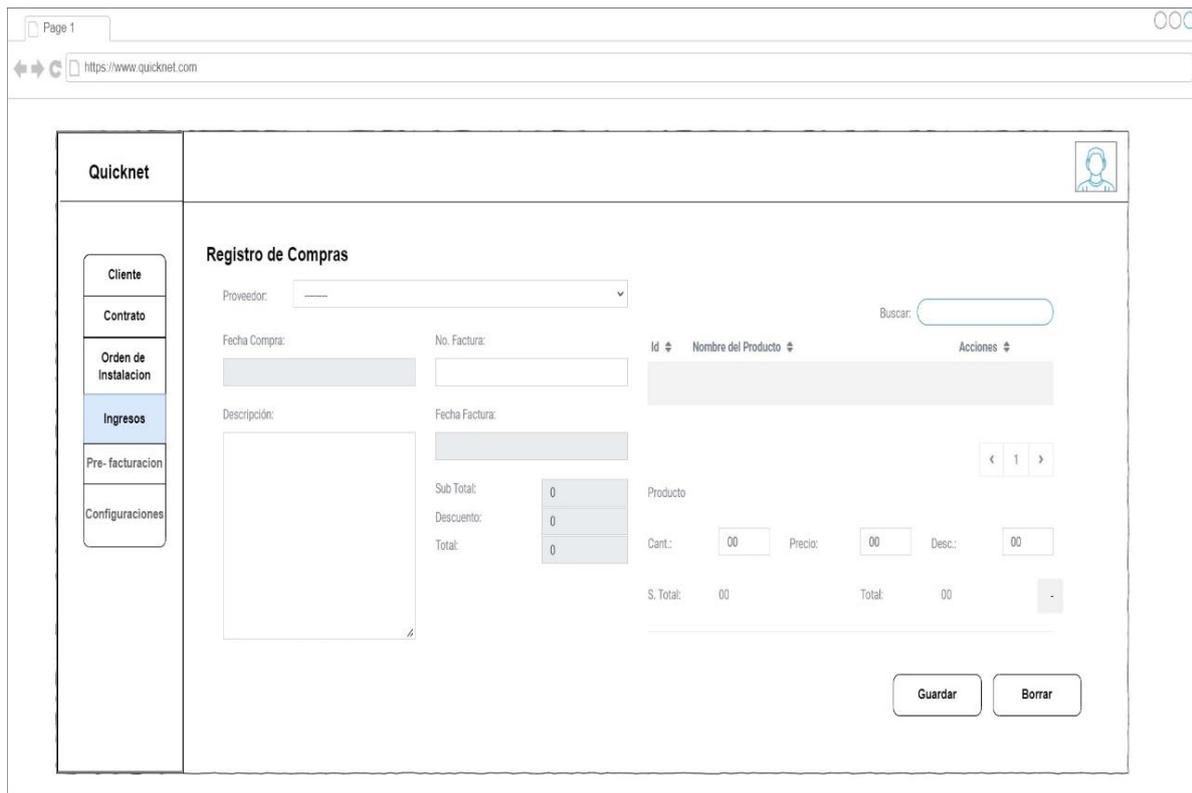


Figura 34 Prototipo de Registro de Compras

3.7.8. PROTOTIPO DE LISTADO DE PRE-FACTURAS

Dentro de esta interfaz se mostrará el listado de pre-facturas de los clientes que tienen valores pendientes, aquí el usuario tendrá la opción de imprimir los recibos de cobro de los clientes

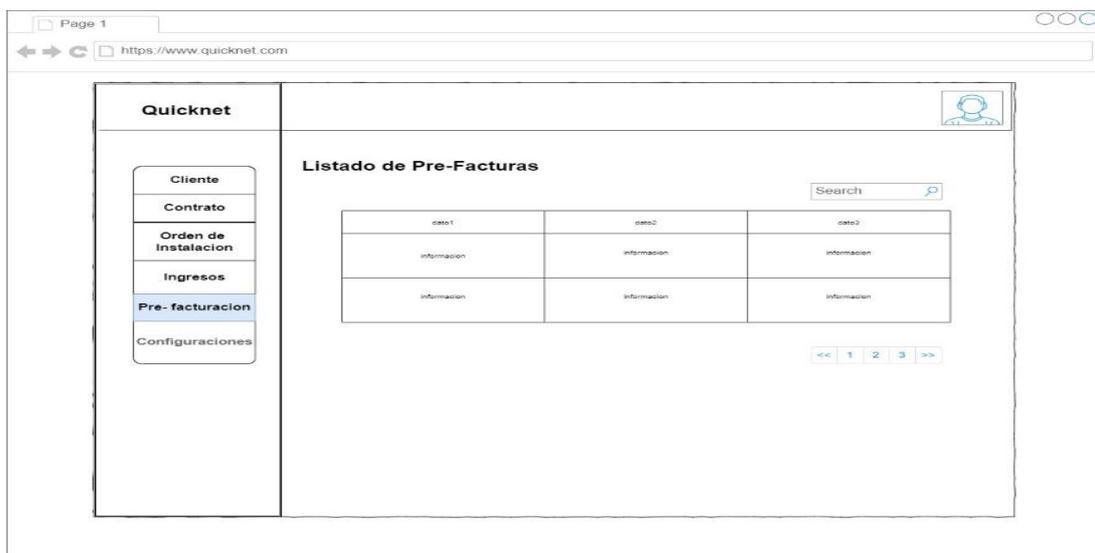
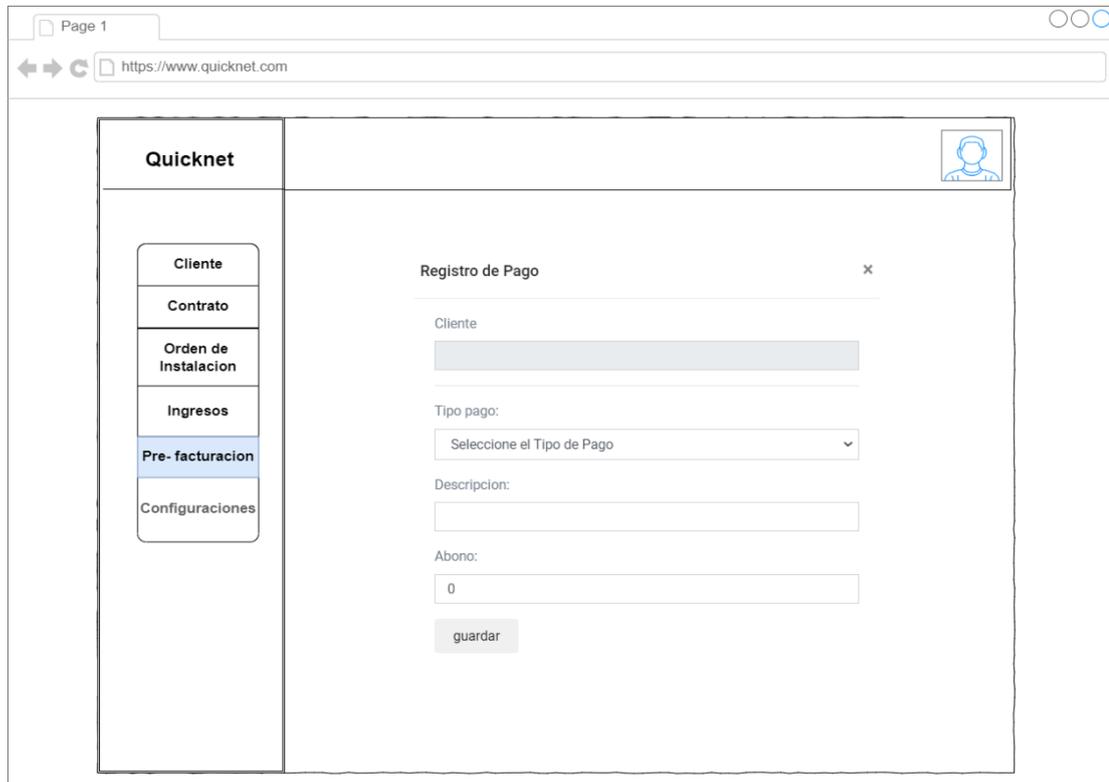


Figura 35 Prototipo de Listado de Pre-facturas

3.7.9. PROTOTIPO REGISTRO DE PAGO

Dentro de este prototipo se muestra un formulario para registrar el pago del cliente seleccionado, seleccionaremos el tipo de pago, agregaremos una descripción y establecemos el valor pagado por el cliente en el input del abono.



The image shows a web browser window with the URL 'https://www.quicknet.com'. The page title is 'Page 1'. The main content area is titled 'Quicknet' and features a sidebar menu with the following items: 'Cliente', 'Contrato', 'Orden de Instalacion', 'Ingresos', 'Pre- facturacion' (highlighted in blue), and 'Configuraciones'. The main content area displays a modal window titled 'Registro de Pago' with a close button 'x'. The modal contains the following fields: 'Cliente' (text input), 'Tipo pago:' (dropdown menu with the text 'Seleccione el Tipo de Pago'), 'Descripcion:' (text input), and 'Abono:' (text input with the value '0'). A 'guardar' button is located at the bottom of the modal.

Figura 36 Prototipo de Registro de Pago

3.7.10. PROTOTIPO DE LA ORDEN DE COBRO (PDF)

Este prototipo es el formato que se imprimirá las ordenes de cobro para entregárselas al cliente, mostrará la información del cliente, la fecha de emisión y la orden de cobro, después tendrá una tabla con los detalles de los valores a pagar.

ORDEN DE COBRO

Cedula:
Nombre y Apellido :
Mes de Cobro :

No de Contrato:
Fecha de Emision:

ITEM	DESCRIPCION	CANTIDAD	P. UNITARIO	P. TOTAL
1		1		
2		1		

Valor a Pagar :

Efectivo ___ Transferencia ___ Deposito ___

RECIBI CONFORME **ENTREGUE**

FIRMA
NOMBRE

FIRMA
NOMBRE

Figura 37 Prototipo de la Orden de Cobro (pdf)

4. CONSTRUCCION Y PRUEBAS

En este capítulo se hablaremos sobre la construcción del sistema informático web describiendo las tecnologías y herramientas que se utilizara en el transcurso del desarrollo, a su vez, mostraremos fragmentos de código que se utilizó en el desarrollo del aplicativo.

Para el desarrollo del sistema informático web utilizaremos un framework muy popular llamado Django, este framework facilita la creación de aplicaciones dinámicas de una manera rápida y segura, Django es compatible con la gran mayoría de base de datos ofreciendo agilidad a la hora de configurar la conexión y, algo interesante es el sitio de administración de Django, una ventaja grande contra otros framework, este sitio de administración es muy potente ayudando a gestionar los modelos creados sin la necesidad de interactuar directamente con la base de datos.

4.1. HERRAMIENTAS Y TECNOLOGÍAS

A continuación, se muestra las diversas tecnologías y herramientas que se usara para la creación del aplicativo web.

4.1.1. Python 3.9.2

Python es un lenguaje de programación de alto nivel multiparadigma, soporta la programación imperativa, funcional, programación orientada a objetos con una semántica dinámica integrada. Su licencia es de código abierto administrado por Python Software Foundation, esta licencia fue aprobada por la Open Source Initiative (OSI).

4.1.2. Django 3.2

Django es uno de los framework más populares en el desarrollo web de aplicación, se encuentra escrito en Python por lo que existe una gran comunidad en la creación de nuevas librerías y plugin para el uso de esta. Django ofrece un crecimiento brutal en el desarrollo de nuevas aplicaciones web teniendo encuentra su rapidez, seguridad y facilidad de programar código.

Herramientas usadas:

- PyPDF2 – Librería para la creación de pdf
- Sqlparse – librería que permite la conexión de una base de datos MySQL
- Bootstrap – Framework de diseño fron-end
- Bootstrap-tabla – Framework de diseño para las tablas
- JQuery – Librería de JavaScript
- django-widget-tweaks – librería para manejo de formularios
- import_export – libreria que permite la importación y exportación de información de la base de datos
- django_userforeignkey – librería que permite la creación de id de los modelos.

4.1.3. VALIDACIONES PROPIAS DE DJANGO

Django es un potente framework capas de tener sus propios métodos de validaciones sin la necesidad de programar todo el proceso de validación.

En los métodos llamados (validator), Django ofrece diversos métodos que ayuda en los procesos de validaciones. A continuación, detallo los más importantes y usados en este proyecto.

- EmailValidator. - permite las validaciones de los correos electrónicos
- MaxValueValidator. - valida el valor máximo en los campos
- MinValueValidator. – valida el valor mínimo en los campos

4.1.4. INTERFACES PRINCIPALES DEL SISTEMA

A continuación, explicare las interfaces principales del sistema tomando en cuenta las validaciones y código del aplicativo web.

4.1.4.1. INTERFAZ LOGIN DEL SISTEMA.

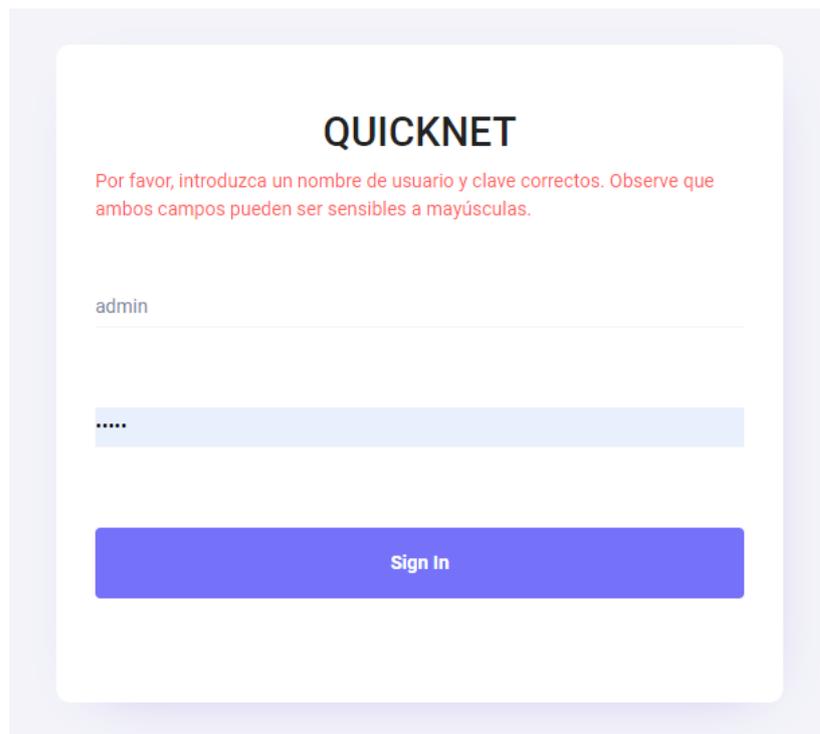


Figura 38 Interfaz, login del sistema

Esta pantalla muestra un formulario el que pedirá el usuario y contraseña, el formulario se encuentra validado para los siguientes puntos:

- Validación de usuario y contraseña
- Validación de usuario no existente.

```

def clean(self):
    username = self.cleaned_data.get("username")
    password = self.cleaned_data.get("password")

    if username is not None and password:
        self.user_cache = authenticate(
            self.request, username=username, password=password
        )
        if self.user_cache is None:
            raise self.get_invalid_login_error()
        else:
            self.confirm_login_allowed(self.user_cache)

    return self.cleaned_data

```

Figura 39 Código de Validación del Login

Esta función clean es una función de una clase llamada LoginView proporcionada por Django, esta clase permite la captura de información del formulario permitiendo la validación de la información ingresada, también verifica que no se encuentren vacíos los campos.

4.1.4.2. PANTALLA DE REGISTRO DE USUARIO

La pantalla de registro de usuario despliega un formulario totalmente validado por los siguientes errores:

- El nombre de usuario no se puede repetir, es único el usuario
- La validación del nombre y apellido tiene un máximo de 150 caracteres permitidos, no se permite valores en blanco
- La contraseña se valida de la siguiente manera:
 - o La contraseña no debe ser demasiada corta
 - o La contraseña no tiene que ser palabras o frases comunes
 - o La contraseña no tiene que ser completamente numérica
 - o Se tiene que verificar la contraseña repitiéndola para poder registrar el usuario nuevo
- El nombre y apellido se permiten almacenar en blanco, no es requerido
- El permiso de usuario siempre se debe escoger para establecer que rol tendrá el usuario

Ya existe un usuario con este nombre.

Esta contraseña es demasiado corta. Debe contener al menos 8 caracteres.

Esta contraseña es demasiado común.

Crear Nuevo Usuario

Nombre de Usuario

Nombre

Apellido

Contraseña Repetir Contraseña

Permiso de Usuario

Figura 40 Creación de Usuario

```
AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]
```

Figura 41 Configuración de Validaciones

En las administraciones de contraseña, Django proporciona un conjunto flexible y seguro de herramientas para el manejo y validaciones de contraseñas

Django trae integrado un módulo llamado **hashers**, este módulo proporciona las funciones pertinentes para el uso de validaciones de las contraseñas.

```
def clean_password2(self):
    password1 = self.cleaned_data.get("password1")
    password2 = self.cleaned_data.get("password2")
    if password1 and password2 and password1 != password2:
        raise ValidationError(self.error_messages["password_mismatch"], code="password_mismatch",)
    return password2

def _post_clean(self):
    super()._post_clean()
    password = self.cleaned_data.get("password2")
    if password:
        try:
            password_validation.validate_password(password, self.instance)
        except ValidationError as error:
            self.add_error("password2", error)

def save(self, commit=True):
    user = super().save(commit=False)
    user.set_password(self.cleaned_data["password1"])
    if commit:
        user.save()
    return user
```

Figura 42 Código de las Validaciones de las Contraseñas

En la figura 41, el código muestra la validación de la contraseña que se almacena en el formulario de creación de nuevo usuario, se obtiene los datos del password1 y el password2 que son los ingresados por el usuario, después verifica si las contraseñas existen y si son distintas, al cumplir con esa condición envía un mensaje de error.

La función post_clean almacena el password2 guardándola en una variable, la condicional verifica si existe la variable en el caso de existir llama al método validate_password para validar si cumple con la condición.

La función save, guarda en la variable user la contraseña almacenada correctamente después de pasar por los filtros de validación.

4.1.4.3. Pantalla de Creación de Clientes Y Contratos

La pantalla de creación de clientes despliega 2 formularios, el primer formulario despliega la información del cliente y el segundo formulario el registro de contratos.

Los dos formularios disponen de validaciones independientes, en el caso del registro del cliente la validación se encuentra en el campo cedula, provincia, cantón y parroquia. En los demás campos no son requeridos ya que la empresa nos indicó que esos campos no son obligatorios llenarlos. En el formulario del registro de contrato la validación se encuentra en 3 campos, el número de contrato que no se tiene que repetir, la fecha del contrato la cual no permite tomar fechas posteriores a la actual y el campo plan obligándole al usuario a seleccionar un tipo de plan, los demás campos del formulario no son requeridos.

Figura 43 Pantalla de Registro de Clientes y Contratos

```

contratoForm = inlineformset_factory(Cliente, Contrato, form=ContratoForm, fk_name="cliente", fields=
('direccion', 'localizacion', 'sector', 'contrato', 'fecha_contrato', 'plan'), extra=1, max_num=1,
can_delete=False)

```

Figura 44 Código inline form

En la figura 43, se muestra el código para crear un formulario relacionado mediante una clave foránea, permitiendo generar el formulario de contrato para poder desplegar en la pantalla del template.

```
class ClienteForm(forms.ModelForm):
    nombre = forms.CharField(min_length=1)
    apellido = forms.CharField(min_length=1)
    cedula = forms.CharField(min_length=10, max_length=13,
                             error_messages={'unique': 'Numero de Cedula Repetido'})

    class Meta:
        model = Cliente
        fields = '__all__'
        exclude = ["fecha_edicion", "activo"]

    def __init__(self, *args, **kwargs):
        super(ClienteForm, self).__init__(*args, **kwargs)

        for field in iter(self.fields):
            if field == "plan":
                self.fields[field].widget.attrs.update(
                    {"class": "form-control"})
                self.fields[field].empty_label = "Seleccione el Tipo de Plan"
            else:
                self.fields[field].widget.attrs.update(
                    {"class": "form-control"})
```

Figura 45 Formulario del Cliente

En la figura 44, se muestra la creación del formulario del cliente teniendo en cuenta que el nombre y el apellido se le da la restricción que mínimo contenga 1 carácter, en el campo cedula le damos un mínimo de 10 y un máximo de 13 caracteres permitidos, a su vez en el campo cedula se encuentra validado que solo se permita el ingreso de números.

```
window.onload = function () {
    var fecha = new Date(); //Fecha actual
    var mes = fecha.getMonth() + 1; //obteniendo mes
    var dia = fecha.getDate(); //obteniendo dia
    var ano = fecha.getFullYear(); //obteniendo año
    if (dia < 10) dia = "0" + dia; //agrega cero si el menor de 10
    if (mes < 10) mes = "0" + mes; //agrega cero si el menor de 10
    document.getElementById("id_contrato_set-0-fecha_contrato").max =
        ano + "-" + mes + "-" + dia;
};
```

Figura 46 Validación de Fecha del Contrato

En la figura 45 se valida la fecha, esta validación permite que no pueda elegir fechas posteriores a la actual para el ingreso del contrato.

4.1.4.4. PANTALLA ORDENES DE INSTALACIÓN

En la pantalla de ordenes de instalación se muestra un listado de los clientes, esta opción permite el registro de materiales individual por cada cliente.

# de Contrato	Nombre	Apellido	Cedula	Editar Orden
0301-002	MARIA CARMELINA	LICINTUÑA TUASA	0201317963	Registro de Materiales
0301-003	NESTOR ADAN	AZOGUE AZOGUE	0202357158	Registro de Materiales
0301-004	AZOGUE SIGCHA	LAURA LUZMILA	0201319548	Registro de Materiales
0301-005	JOSE MARIA	CAIZA SIGCHA	0201774379	Registro de Materiales
0301-006	SEGUNDO MIGUEL	RUIZ GUAMAN	0200752335	Registro de Materiales
0301-007	NESTOR ANIBAL	CHUGCHILAN SIGCHA	0201756814	Registro de Materiales
0301-008	RUBEN STALIN	POAQUIZA TIXILEMA	0250353455	Registro de Materiales
0301-009	NESTOR ISAIAS	AYME AYME	0201185980	Registro de Materiales
0301-010	SEGUNDO ANGEL	CAIZA CAIZAHUANO	0201310306	Registro de Materiales
0301-011	HECTOR FAVIAN	AZOGUE COLLAY	0201908761	Registro de Materiales

Figura 47 Listado de Ordenes de instalación

4.1.4.5. PANTALLA DE REGISTRO DE MATERIALES

El registro de materiales se divide en dos partes, el primer formulario es la orden de instalación y la otra los materiales, los dos formularios cuentan con muy pocas validaciones por los requisitos empleados por la empresa, en la figura 47 observamos la pantalla de los materiales. Algo muy importante al registrar los materiales es la fecha de instalación del servicio a partir de esta fecha se genera los recibos de cobros.

QUICKNET 👤

SISTEMA QUICKNET

- 👤 Cliente >
- 📄 Contrato >
- 📍 Ordenes de Instalacion >
- 📥 Ingresos >
- 📄 Pre-Facturacion >
- ⚙️ Configuraciones >

REGISTRO DE MATERIALES

Datos del Cliente

Nombre MARIA CARMELINA	N. de Contrato 0301-002
Cedula 0201317963	Fecha 20 de Octubre de 2021
Plan Servicio de internet Plan Familiar 15 Mbps	Telefono 0968653755
Direccion GONZALES SUAREZ Y ECHEANDIA	

Latitud Longitud

Materiales

Serie	Descripcion	Cantidad (u)
<input type="text"/>	KIT 1 ROSETA	<input type="text" value="0.0"/>
<input type="text" value="(235)AMC-9CE-37408EDF6.20 (24)"/>	ONT	<input type="text" value="1.0"/>
<input type="text"/>	WIRELESS	<input type="text" value="1.0"/>
<input type="text"/>	ONT/WIRES	<input type="text" value="0.0"/>
<input type="text"/>	CONECTOR FC/UPC	<input type="text" value="1.0"/>
<input type="text"/>	PIGTAIL SC-APC (VERDE)	<input type="text" value="1.0"/>
<input type="text"/>	PIGTAIL SC-UPC (AZUL)	<input type="text" value="0.0"/>
<input type="text"/>	SENSOR	<input type="text" value="6.0"/>

Figura 48 Pantalla Registro de Materiales

Como explicaba anteriormente, el registro de la fecha de la orden de instalación es lo fundamental para continuar con el proceso de cobro. A continuación, se muestra el proceso a realizar para la creación de la primera orden de cobro del cliente a partir de la fecha registrada de la instalación.

```

def form_valid(self, form):
    context = self.get_context_data()
    orden = context["orden"]
    mes_gratis = self.request.POST["radio"]
    fecha_atencion = form.instance.fecha_atencion
    contrato_guardado = form.instance.contrato
    print(contrato_guardado)
    estado = form.instance.modifico
    orden.instance = self.object

    if orden.is_valid() and form.is_valid():

        if (fecha_atencion and estado == 0):
            form.instance.modifico = 1 # ! modificar esto al final
            mes_ultimo = fecha_atencion.replace(day=calendar.monthrange(
                fecha_atencion.year, fecha_atencion.month)[1])
            ultimo_dia_del_mes = mes_ultimo.day
            dia_atencion = fecha_atencion.day
            modelo = super(OrdenEdit, self).get_queryset().filter(
                id=self.kwargs["pk"])

            saldoPlan = [i.contrato.plan.precio for i in modelo]
            contrato = [i.contrato.id for i in modelo]

            dias_restantes = (ultimo_dia_del_mes - dia_atencion)
            if(mes_gratis == "ok"):
                valorPagar = 0
                valor_mes_gratis = fecha_atencion + timedelta(30)
                mes_ultimo = valor_mes_gratis
                contra = Contrato.objects.get(pk=contrato[0])
                factura_encabezado = FacturaEncabezado(
                    contrato=contra, fecha_orden=fecha_atencion, sub_total=valorPagar, descuento=0,
                total=valorPagar)
            else:
                if(dia_atencion == 1):
                    valorPagar = saldoPlan[0]
                elif(dias_restantes == 0):
                    valorPagar = 0
                else:
                    valorPagar = (
                        saldoPlan[0]/(ultimo_dia_del_mes))*(dias_restantes)
                    valorPagar = round(valorPagar, 2)
                contra = Contrato.objects.get(pk=contrato[0])
                factura_encabezado = FacturaEncabezado(
                    contrato=contra, fecha_orden=fecha_atencion, sub_total=valorPagar, descuento=0,
                total=valorPagar)

            if factura_encabezado:
                factura_encabezado.save()
                factura = FacturaEncabezado.objects.get(
                    id=factura_encabezado.id)
                factura_detalle = FacturaDetalle(factura=factura,
                    producto=None,
                    cantidad=1,
                    nombre_plan=factura.contrato.plan.nombre_plan,
                    precio=valorPagar,
                    sub_total=valorPagar,
                    descuento=0,
                    total=valorPagar)

                if(mes_gratis == "ok"):
                    pago_factura = Pago_Factura(factura=factura, fecha_factura=factura.fecha,
                    fecha_del_mes=fecha_atencion,
                    fecha_plazo=mes_ultimo, deuda=valorPagar,
                    saldo=valorPagar, generada=True, mes_gratis=True)
                else:
                    pago_factura = Pago_Factura(factura=factura, fecha_factura=factura.fecha,
                    fecha_del_mes=fecha_atencion,
                    fecha_plazo=mes_ultimo, deuda=valorPagar,
                    saldo=valorPagar, generada=True)

            if factura_detalle:
                factura_detalle.save()
                if pago_factura:
                    pago_factura.save()

```

Figura 49 Código para la Creación de la Primera Orden de Cobro

En la figura 48 se observa una función form_valid, esta función permite verificar si el formulario es válido, realizamos una condicional para ver si son válidos los formularios, en el caso de ser así modificamos el estado para ocultar el campo fecha de instalación del formulario.

Una vez realizada esa validación recuperamos los datos del cliente para obtener las fechas de atención, el valor a pagar del plan, cuantos días restantes uso el servicio de internet, calculamos el valor a pagar y guardamos los datos en la base de datos.

4.1.4.6. REGISTRO DE PRODUCTOS

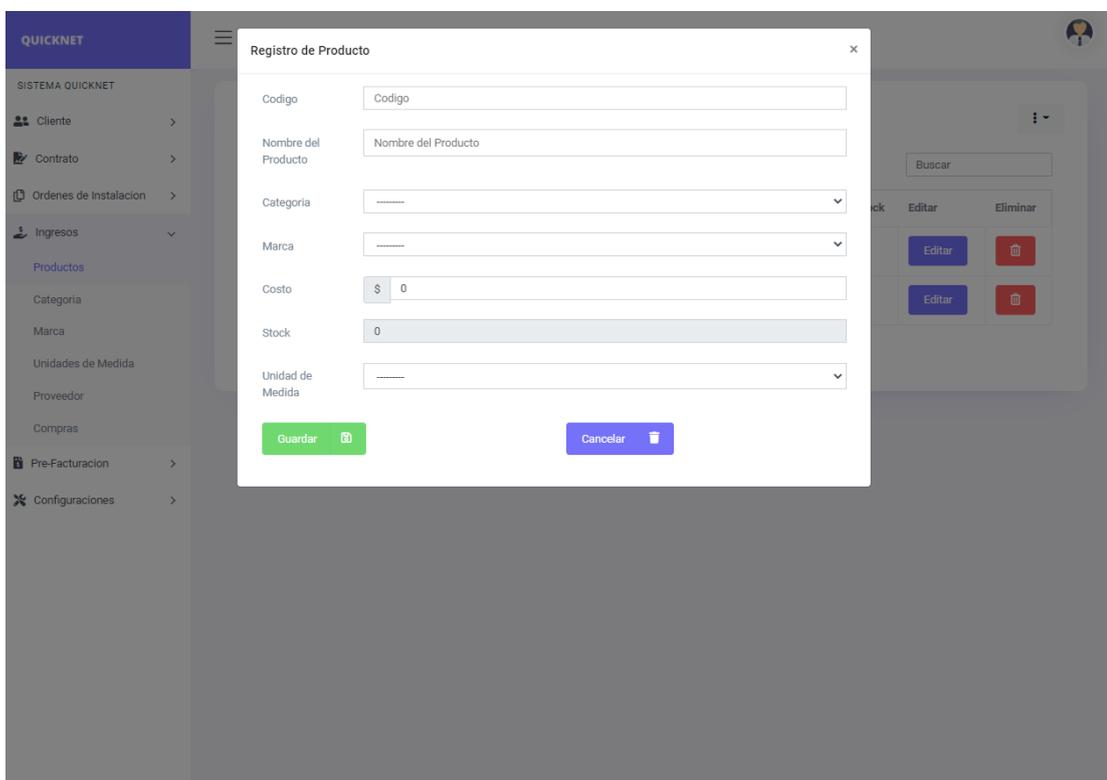


Figura 50 Registro de Productos

En la figura 49, se observa un modal con el formulario para registrar los productos, este formulario registra los productos adquiridos para después venderlos, existen varias validaciones, el código no se puede repetir es único, la categoría, la marca, la unidad de medida y el costo son campos obligatorios.

```
Class ProductoNew(SuccessMessageMixin, generic.CreateView):
    model = Productos
    template_name = 'inventario/producto_form.html'
    context_object_name = 'obj'
    form_class = ProductosForm
    success_url = reverse_lazy("inventario:productos")
    success_message = "Producto Creado Satisfactoriamente"
    permission_required = "inventario.add_producto"

    def form_valid(self, form):
        form.instance.estado = True
        return super().form_valid(form)

    def form_invalid(self, form):
        response = super().form_invalid(form)
        print(form.errors)
        return response
```

Figura 51 Código de Productos

En la figura 50 se muestra el código de la clase nuevo producto, esta clase hereda de `CreateView`, siendo esta específico para registrar nuevos productos. Las clases de Django nos ayudan a optimizar código y a realizar acciones específicas de una manera más rápida, una desventaja de esto es que no es muy personalizable a diferencia de las vistas basada en funciones.

4.1.4.7. PANTALLA REGISTRO DE COMPRAS

Figura 52 Registro de Compras

En la figura 51, se muestra la pantalla de registrar compras, el usuario seleccionará el proveedor, ingresará el número de factura, la fecha de compra, en la parte derecha se seleccionará un producto adquirido, al seleccionar el botón de acciones automáticamente se seleccionará el producto para llenar los valores de los productos tanto el precio, la cantidad y el descuento.

Existen diversas validaciones que describiré a continuación:

- El usuario deberá seleccionar un proveedor
- el número de factura, fecha de compra y fecha factura son campos obligatorios
- siempre se debe seleccionar un producto, caso contrario no se podrá efectuar la compra
- los campos cantidad, precio y descuento despliegan un mensaje si no se registra valores

```
$("#frmCompras").submit(function (e) {
    var no_fact = $("#id_no_factura").val();
    console.log(no_fact);

    if (no_fact.trim() == "") {
        e.preventDefault();
        $.alert({
            type: "red",
            title: "",
            content: "Debe Especificar No. de Factura",
        });
        return false;
    }

    if (
        $("#id_fecha_compra").val() == "" ||
        $("#id_fecha_factura").val() == ""
    ) {
        e.preventDefault();
        $.alert({
            type: "red",
            title: "",
            content: "Debe Especificar Fechas",
        });
        return false;
    }

    if ($("#id_total_detalle").val() == 0 || $("#id_id_producto").val() == 0) {
        e.preventDefault();
        //alert("No ha agregado producto o está sin cantidades");
        $.alert({
            type: "red",
            title: "",
            content: "No ha agregado producto o está sin cantidades",
        });
    }
});
```

Figura 53 Validaciones Para el Registro de Compras

Al registrar una compra el usuario visualizara el listado de compras donde se observa el estado de la compra, las observaciones, el total de la compra y la fecha.

#	Fecha Compra	Observaciones	No. Factura	Sub Total	Descuento	Total	Estado	Editar	Inactivar
1	12 de Marzo de 2021	COMPRA DE ONT-ONU PARA EL CARMELO	624	1475,0	0,0	1475,0	Activo		
2	2 de Marzo de 2022		0001	1000,0	0,0	1000,0	Activo		
3	22 de Junio de 2022	PRUEBA	34434	20,0	0,0	20,0	Activo		

Mostrando desde 1 hasta 3 - En total 3 resultados

Figura 54 Pantalla Listado de Compras

4.1.4.8. PANTALLA IMPRESIÓN DE COBROS

Contrato	Cliente	Total	Mes de Cobro	Fecha de Inicio	Fecha de Final	Estado	Lugar	Acciones
0002	Gustavo Lombeida	12,00 \$	Junio	2022-06-14	2022-06-30	Adjudado	SIMIATUG	[Iconos]
0301-001	RAUL MARIA AZOGUE	30,83 \$	Diciembre	2021-12-01	2021-12-31	Adjudado	SIMIATUG	[Iconos]
0301-001	RAUL MARIA AZOGUE	39,16 \$	Enero	2022-01-01	2022-01-31	Adjudado	SIMIATUG	[Iconos]
0301-001	RAUL MARIA AZOGUE	47,50 \$	Febrero	2022-02-01	2022-02-28	Adjudado	SIMIATUG	[Iconos]
0301-001	RAUL MARIA AZOGUE	22,50 \$	Marzo	2022-03-01	2022-03-31	Adjudado	SIMIATUG	[Iconos]
0301-001	RAUL MARIA AZOGUE	47,50 \$	Abril	2022-04-01	2022-04-30	Adjudado	SIMIATUG	[Iconos]
0301-001	RAUL MARIA AZOGUE	22,50 \$	Mayo	2022-05-01	2022-05-31	Adjudado	SIMIATUG	[Iconos]

Figura 55 Listado de Facturas de los Clientes

En la figura 54, se observa el listado de las personas que tienen valores pendientes, en la pantalla se observa 2 botones, el primer botón permite añadir productos que adquirió el cliente y el segundo botón el usuario imprimirá la orden de cobro de la persona seleccionada.

ORDEN DE COBRO

Cedula: 1454525452 No de Contrato: 0002
 Nombre y Apellido : Gustavo Lombeida Fecha de Emision: 2022-07-01
 Mes de Cobro : Junio 2022 Descuento: Normal

ITEM	DESCRIPCION	CANTIDAD	P. UNITARIO	P. TOTAL
1	Servicio de internet: Plan Familiar 15 Mbps	1	\$ 12,00	\$ 12,00
Valor a Pagar :				\$ 12,00

Efectivo ___ Transferencia ___ Deposito ___

RECIBI CONFORME **ENTREGUE**

FIRMA _____ FIRMA _____
 NOMBRE _____ NOMBRE _____

RECIBO DE COBRO

Cedula: 1454525452 No de Contrato: 0002
 Nombre y Apellido : Gustavo Lombeida Fecha de Emision: 2022-07-01
 Mes de Cobro : Junio 2022 Descuento: Normal

ITEM	DESCRIPCION	CANTIDAD	P. UNITARIO	P. TOTAL
1	Servicio de internet: Plan Familiar 15 Mbps	1	\$ 12,00	\$ 12,00
Total a Pagar :				\$ 12,00

Efectivo ___ Transferencia ___ Deposito ___

RECIBI CONFORME **ENTREGUE**

Figura 56 Pdf de la Orden de Cobro

En la figura 55, se observa el pdf de la orden de cobro, este pdf fue creado con una herramienta llamada xhtml2pdf, esta herramienta genera pdf mediante un html dándole así los atributos para poder crear el pdf, a continuación, se muestra el código para crear el pdf en Django.

```
def recibo_de_cobro(request, pk):
    encabezado = Pago_Factura.objects.get(factura=pk)
    detalle = FacturaDetalle.objects.filter(factura=pk)

    for i in detalle:
        print(i.descripcion)
    if(encabezado.factura.fecha_orden):
        one_day = timedelta(days=1)
        fecha = encabezado.factura.fecha_orden #fecha de la orden de instalacion
        mes2 = fecha.strftime('%F')
        mes = int(fecha.strftime('%m')) # mes de la orden de instalacion
        anio = int(fecha.strftime('%Y')) # año de la orden de instalacion
        ultimo_dia = monthrange(anio, mes)[1] #ultimo dia del mes
        fecha_ultima = "{}-{}-{}".format(anio,mes,ultimo_dia) #damos formato al ultimo dia
        fecha_ultima = datetime.strptime(fecha_ultima,'%Y-%m-%d').date() #transformamos a fecha el ultimo
        fecha del mes

        fecha_siguiete = fecha_ultima + one_day #primer dia del proximo mes
        fecha_siguiete.strftime('%Y-%m-%d')
        print(fecha_siguiete)

        context = {
            'orden': 'ORDEN DE COBRO',
            'pago': 'RECIBO DE COBRO',
            'encabezado':encabezado,
            'fecha_siguiete': fecha_siguiete,
            'detalle':detalle,
        }
        template_path = 'pdf/recibo2.html'

        response = HttpResponse(content_type='application/pdf')
        response['Content-Disposition'] = f'attachment; filename="
{encabezado.factura.contrato.contrato}_{encabezado.factura.contrato.cliente.nombre}
{encabezado.factura.contrato.cliente.apellido}_{mes2}_{encabezado.factura.contrato.cliente.parroquia}.pdf
"'

        # find the template and render it.
        template = get_template(template_path)
        html = template.render(context)
        pisa_status = pisa.CreatePDF(html, dest=response)
        # create a pdf
        if pisa_status.err:
            return HttpResponse('We had some errors <pre>' + html + '</pre>')
        return response
    else:
        return redirect("facturacion:factura_edit", id=pk)
```

Figura 57 Código para la creación del Recibo de Cobro

Se crea una función llamada `recibo_de_cobro` esta función tiene 2 parámetros, la `request` que se usa para hacer peticiones y el `pk` que es la identificación del cliente, realizamos una consulta al encabezado y al detalle de la factura para obtener los datos respectivos del cliente, recorremos el detalle de la factura y almacenamos en variables los valores obtenidos, en la

variable context enviamos las variables que se mostrara en la orden, establecemos el nombre de la orden de cobro, el template y creamos el pdf.

4.1.4.9. PANTALLA DE REGISTRO DE PAGO

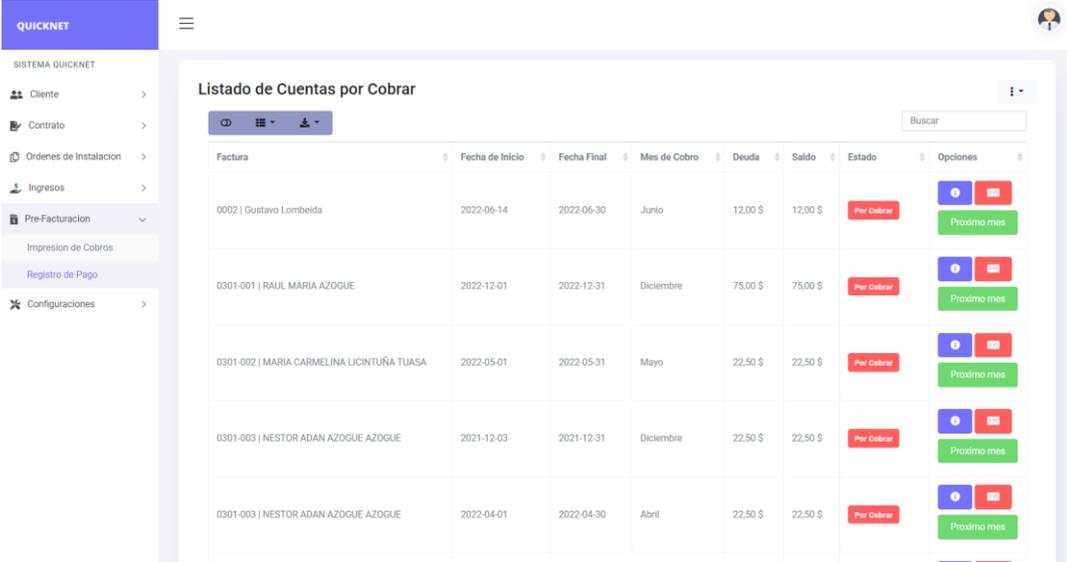


Figura 58 Listado de Registro de Pago

Esta pantalla muestra el listado de los clientes con sus valores faltante por pagar, existen 3 botones el primer botón muestra la información de sus pagos, la siguiente muestra la pantalla de registrar pago y el botón “próximo mes” genera el siguiente mes sin haber pagado el mes anterior, este proceso registra el valor acumulado de los meses en la impresión de la orden de cobro.

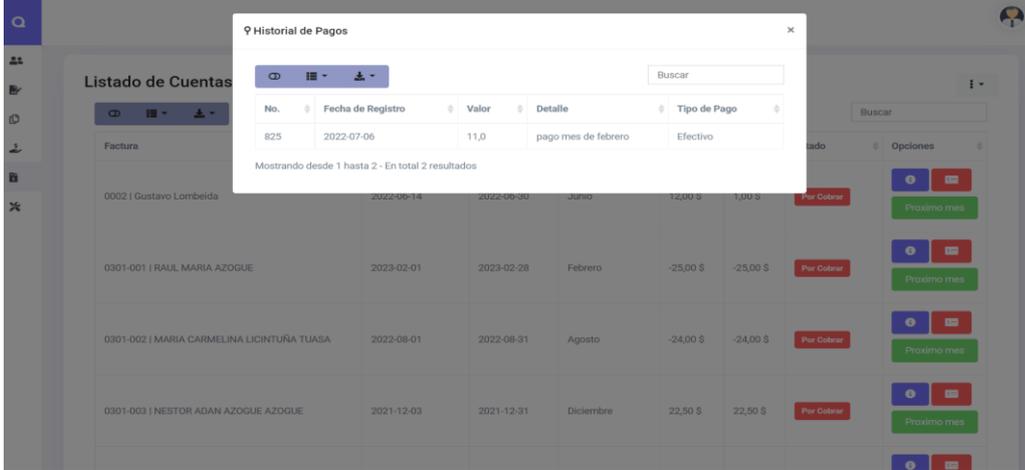


Figura 59 Historial de Pagos de los Clientes

En la figura 58, se observa un historial de pagos realizados, nos muestra la fecha del pago, el valor cancelado, el detalle y el tipo de pago.

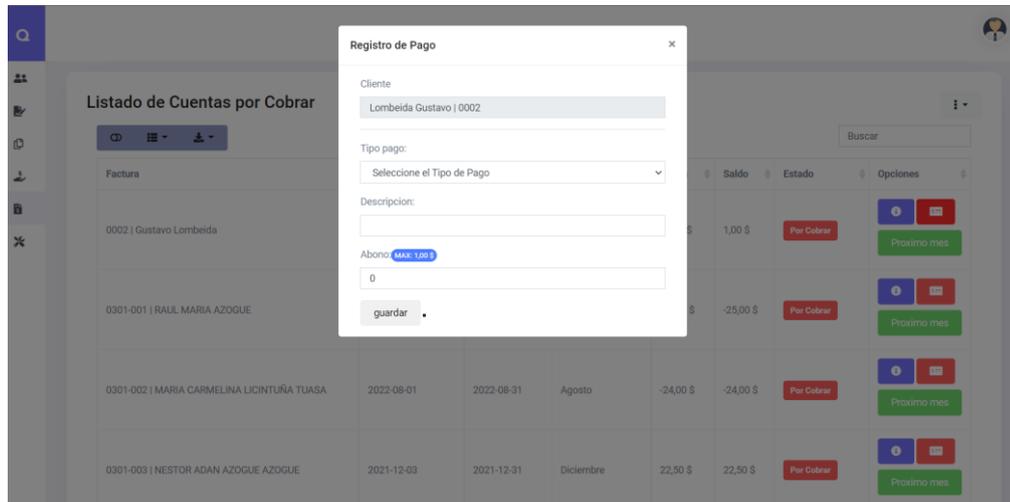


Figura 60 Formulario del Registro de Pago

En la figura 59, observamos que existe un formulario dentro de un modal, este modal dispone de varios campos, el cliente, el tipo de pago que se puede seleccionar los siguientes: deposito, transferencia o efectivo, este campo es obligatorio llenarlo, se encuentra validado para que se seleccione. El campo abono ingresaremos el valor que el cliente abona, este campo se encuentra validado de la siguiente forma:

- validado para que ingrese solo números
- validado para que el valor de abono no sea menor a 0 o igual
- validado para que el valor ingresado no sea mayor que el saldo



Figura 61 Código de validación del Formulario de Pago

En la figura 60, se observa la validación del campo abono, validamos 3 cosas, que el abono no sea menor o igual que 0 y que el abono siempre tiene que ser mayor al valor de deuda.

Observamos que en un template podemos mezclar JavaScript con sintaxis de Django, eso nos facilita únicamente si usamos el script dentro del html.

4.1.4.10. GitHub

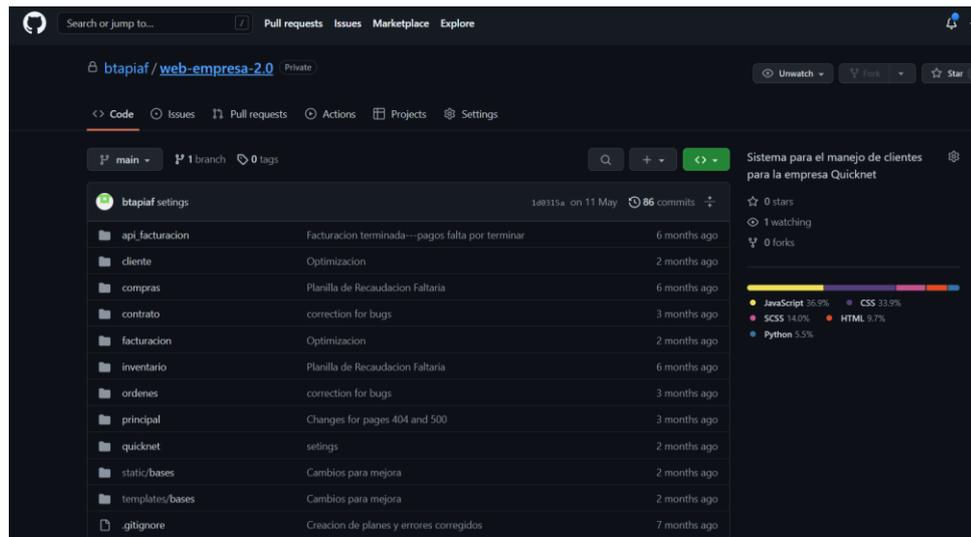


Figura 62 Control de Versiones del Proyecto

Para el control de versiones del proyecto usamos GitHub una herramienta muy útil para almacenar y controlar el desarrollo de nuestra aplicación. El repositorio se encuentra privado por motivos de confidencialidad de la empresa.

4.2. PRUEBAS

A continuación, se realizó dos tipos de pruebas: pruebas de carga y pruebas dentro del código. Las pruebas se ejecutaron con el propósito de evidenciar el trabajo del sistema informático web de la empresa Quicknet.

4.2.1. PRUEBAS DEL CÓDIGO

Para las pruebas de código utilizamos la plataforma de código abierto SonarQube. SonarQube es una herramienta que nos ayuda al análisis de código fuente. Nos proporciona métricas que nos permite detectar vulnerabilidades en nuestro software.

La plataforma tardó un tiempo de 53.4345 segundos para analizar nuestro sistema, el tiempo del análisis varía dependiendo el número de archivos que se encuentra en la aplicación.

SonarQube nos muestra un resumen amplio de las pruebas realizadas.

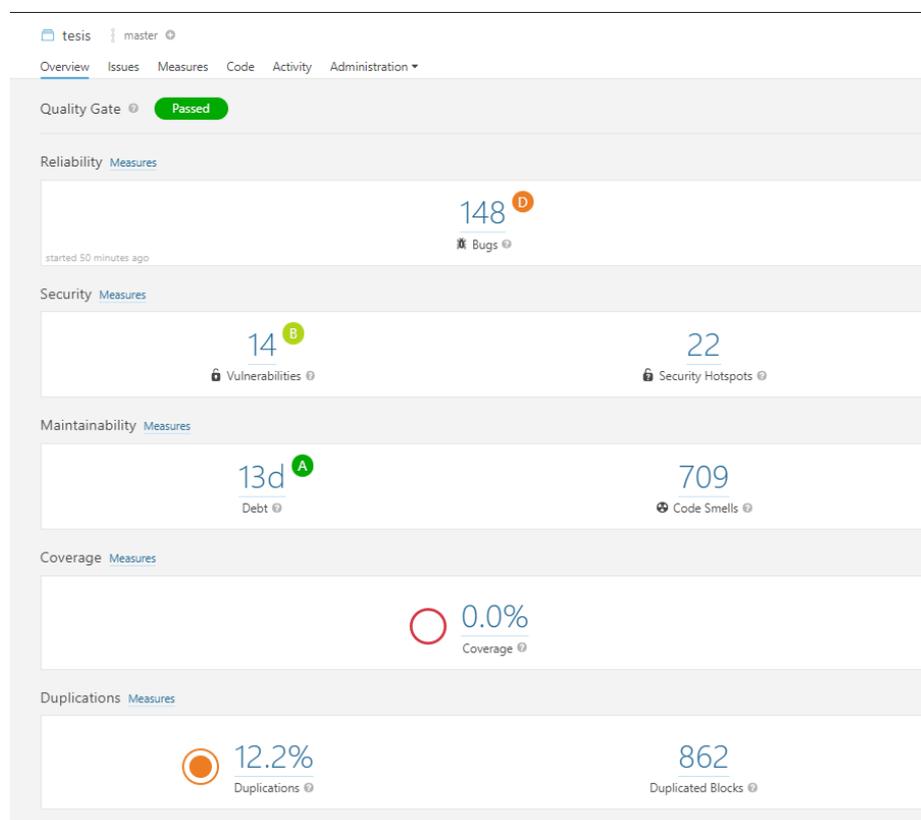


Figura 63 Resumen de las Pruebas Realizadas

En la figura 62, observamos como resultado que existen 148 bugs, 14 vulnerabilidades, 22 errores de seguridad, 13 días se tomaría en dar mantenimiento a la aplicación, 709 fragmentos de código difícil de mantener, 12.2% de código duplicado y 862 bloques duplicados.

4.2.2. MEDIDAS DE FIABILIDAD

En las pruebas realizadas observamos 148 problemas de bugs en la aplicación. Al revisar, la mayoría de los errores se enfocan en archivos HTML generado por una etiqueta.

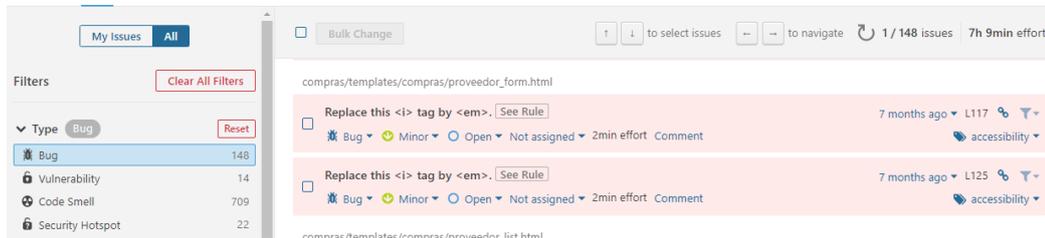


Figura 64 Detalle de las Medidas de Fiabilidad

4.2.3. MEDIDAS DE SEGURIDAD

En las pruebas realizadas observamos 14 vulnerabilidades, al analizar el código podemos observar que las vulnerabilidades son del código de los plugin utilizados. Para la corrección eliminamos los errores que nos muestran en el detalle de estas.

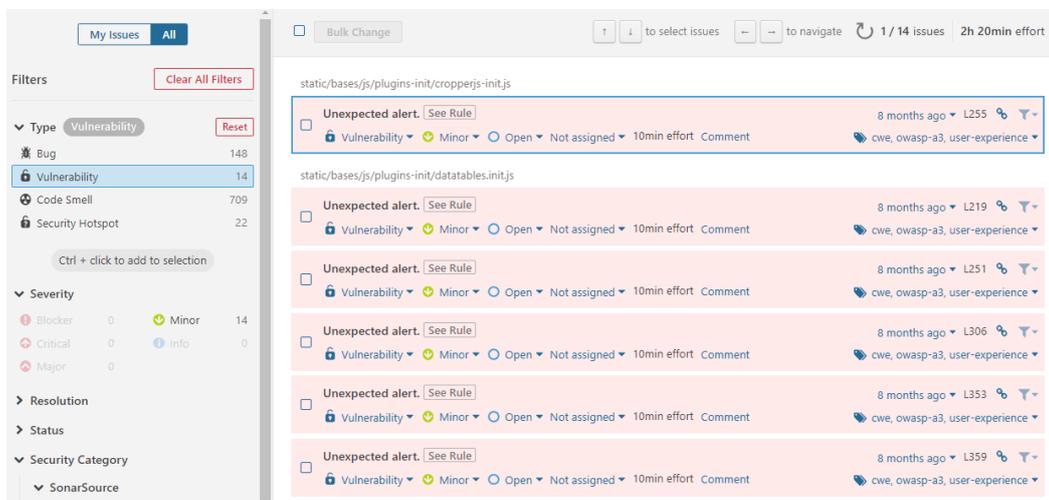


Figura 65 Detalle de las Vulnerabilidades

4.2.4. DUPLICIDAD DEL CÓDIGO

Se identifico el 12.2% de código duplicado en todo el código fuente del sistema, los archivos de Bootstrap son los que presentan mayor código duplicado A continuación, se muestra el resultado de las pruebas de duplicidad:

Duplicated Lines (%) 12.2% 

	Duplicated Lines (%)	Duplicated Lines
 static/bases/js/plugins-init/bootstrap-notify-init.js	98.1%	622
 static/bases/plugins/multiselect/jquery.multi-select.js	97.8%	524
 static/bases/plugins/multiselect/js/jquery.multi-select.js	97.8%	524
 static/bases/js/plugins-init/chartjs-widget-init.js	95.2%	1,485
 static/bases/plugins/bootstrap-fileinput/js/locales/de.js	90.9%	90
 static/bases/plugins/bootstrap-fileinput/js/locales/LANG.js	90.0%	90
 static/bases/plugins/bootstrap-fileinput/js/locales/pt.js	90.0%	90
 static/bases/plugins/bootstrap-fileinput/js/locales/th.js	90.0%	90
 static/bases/plugins/bootstrap-fileinput/js/locales/et.js	89.9%	89
 static/bases/plugins/bootstrap-fileinput/js/locales/sv.js	89.9%	89
 static/bases/plugins/bootstrap-fileinput/js/locales/bg.js	89.1%	90
 static/bases/plugins/bootstrap-fileinput/js/locales/ca.js	89.1%	90
 static/bases/plugins/bootstrap-fileinput/js/locales/cs.js	89.1%	90

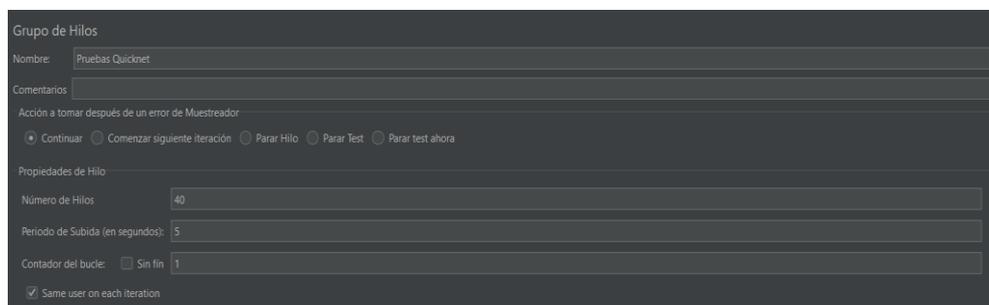
Figura 66 Duplicidad del código

4.2.5. PRUEBAS DE RENDIMIENTO

La aplicación QUICKNET se encuentra diseñada para usuarios de uso exclusivo de la empresa con conocimiento del proceso de manejo técnico y administrativo del proveedor de servicio de internet.

Para las pruebas de rendimiento se utilizó una herramienta de software libre llamada JMeter, esta herramienta realiza las pruebas de rendimiento y de sobrecarga con las peticiones respectivas.

Se realizó las pruebas de rendimiento con la siguiente configuración.



Grupo de Hilos

Nombre: Pruebas Quicknet

Comentarios:

Acción a tomar después de un error de Muestreador

Continuar Comenzar siguiente iteración Parar Hilo Parar Test Parar test ahora

Propiedades de Hilos

Número de Hilos: 40

Período de Subida (en segundos): 5

Contador del bucle: Sin fin 1

Same user on each iteration

Figura 67 Configuración Para las Pruebas de Rendimiento

Para las pruebas de rendimientos se determinó un escenario de 40 usuarios, cada usuario se le permitió que acceda cada 5 segundo al sitio web. Se evalúa desde que se registra el cliente hasta que se imprime la orden de cobros.

Etiqueta ↓	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Min	Máx	% Error	Rendimiento	Kb/sec	Sent KB/sec
Usuarios	40	766	841	918	948	973	112	973	0,00%	58,9/min	43,88	0,12
Pre-Facturas	40	21551	23938	24551	24700	24990	371	24990	0,00%	1,4/sec	566,12	0,19
Ordenes de Instalacion	40	189	68	572	594	634	28	634	0,00%	6,8/sec	31,42	3,31
Cuentas por Cobrar	40	13398	13628	14452	14676	14950	636	14950	0,00%	58,6/min	282,93	0,14
Contratos	40	263	87	663	692	870	37	870	0,00%	6,9/sec	33,78	3,60
Compras	40	253	43	699	792	911	14	911	0,00%	6,1/sec	69,00	0,79
Clientes	40	196	88	486	529	579	41	579	0,00%	7,8/sec	35,56	3,05
Total	280	5231	572	21714	24191	24640	14	24990	0,00%	6,7/sec	730,95	1,85

Figura 68 Resumen, Reporte de Carga

Como observamos en la figura 67, se muestra el resultado de la prueba de carga generada con 5231 milisegundos y un porcentaje de error del 0% determinando que la aplicación no generara. Se determinó que el rendimiento del sistema es óptimo para el uso en la empresa.

4.2.6. ANÁLISIS DE RESULTADOS.

De acuerdo con las pruebas proporcionadas, el sistema tiene un desempeño satisfactorio en las siguientes en funcionalidad y en codificación. En el sistema existe poca probabilidad de que accedan más de dos usuarios.

5. LISTA DE REFERENCIAS

Alexandra, E., Moncada, G., Daniel, I., & Mata López, A. (2019). TESIS PRESENTADA POR: TUTOR DE TESIS. *Tesis de Grado*.

Alvarado, M., & Nicols, B. (2020). Sistema Web para el Proceso de Cobranza Académica en el Colegio Saco Oliveros [Universidad César Vallejo]. In *Repositorio Institucional - UCV*. <https://repositorio.ucv.edu.pe/handle/20.500.12692/51952>

CHAVEZ, R., & JACK, F. (2018). *UNIVERSIDAD PERUANA DE LAS AMÉRICAS ESCUELA*

PROFESIONAL DE INGENIERÍA DE COMPUTACIÓN Y SISTEMAS Implementación de un Sistema Web para Optimizar el Proceso de Gestión de Cobranza en la Empresa Service Collection. 1–119.

[http://repositorio.ulasamericas.edu.pe/bitstream/handle/upa/258/IMPLEMENTACIÓN DE UN SISTEMA WEB PARA OPTIMIZAR EL PROCESO DE GESTIÓN DE COBRANZA EN LA EMPRESA SERVICE.pdf?sequence=1&isAllowed=y](http://repositorio.ulasamericas.edu.pe/bitstream/handle/upa/258/IMPLEMENTACIÓN%20DE%20UN%20SISTEMA%20WEB%20PARA%20OPTIMIZAR%20EL%20PROCESO%20DE%20GESTIÓN%20DE%20COBRANZA%20EN%20LA%20EMPRESA%20SERVICE.pdf?sequence=1&isAllowed=y)

Django. (2005). *The Web framework for perfectionists with deadlines*. Django Project. <https://www.djangoproject.com/>

Eduardo Pérez Bautista, A. (2012). UNIVERSIDAD TÉCNICA DE AMBATO. *Tesis de Grado*.

Hernández Calderón, L. Y. (2019). UNIVERSIDAD TÉCNICA DEL NORTE FACULTAD DE CIENCIAS ADMINISTRATIVAS Y ECONÓMICAS CARRERA DE INGENIERÍA EN CONTABILIDAD Y AUDITORÍA CPA. <http://repositorio.utn.edu.ec/bitstream/123456789/10468/2/F02%2520ICA%25201649%2520TRABAJO%2520GRADO.pdf>

Mateo Taboada, B. J. (2018). *Sistema web para el proceso de cobranzas en el área de operaciones de la empresa Lerma S.A.C.*

Millan, J. M. (2019). *Bolivia | Por mal servicio y cobro excesivo de empresas, Gobierno regulará precios de internet* | DPLNews. Periodio En Línea. <https://digitalpolicylaw.com/bolivia-por-mal-servicio-y-cbro-excesivo-de-empresas-gobierno-regulara-precios-de-internet/>

Morales, A., & Morales, J. (2014). Crédito y Cobranza. In *Crédito y Cobranza* (pp. 1–18). www.editorialpatria.com.mx

MySQL. (2021). <https://www.mysql.com/>

Ramos, R. (2021, September 25). *¿Qué es JavaScript y para qué sirve?*
<https://soyrafaramos.com/que-es-javascript-para-que-sirve/>

Rodriguez Arias, J. J., & Yopez Jordan, L. J. (2021). *SISTEMA WEB PARA LA AUTOMATIZACIÓN DE LOS PROCESOS DE NEGOCIOS Y GESTIÓN DE LA EMPRESA BELLNOVAINSER S.A. DEL CANTÓN BABAHOYO PROPUESTA TECNOLÓGICA.*

Solis, K., & Lizardo, F. (2019). *UNIVERSIDAD PERUANA DE CIENCIAS E INFORMÁTICA FACULTAD DE CIENCIAS E INGENIERÍA INGENIERÍA DE SISTEMAS E INFORMÁTICA “SISTEMA WEB PARA EL PROCESO DE COBRANZA EN LA EMPRESA DE CRÉDITOS SEBASTIÁN” PRESENTADO POR: PARA OPTAR EL TÍTULO PROFESIONAL DE INGENIER.*

techtarget. (n.d.). *¿Qué es Framework? - Definición en WhatIs.com.* Definition Framework.
Retrieved September 28, 2021, from
<https://searchdatacenter.techtarget.com/es/definicion/Framework>

Vega, F., & Van Der Henst S, C. (2011). *Guía HTML5: El presente de la web* (Vol. 1).
http://www.lamolina.edu.pe/osi/manual/Curso_HTML5_v1.pdf