UNIVERSIDAD ESTATAL DE BOLÍVAR

FACULTAD DE CIENCIAS ADMINISTRATIVAS, GESTIÓN EMPRESARIAL E INFORMÁTICA

CARRERA DE SOFTWARE

TRABAJO DE INTEGRACIÓN CURRICULAR PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIEROS EN SOFTWARE

FORMA: PROYECTO TECNOLÓGICO

TEMA:

SISTEMA ON LINE PARA LA GESTIÓN DE VENTAS DEL LOCAL COMERCIAL ISABELITA EN LA CIUDAD DE GUARANDA

AUTORES:

JUAN CARLOS ALDAS ALDAZ,

ANGELO DAVID GARCÍA ARTEAGA

DIRECTORA:

ING. MÓNICA ELIZABETH BONILLA MANOBANDA

GUARANDA – ECUADOR

2024

TEMA DEL PROYECTO TECNOLÓGICO

SISTEMA ON LINE PARA LA GESTIÓN DE VENTAS DEL LOCAL COMERCIAL ISABELITA EN LA CIUDAD DE GUARANDA

DEDICATORIA

El presente proyecto está dedicado a mis seres queridos, mi familia, quienes han sido fundamentales para poder culminar mis estudios universitarios.

A mis padres, José y Juana, que con su amor, apoyo y sacrificios han sido el pilar fundamental en mi formación y crecimiento, quienes siempre me apoyaron incondicionalmente y en momentos difíciles, mi motivación para seguir adelante

Juan C. Aldas

Dedico este proyecto de titulación a mis padres, quienes han sido mi mayor apoyo, les agradezco profundamente. Su amor incondicional, sacrificios y aliento constante me han motivado a seguir adelante incluso en los momentos más difíciles que me han enseñado el valor del esfuerzo y la perseverancia, y por ello, este logro es también un reflejo de su fe en mí, este viaje no ha sido fácil; he enfrentado desafíos que han puesto a prueba mi determinación y resiliencia, cada obstáculo superado ha sido una lección valiosa y un paso más hacia el éxito que celebro hoy.

Hoy miro hacia el futuro con confianza, sabiendo que nada es imposible para quien se esfuerza y persiste, este logro es un testimonio de que el trabajo duro y la dedicación valen la pena.

Angelo D. García.

AGRADECIMIENTO

En primer lugar, agradecemos a Dios por habernos brindado la oportunidad de culminar este proyecto.

Queremos expresar nuestros agradecimientos a la Universidad Estatal de Bolívar por la oportunidad de crecer y desarrollarme en un ambiente académico de excelencia.

A nuestra directora de trabajo de titulación, Ing. Mónica Bonilla Manobanda, por su guía y apoyo durante todo el proceso del proyecto. A nuestros pares académicos, Ing. Rodrigo Del Pozo y Lic. Edgar Rivadeneira, por su colaboración y recomendaciones a nuestro proyecto que han enriquecido nuestro trabajo.

A nuestro coordinador de trabajo de integración curricular, Ing. Darwin Carrión, por su apoyo y orientación en el proceso de titulación, a todos los profesores de la carrera que nos impartieron su conocimiento y compartieron sus experiencias en el ámbito profesional, lo que nos ha servido para el desarrollo de nuestro tema.

Juan C. Aldaz y Angelo D. García.

CERTIFICADO DE VALIDACIÓN

UNIVERSIDAD ESTATAL DE BOLIVAR

FACULTAD DE CIENCIAS ADMINISTRATIVAS, GESTIÓN EMPRESARIAL E INFORMÁTICA

CERTIFICADO DE VALIDACIÓN

Ing. Mónica Elizabeth Bonilla Manobanda, Ing. Edgar Rivadeneira Ramos e Ing. Rodrigo Humberto Del Pozo Durango, en su orden directora y Pares Académicos del Trabajo de Integración Curricular "Sistema on line para la gestión de ventas del local comercial ISABELITA en la ciudad de Guaranda" desarrollado por los estudiantes Juan Carlos Aldas Aldaz y Angelo David García Arteaga.

CERTIFICAN

Que, luego de revisado el Trabajo de Integración Curricular en su totalidad, cumple con las exigencias académicas de la carrera SOFTWARE, por lo tanto, autorizamos su presentación y defensa.

Guaranda, 06 de diciembre del 2024



Ing. Mónica Monilla

Directora



Ing. Edgar Rivadeneira

Par Académico



Ing. Rodrigo Del Pozo

Par Académico

Dirección: Av. Ernesto Che Guevara y Gabriel Secaira
Guaranda-Ecuador
Teléfono: (593) 3220 6059
www.ueb.edu.ec

DERECHOS DE AUTOR

UEB UNIVERSIDAD ESTATAL DE BOLIVAR

BIBLIOTECA GENERAL

DERECHOS DE AUTOR

Yo **JUAN CARLOS ALDAS ALDAZ** con Cédula de Identidad No 0250046372 y **ANGELO DAVID GARCÍA ARTEAGA** con Cédula de Identidad No 0202486650 en calidad de autores y titulares de los derechos morales y patrimoniales del Trabajo de Titulación: **SISTEAM ON LINE PARA LA GESTIÓN DE VENTAS EN EL LOCAL COMERCIAL ISABELITA DE LA CIUDAD DE GUARANDA** modalidad PROYECTO TECNOLÓGICO, de conformidad con el Art. 114 del CÓDIGO ORGÁNICO DE LA ECONOMÍA SOCIAL DE LOS CONOCIMIENTOS, CREATIVIDAD E INNOVACIÓN, concedemos a favor de la Universidad Estatal de Bolívar, una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos.

Conservamos a nuestro favor todos los derechos de autor sobre la obra, establecidos en la normativa citada.

Así mismo, autorizamos a la Universidad Estatal de Bolívar, para que realice la digitalización y publicación de este trabajo de titulación en el Repositorio Digital, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Los autores declaran que la obra objeto de la presente autorización es original en su forma de expresión y no infringe el derecho de autor de terceros, asumiendo la responsabilidad por cualquier reclamación que pudiera presentarse por esta causa y liberando a la Universidad de toda responsabilidad.

Juan Carlos Aldas Aldaz

Angelo David García Arteaga

INDICE

TEMA DEL PROYECTO TECNOLÓGICO i
DEDICATORIA ii
AGRADECIMIENTO iii
CERTIFICADO DE VALIDACIÓN iv
DERECHOS DE AUTOR v
INTRODUCCIÓN 1
RESUMEN 2
ABSTRACT 3
CAPÍTULO I 4
FORMULACIÓN GENERAL DEL PROYECTO 4
1.1. Tema 4
1.2. Descripción del Problema 4
1.3. Justificación 5
1.4. Objetivos 6
1.4.1. General 6
1.4.2. Específicos 6
CAPÍTULO II 7
MARCO TEÓRICO 7
2.1. Antecedentes 7
2.2. Científico 9
Ingeniería de software. 9
Ciclo de vida de software 9
Modelos de desarrollo de software. 10
Sistemas de gestión de ventas 10
Control de inventario 10
Métodos de control de inventarios 10
Análisis ABC 10
Stock de seguridad 11
2.3. Conceptual 11
Automatización 11
Arquitectura de software 11
Backend 11

Base de datos 12
Base de datos relacionales. 12
Desarrollo web 12
Frontend 12
Framework. 12
Facturación electrónica 13
Gestor de base de datos 13
JavaScript 13
Lenguajes de programación 13
Modelo Vista Controlador 13
Patrones de arquitectura 13
PostgreSQL 14
Proceso de ventas 14
Régimen Simplificado para Emprendedores y Negocios Populares (RIMPE) 14
React JS 14
Seguridad del software 14
Sistemas de información 14
Ventas 15
API REST 15
Protocolo HTTP 15
2.4. Legal 15
2.5. Georreferencial 16
CAPITULO III 17
METODOLOGÍA 17
3.1. Metodología de Desarrollo de Software 17
3.1.1. Metodologías Agiles 17
3.1.2. Metodología XP 17
3.1.3. Metodología Scrum 18
3.1.4. Metodología Kanban 18
3.1.5. Metodología Desarrollo de sistemas dinámicos (DSDM) 19
3.1.6. Metodología Desarrollo orientado a funcionalidades (FDD) 19
3.1.7. Metodología de desarrollo Scrum 21

3.1.7.1. Teoría de Scrum 21
3.1.7.2. Caracterización de Scrum 22
3.1.7.3. Modelo Scrum para Administración de proyectos de software 22
3.1.7.4. Equipo Scrum 22
3.1.7.5. Eventos de Scrum 23
3.1.7.6. Artefactos de Scrum 24
3.2. Técnicas e Instrumentos de Recopilación de Datos 25
3.2.1. Entrevista 25
3.2.2. Observación directa 25
CAPITULO IV 26
INGENIERÍA DEL PROYECTO 26
4.1. Inicio 26
4.1.1. Introducción 26
4.1.2. Alcance del Producto 26
4.1.3. Limitaciones 27
4.1.4. Definiciones y acrónimos 27
4.1.5. Roles 28
4.1.6. Características de los usuarios 29
4.1.7. Requisitos de usuario 31
3.1.7.7. Historias de usuario 31
3.1.7.8. Requisitos no funcionales 36
3.1.7.9. Requisitos de interfaz 37
4.1.7.1.1. Interfaz de Hardware 37
4.1.7.1.2. Interfaz de Software 37
4.1.7.1.3. Interfaz de Comunicación 37
4.2. Planificación y estimación 38
4.2.1. Product Backlog 38
4.2.2. Estimación del Product Backlog 39
4.3. Implementación 39
4.3.1. Desarrollo del sprint 1. Diseño de alto nivel 39
3.1.7.10. Planificación del sprint 1 39
3.1.7.11. Tareas del sprint 1 40
3.1.7.12. Ejecución del sprint 1 – Scrum diario 41

3.1.7.13. Resultados 43
4.3.1.1.1. Arquitectura del software 43
4.3.1.1.2. Diseño de la navegación del sistema 46
4.3.1.1.3. Modelo entidad relación de la base de datos 47
4.3.2. Desarrollo del sprint 2. Diseño detallado 48
3.1.7.14. Planificación del sprint 2 48
3.1.7.15. Tareas del sprint 2 48
3.1.7.16. Ejecución del sprint 2 – Scrum diario 49
3.1.7.17. Resultados 51
4.3.2.1.1. Diagrama de casos de uso 51
Diagrama de casos de uso de las funciones de Administrador 51
Diagrama de casos de uso de las funciones del Propietario. 52
Diagrama de casos de uso de las funciones del Contador 55
Diagrama de casos de uso de las funciones del Vendedor 56
4.3.2.1.2. Diagrama de secuencia 58
Diagrama de secuencia de inicio de sesión 58
Diagramas de secuencia usuario Administrador 59
Diagramas de secuencia del usuario Propietario 60
Diagramas de secuencia del usuario Contador 66
Diagramas de secuencia del usuario Vendedor 66
4.3.2.1.3. Diseño de la interfaz de usuario 68
Diseño de interfaz de la página de inicio de sesión 68
Interfaces de usuario del Propietario. 69
Interfaces de usuario del Vendedor. 72
Interfaces de usuario del Contador. 74
4.3.3. Desarrollo del sprint 3. Programación Backend 75
3.1.7.18. Planificación del sprint 3 75
3.1.7.19. Tareas del sprint 3 76
3.1.7.20. Ejecución del sprint 3 – Scrum diario 78
3.1.7.21. Resultados 83
4.3.3.1.1. Desarrollo del Backend 84
Creación de entorno backend e instalación de paquetes 84
Conexión con la base de datos de PostgresQL 85
Algoritmo de conexión a la base de datos 85

Creación de la API REST y el servidor 86
Algoritmo de crear el servidor y la API REST 86
Creación del esquema de datos para la base de datos 87
Algoritmo del esquema para la base de datos 87
Construcción de los modelos 107
Algoritmo del modelo para la gestión del inventario 108
Controladores de los modelos 113
Algoritmo del controlador del Kardex 113
Creación de rutas para la API REST 116
Algoritmo de implementacion de rutas 117
4.3.4. Desarrollo del sprint 4. Programación Frontend 118
3.1.7.22. Planificación del sprint 4 118
3.1.7.23. Tareas del sprint 4 119
3.1.7.24. Ejecución del sprint 4 121
4.3.4.1.1. Resultados 129
4.3.4.1.2. Desarrollo del frontend 129
Desarrollo de la página principal del usuario “vendedor” – página de registro de ventas 130
Algoritmo de llamada al componente de la vista principal del usuario “vendedor” 131
Desarrollo de la página principal del usuario “propietario” 131
Algoritmo de maquetación de la página principal del usuario “propietario” 131
Desarrollo de la página de gestión de ventas del usuario “propietario” 134
Lógica de programación de la página de gestión de ventas del usuario “propietario” 134
4.3.5. Desarrollo del sprint 5. Análisis y Planificación de pruebas 137
3.1.7.25. Planificación del sprint 5 137
3.1.7.26. Tareas del sprint 5 138
3.1.7.27. Ejecución del sprint 5 138
3.1.7.28. Resultados 139
4.3.5.1.1. Objetivo de las pruebas 139
4.3.5.1.2. Cronograma de actividades para las pruebas 140
4.3.5.1.3. Alcance de las pruebas 140

4.3.6. Desarrollo del sprint 6. Ejecución de las pruebas 143
3.1.7.29. Planificación del sprint 6..... 143
3.1.7.30. Tareas del sprint 6 143
3.1.7.31. Ejecución del sprint 6..... 144
3.1.7.32. Resultados 145
4.3.6.1.1. Pruebas de funcionalidad..... 145
Pruebas Unitarias 152
Pruebas de integración 155
Pruebas de errores y excepciones 159
Pruebas de rendimiento..... 162
Pruebas de usabilidad..... 164
4.4. Prueba de cierre 165
4.5. Lanzamiento 166
4.5.1. Implementación..... 166
3.1.7.33. Preparación del entorno en la VPS..... 167
3.1.7.34. Despliegue del sistema web 167
3.1.7.35. Configuración de domino y subdominios 167
3.1.7.36. Configuración de certificados SSL 168
3.1.7.37. Resultado del despliegue..... 168
3.1.7.38. Fecha de implementación..... 169
CONCLUSIONES 171
RECOMENDACIONES..... 172
BIBLIOGRAFÍA 173
ANEXOS 176

INDICE DE TABLAS

Tabla 1. Metodologías Ágiles. 20
Tabla 2. Roles de la metodología Scrum. 28
Tabla 3. Características de los usuarios del sistema. 29
Tabla 4. Historia de usuario de inicio de sesión. 31
Tabla 5. Historia de usuario creación del usuario administrador. 31
Tabla 6. Historia de usuario de registro y gestión de usuarios. 31
Tabla 7. Historia de usuario de registro de productos en inventario. 32
Tabla 8. Historia de usuario de consulta de inventario de productos. 32
Tabla 9. Historia de usuario de registro de ventas. 33
Tabla 10. Historia de usuario de registro de clientes. 33
Tabla 11. Historia de usuario de gestión de usuario. 34
Tabla 12. Historia de usuario de generación de reportes. 34
Tabla 13. Historia de usuario de búsqueda y consulta. 34
Tabla 14. Historia de usuario de análisis de ventas. 35
Tabla 15.Historia de usuario de consulta de ventas realizadas, facturadas y no facturadas. 35
Tabla 16. Historia de usuario de visualización de alertas. 35
Tabla 17. Product Backlog. 38
Tabla 18. Estimación del Product Backlog. 39
Tabla 19. Tareas del sprint 1 - Diseño de alto nivel. 40
Tabla 20. Scrum diario del sprint 1 - Diseño de alto nivel. 41
Tabla 21. Tareas del sprint 2 - Diseño detallado. 48
Tabla 22. Scrum diario del sprint 2 - Diseño detallado. 49
Tabla 23. Tareas del sprint 3 -Desarrollo Backend. 76
Tabla 24. Scrum diario del sprint 3 - Desarrollo Backend. 78
Tabla 25. Tareas del sprint 4 - Desarrollo Frontend. 119
Tabla 26. Scrum diario del sprint 4 - Desarrollo frontend. 121
Tabla 27. Tareas del sprint 5 – Análisis y Planificación de pruebas. 138
Tabla 28. Scrum diario del sprint 5 - Análisis y Planificación de pruebas. 138
Tabla 29. Cronograma de actividades para las pruebas. 140
Tabla 30. Pruebas de funcionalidad. 141

Tabla 31. Pruebas unitarias. 141
Tabla 32. Pruebas de integración. 141
Tabla 33. Pruebas de errores y excepciones..... 142
Tabla 34. Pruebas de rendimiento. 142
Tabla 35. Pruebas de usabilidad. 142
Tabla 36. Tareas del sprint 6 - Ejecución de las pruebas. 143
Tabla 37. Scrum diario del sprint 6 - Ejecución de las pruebas. 144
Tabla 38. Caso de prueba 1 - Autentificación de usuario. 145
Tabla 39. Caso de prueba 2 - Ingreso a la página principal del sistema. 146
Tabla 40. Caso de prueba 3 – Ingreso al módulo de inventario. 147
Tabla 41. Caso de prueba 4 - Ingreso al módulo de proveedores. 148
Tabla 42. Caso de prueba 5 - Ingreso al módulo de gestión de ventas. 149
Tabla 43. Caso de prueba 6 - Ingreso al módulo de usuarios. 150
Tabla 44. Caso de prueba 7 - Ingreso al módulo de clientes..... 151
Tabla 45. Caso de prueba 8 - Agregar productos al inventario..... 152
Tabla 46. Caso de prueba 9 - Registrar nuevo proveedor. 152
Tabla 47. Caso de prueba 9 - Registrar nuevo usuario 153
Tabla 48. Caso de prueba 10 - Registro de clientes. 154
Tabla 49. Caso de prueba 11 - Realizar venta. 155
Tabla 50. Caso de prueba 12 - Integración de Kardex. 156
Tabla 51. Caso de prueba 13 - Generación de reportes..... 157
Tabla 52. Caso de prueba 14 - No hay productos seleccionados para realizar la venta. 159
Tabla 53. Caso de prueba 15 - Archivo no seleccionado para la carga de productos o servicios al inventario. 160
Tabla 54. Caso de prueba 16 - Formato de archivo incorrecto para agregar productos o servicios al inventario. 161
Tabla 55. Caso de prueba 17 - Carga y estrés de usuarios y peticiones con un usuario. 162
Tabla 56. Caso de prueba 18 - Carga y estrés de usuarios y peticiones con 5 usuarios. 163
Tabla 57. Caso de prueba 19 - Procesar una venta estándar. 164

Tabla 58. Caso de prueba 20 - Ver historial de ventas..... 164
Tabla 59. Prueba de cierre..... 165

INDICE DE FIGURAS

Gráfico 1. Dirección local comercial Isabelita. Fuente: Google Maps. 16
Gráfico 2. Funcionamiento de la aplicación según el modelo MVC. Fuente: Mikaelson (2016) 45
Gráfico 3. Diseño de la navegación web del sistema. 46
Gráfico 4. Diagrama Entidad - Relación Lógico. 47
Gráfico 5. Modelo Entidad - relación de la base de datos. 47
Gráfico 6. Caso de uso - Gestión de la configuración – Usuario Administrador.. 51
Gráfico 7. Caso de uso - Visualización del Dashboard – Usuario Propietario. 52
Gráfico 8. Caso de uso - Gestión de inventario – Usuario Propietario. 52
Gráfico 9. Caso de uso - Gestión de clientes – Usuario Propietario. 53
Gráfico 10. Caso de uso - Gestión de ventas – Usuario Propietario. 53
Gráfico 11. Caso de uso - Gestión de reportes – Usuario Propietario. 54
Gráfico 12. Caso de uso - Gestión de usuarios – Usuario Propietario. 54
Gráfico 13. Caso de uso - Gestión de proveedores – Usuario Propietario. 55
Gráfico 14. Caso de uso - Visualización de alertas - Usuario Propietario 55
Gráfico 15. Caso de uso - Consulta de ventas realizas – Usuario Contador. 56
Gráfico 16. Caso de uso - Visualizar dashboard – Vendedor. 56
Gráfico 17. Caso de uso -Gestión de inventario - Vendedor 57
Gráfico 18. Caso de uso - Gestión de clientes - Vendedor. 57
Gráfico 19. Caso de uso - Gestión de ventas - Usuario Vendedor. 58
Gráfico 20. Diagrama de secuencia de inicio de sesión. 59
Gráfico 21. Diagrama de secuencia - Configuración del local comercial. 59
Gráfico 22. Diagrama de secuencia - Configuración de permisos y roles. 60
Gráfico 23. Diagrama de caso de uso - Registrar usuario 60
Gráfico 24. Diagrama de secuencia - Editar usuario. 61
Gráfico 25. Diagrama de secuencia - Registrar producto en inventario. 61
Gráfico 26. Diagrama de secuencia - Editar producto de inventario. 62
Gráfico 27. Diagrama de secuencia - Registrar proveedor. 62
Gráfico 28. Diagrama de secuencia -Editar Proveedor. 63
Gráfico 29. Diagrama de secuencia - Registrar cliente. 63
Gráfico 30. Diagrama de secuencia - Editar cliente. 64

Gráfico 31. Diagrama de secuencia - Gestionar ventas. 64
Gráfico 32. Diagrama de secuencia - Gestionar reportes. 65
Gráfico 33. Diagrama de secuencia - Visualizar alertas. 65
Gráfico 34. Diagrama de secuencia – Consulta de ventas realizadas. 66
Gráfico 35. Diagrama de secuencia - Registro de cliente por vendedor. 66
Gráfico 36. Diagrama de secuencia - Editar cliente por vendedor. 67
Gráfico 37. Diagrama de secuencia - Consultar inventario. 67
Gráfico 38. Diagrama de secuencia - Registro de venta. 68
Gráfico 39. Estructura de la vista principal del sistema. 70
Gráfico 40. Inicialización de dependencias para el entorno Backend. 85
Gráfico 41. Instalación de dependencias de node.js para el desarrollo del frontend. 130
Gráfico 42.Caso de prueba 17 - Carga y estrés de usuarios y peticiones. .... **¡Error! Marcador no definido.**
Gráfico 43. Configuración del servidor virtual privado. 167
Gráfico 44. Configuración de dominio y subdominios. 168
Gráfico 45. Configuración de certificados SSL. 168
Gráfico 46. Servicios del sistema On-line desplegados. 169
Gráfico 47. Servidor virtual privado activo y ejecutando el sistema. 169

INTRODUCCIÓN

La implementación de un sistema web para la gestión de ventas en el local comercial "ISABELITA" de la ciudad de Guaranda es un proyecto de carácter tecnológico que se enmarca en la línea de Ingeniería de Software, Redes y Telecomunicaciones, y las sub líneas de Diseño e implementación de sistemas de información.

Las nuevas tecnologías de información y comunicación han evolucionado drásticamente, permitiendo automatizar procesos manuales y repetitivos a través de soluciones de software. Este proyecto busca aprovechar estos avances tecnológicos para desarrollar un sistema que satisfaga las necesidades de gestión de ventas que permita registrar, controlar el inventario de los productos del local comercial "ISABELITA" para mejorar la atención de sus clientes, manteniendo la información actualizada del stock de productos, estadísticas de los productos más vendidos, como funcionalidades para el registro y control de inventario, generación de reportes, y atención al cliente, brindando una experiencia ágil, eficiente y segura para el propietario del local.

Para el desarrollo, se ha optado por utilizar la metodología ágil Scrum, la cual permite llevar a cabo el proyecto de forma iterativa e incremental, con una retroalimentación constante por parte del cliente. Esta metodología se enfoca en la entrega de valor en cada iteración, permitiendo una adaptación ágil a los cambios en el proyecto.

En este documento, se explicará a detalle el proceso de desarrollo del sistema web para la gestión de ventas. En el Capítulo I, se aborda la formulación general del proyecto, describiendo el problema, la justificación y los objetivos. En el Capítulo II, se presenta el marco teórico, incluyendo investigaciones previas, teorías y conceptos relevantes. En el Capítulo III, se detalla la metodología de desarrollo utilizando Scrum y las técnicas e instrumentos para la definición de los requerimientos y necesidades de la aplicación web. En el Capítulo IV, se establece la ingeniería del proyecto y las pruebas para evaluar el correcto funcionamiento del sistema desarrollado.

RESUMEN

Este proyecto tecnológico automatiza la gestión de ventas en el local comercial Isabelita de la ciudad de Guaranda, superando las limitaciones de los procesos manuales que afectan la toma de decisiones y la competitividad del negocio, la implementación del software se basa en modelos de desarrollo que ofrecen un marco de trabajo controlado para sistemas informáticos.

El sistema se desarrolló utilizando la metodología ágil de desarrollo de software Scrum que brinda un marco de trabajo para la construcción de sistemas rápidos y un mayor acercamiento con el cliente, las herramientas de recopilación de datos aplicadas permitieron el levantamiento de los requerimientos para desarrollar una aplicación informática de calidad.

Se uso la arquitectura basada en el Modelo Vista Controlador (MVC), usada para crear aplicaciones rápidas y escalables, donde se detalla la estructura del sistema y las funcionalidades, separando la lógica de negocio con la lógica de presentación.

Para la construcción del sistema se utilizó el gestor de base de datos PostgreSQL que almacena y gestiona información de ventas, clientes, productos, proveedores y demás, garantizando una gestión eficiente y segura.

El lenguaje de programación que se uso es JavaScript, junto con su entorno de ejecución Node JS y su biblioteca React JS, usada para desarrollar aplicaciones web dinámicas. Los resultados indican una reducción del tiempo en el registro de ventas, la precisión del inventario y una mejora en la atención al cliente.

Finalmente, se establecen las conclusiones y recomendaciones que abordan los objetivos planteados.

ABSTRACT

This technology project automates sales management at the Isabelita commercial establishment in the city of Guaranda, overcoming the limitations of manual processes that affect decision-making and business competitiveness. The software implementation is based on development models that offer a controlled framework for computer systems.

The system was developed using the Scrum agile software development methodology, which provides a framework for building rapid systems and closer client interaction. The data collection tools applied allowed for the gathering of requirements to develop a high-quality computer application.

A Model-View-Controller (MVC) architecture was used to create fast and scalable applications. This details the system's structure and functionalities, separating business logic from presentation logic.

PostgreSQL, a database management system, was used to build the system. It stores and manages sales, customer, product, supplier, and other information, ensuring efficient and secure management.

The programming language used is JavaScript, along with its Node.js runtime environment and React.js library, used to develop dynamic web applications. Results indicate a reduction in sales registration time, improved inventory accuracy, and enhanced customer service.

Finally, conclusions and recommendations addressing the stated objectives are presented.

CAPÍTULO I

FORMULACIÓN GENERAL DEL PROYECTO

1.1. Tema

Sistema On-line para la gestión de ventas del local comercial ISABELITA en la ciudad de Guaranda.

1.2. Descripción del Problema

El local comercial ISABELITA de la ciudad de Guaranda ha experimentado un crecimiento notable en la venta de sus productos, lo que es un indicador positivo de su popularidad y aceptación entre los clientes.

Actualmente el local realiza sus procesos de forma manual, registrando sus ventas y compras en cuadernos y hojas electrónicas de Excel, lo que conlleva una serie de desafíos y problemas tales como la desactualización del inventario, la demora en la atención al cliente en el momento del despacho de los productos y el cobro.

El propietario no tiene acceso a reportes de ventas que le permita analizar e identificar los productos que le generan mayor volumen de comercialización, esto ocasiona gastos innecesarios o excesivos en las adquisiciones, afectando negativamente a la rentabilidad del establecimiento.

La incapacidad de realizar un seguimiento adecuado de las transacciones dificulta mantener registros precisos de clientes y generar informes útiles para la toma de decisiones.

La competencia en el sector comercial minorista también es un desafío, la incapacidad de ofrecer un servicio rápido y óptimo puede limitar su capacidad para competir con otros locales comerciales, tiendas y comercios.

Por tal motivo se identificó la necesidad de desarrollar un sistema On-line para la gestión de ventas, que beneficia a su propietario al contar con una herramienta automatizada que le permita mantener actualizado su stock de productos, registro de proveedores, control de ventas, usuarios, clientes, y generación de reportes para mejorar su capacidad de atención.

1.3. Justificación

Gestionar la información de ventas es de suma importancia en el entorno comercial, en un mundo en constante cambio, el control y seguimiento de las mismas se ha convertido en uno de los aspectos más cruciales para el éxito empresarial debido a que ofrece una solución eficiente al centralizar y automatizar esta información, lo que a su vez ahorra tiempo y recursos en la administración (Catalán & Esparza, 2022)

El desarrollo de este sistema web para la gestión de ventas en el local comercial ISABELITA de la ciudad de Guaranda, mayormente comercial, adquiere una relevancia estratégica, que proporciona una serie de beneficios significativos tanto para el propietario como para sus clientes.

La razón principal que sustentan el desarrollo de este proyecto tecnológico es la optimización de procesos internos del establecimiento como el control del inventario, proveedores, usuarios y ventas, facturación, además que mejoraría la experiencia de sus clientes al recibir una atención de calidad.

La capacidad para recopilar y analizar datos relevantes sobre las ventas y las preferencias de los compradores será fundamental para que el propietario tome las mejores decisiones en beneficio de su negocio, esta información le permitirá ajustar su oferta y, en general, adaptar su estrategia comercial a las necesidades y comportamientos de su público objetivo.

El proyecto fue desarrollado con la finalidad de implementar una solución que permita mejorar el manejo de las actividades comerciales del local, de manera ágil, segura, precisa y confiable.

1.4. Objetivos

1.4.1. General

Implementar un sistema On-line para la gestión de ventas en el local comercial ISABELITA de la ciudad de Guaranda.

1.4.2. Específicos

- Analizar los requerimientos funcionales, no funcionales y las necesidades del local comercial ISABELITA en los procesos de comercialización.
- Evaluar la metodología de desarrollo de software más adecuada para la implementación del sistema On-line para la gestión de ventas.
- Diseñar el sistema On-line para la gestión de ventas.
- Desarrollar el sistema online para la gestión de ventas del local comercial ISABELITA.
- Aplicar un plan de pruebas en el sistema On-line de gestión de ventas para asegurar su correcto funcionamiento.

CAPÍTULO II

MARCO TEÓRICO

Antecedentes

Los sistemas de gestión de ventas son una tendencia creciente en el sector comercial, permitiendo a los negocios mejorar su eficiencia, reducir costos y aumentar su competitividad en el mercado.

Existen muchos sistemas de gestión de ventas que ofrecen soluciones para pequeñas y medianas empresas. La variedad existente se debe a que cada uno está configurado para resolver problemas específicos de las empresas (Furman & Aura, 2022).

En un análisis realizado por Rivera (2021), en su trabajo "SISTEMAS DE GESTIÓN DE VENTAS E INVENTARIOS", con el propósito de mejorar las operaciones de las micro, pequeñas y medianas empresas (MIPYMEs) a través del desarrollo e implementación de un sistema web para la administración y gestión de los inventarios. El estudio contempla una investigación no experimental, donde se analiza a 41 personas pertenecientes a empresas que se dedican al rubro de las prendas de vestir y el calzado, determinando que, en ausencia de un sistema de gestión de ventas e inventarios, una empresa no podría operar de manera óptima, ya que no podría automatizar sus procesos ni tener una visión clara y precisa de su situación actual. Además, resaltó que la implementación de sistemas de gestión de ventas e inventario tiene un impacto positivo en la calidad de los servicios de una empresa, permitiendo que los procesos se completen de manera más rápida y eficiente.

Según García (2022), en su proyecto de titulación de la Universidad Regional Autónoma de los Andes, titulada "SISTEMA PARA LA GESTIÓN DE LAS VENTAS EN LA EMPRESA SUPER 10 UBICADA EN LA CIUDAD DE SANTO DOMINGO DE LOS TSÁCHILAS", empresa dedicada a la comercialización de productos de papelería y bazar, con el objetivo de desarrollar un sistema para la gestión de ventas para dicha empresa, bajo un análisis cuali-cuantitativo para conocer los principales carencias de la empresa y poder ofrecer una solución; menciona que el desarrollo e implementación del sistema de control

de ventas ofreció una solución integral para los problemas de inventario y facturación, brindando un mayor control sobre los productos y la cartera de clientes, y proporcionando informes y listados para asegurar una buena atención al cliente.

En el proyecto de Rojas (2020), "DESARROLLO DE SISTEMA DE CONTROL DE VENTAS EN LA EMPRESA DISCOVERSELLER S.A." realizado en la ciudad de Guayaquil con la finalidad de desarrollar un sistema para el control de ventas en una empresa que se dedica a la venta de equipos informáticos; se realiza un análisis de requerimientos funcionales y no funcionales, identificando la necesidad de cuatro módulos necesarios: mantenimiento, almacén, ventas y reportería. Como conclusión de este trabajo se menciona que con la implementación del sistema se logró centralizar la información, ayudando de manera significativa a los usuarios acceder a ella de manera ágil y eficiente, garantizando el control y reporte de cada una de las ventas, dependiendo de las fechas o por cada empleado.

El estudio realizado por Quiroz (2021), en su trabajo "DESARROLLO DE APLICATIVO WEB PARA MEJORAR LA GESTIÓN DE ALMACÉN Y VENTAS EN UNA EMPRESA PRIVADA", en una empresa dedicada a la comercialización y distribución de equipos de alta tecnología, con el fin de optimizar los procesos de almacén y ventas por la deficiencia de información a través del desarrollo de una solución web utilizando la metodología EssUP, compuesta por cuatro fases (Incepción, Elaboración, Construcción y Transición), muestra que la implementación de la aplicación web para la gestión de ventas e inventarios ha reducido significativamente el tiempo dedicado a varias tareas administrativas, como la elaboración de documentos, registros de clientes y proveedores, cotizaciones, facturas, registros de ventas y compras, inventario y reportes. En promedio, el tiempo dedicado a estas tareas se redujo de 40-60 minutos a 10-20 minutos. Además, la satisfacción de los clientes también mejoró después de la implementación de la solución web.

Considerando los análisis realizados a los proyectos relacionados con la implementación de un sistema de gestión de ventas se ha demostrado que la ausencia de un sistema de gestión de ventas e inventarios impide que las empresas operen de manera óptima, mientras que su implementación tiene un impacto positivo en la calidad de los servicios, permitiendo procesos más rápidos y

eficientes. Además, se menciona que la implementación de soluciones web para la gestión de ventas e inventarios reduce significativamente el tiempo dedicado a tareas administrativas y mejora la satisfacción de los clientes.

Científico

1.1.1. Ingeniería de software.

La ingeniería de software es un término que abarca todos los aspectos relacionados con la producción de software y no se refiere únicamente a la programación de sistemas, sino a todos los procesos que conforman el ciclo de vida (Cubero, 2020). Las áreas de conocimiento fundamentales para la ingeniería de software abarcan:

- Desarrollo de software: requerimientos, diseño, construcción, pruebas y mantenimiento.
- Gestión de software: gestión de configuración, gestión de ingeniería y proceso de ingeniería.
- Calidad y economía: calidad del software y economía de la ingeniería de software.
- Fundamentos: fundamentos de computación, fundamentos matemáticos y fundamentos de ingeniería.
- Práctica profesional: práctica profesional de ingeniería de software.

La aplicación de la ingeniería de software se ha expandido de manera significativa y actualmente es esencial para que las empresas y organizaciones deseen mantener su competitividad y sobrevivir en los mercados globalizados (Vega, 2024).

1.1.2. Ciclo de vida de software.

La industria del desarrollo de software abarca distintos procedimientos (como análisis, implementación, mantenimiento) y servicios (como formación, documentación). Con base en esto, se conoce como ciclo de vida del software al proceso de concebir, crear e implementar una aplicación (programa, sistema, software). Este ciclo debe resultar en un software fiable y de alta calidad que satisfaga y supere las expectativas del cliente, logrando los objetivos para los cuales fue concebido y dentro del presupuesto inicialmente estimado (Domínguez et al., 2022)

1.1.3. Modelos de desarrollo de software.

El desarrollo de software ha sido establecido bajo el control de parámetros de revisión estrictos y metodológicos con la finalidad de la construcción de sistemas de calidad, que logren satisfacer las necesidades de los usuarios finales y principalmente que resuelvan la problemática por la cuales fueron desarrollados (Molina, 2020).

Los modelos de desarrollo de software proporcionan un enfoque estructurado usado para controlar el proceso de construcción de sistemas de información, ofreciendo un marco de referencia que integra una filosofía de desarrollo con las herramientas y técnicas necesarias para la asistencia en el proceso (Armijos & Fernández, 2022).

1.1.4. Sistemas de gestión de ventas

En una empresa, un sistema para la gestión de ventas estandariza y coordina los procesos de venta, permitiendo una mayor eficiencia, control y visibilidad en la gestión de la cartera de clientes, inventario, facturación y reportes, lo que conduce a una mejora en la toma de decisiones y un aumento en la productividad y rentabilidad de la organización (Mecías, 2022).

1.1.5. Control de inventario

El control de inventario se basa en procedimientos o técnicas para garantizar que las existencias en el almacén de una empresa sean suficientes y llevar una planificación y estimación de reabastecimiento, donde llevar el control del inventario eficientemente optimiza el ciclo de compra y la planificación de del flujo de caja (SafetyCulture, 2024).

1.1.6. Métodos de control de inventarios

Según Flat (2019), el control de inventario implica técnicas y procedimientos para la supervisión de los productos o materiales, de manera que este permita conocer la cantidad exacta de artículos y conciliar las existencias físicas con las registradas en la documentación.

1.1.7. Análisis ABC

El análisis ABC es un método para llevar el control de inventario que consiste en dividir en tres categorías los productos según su importancia, cantidad y valor, siendo más fácil identificar los productos más valiosos y con mayor necesidad de atención (Ortega et al., 2022).

- Clase A: Productos de lujo (alto precio, baja cantidad)
- Clase B: Productos medios (precio medio, cantidad media)
- Clase C: Productos básicos (bajo precio, alta cantidad)

1.1.8. Stock de seguridad

El stock de seguridad es una técnica de reserva adicional de productos que se mantiene en el almacén como una medida de precaución para enfrentar las fluctuaciones impredecibles del mercado (Ortega et al., 2022). La cantidad de stock de seguridad debe ser mayor que la demanda promedio del producto.

Este stock actúa como un colchón protector en caso de que la demanda de los clientes supere las expectativas, y también proporciona una capa de seguridad en caso de problemas en el suministro, como retrasos en el envío, garantizando que se pueda mantener el nivel de servicio al cliente incluso en situaciones impredecibles (Flat, 2019).

Conceptual

1.1.9. Automatización

Cuando se menciona la automatización, se hace referencia a los sistemas que facilitan la transferencia de tareas y actividades de producción, que normalmente son realizadas por operadores humanos, a un conjunto de elementos tecnológicos disponibles, suponiendo la mejora de las condiciones laborales del personal, la optimización del tiempo y el aumento de la seguridad en los procesos (Domínguez et al., 2022).

1.1.10. Arquitectura de software

La arquitectura de software se refiere a la configuración fundamental de un sistema de software, que define su estructura y organización a un nivel alto, es una decisión crítica, ya que debe cumplir con los requisitos del sistema, tanto funcionales como no funcionales, y tiene un impacto significativo en el desempeño, escalabilidad y mantenimiento del software (Tian et al., 2022).

1.1.11. Backend

El backend, aunque no es visible, es una parte esencial de una aplicación, ya que se encarga de manejar la lógica y el procesamiento de datos necesarios para que todo funcione de manera correcta y segura, se encarga de diversas tareas fundamentales, como almacenar y recuperar datos de una base de datos, procesar formularios,

autenticar usuarios y gestionar la seguridad del sitio, interactuando con el servidor (Maldeadora, 2018).

1.1.12. Base de datos

Una base de datos es un conjunto organizado de datos almacenados de manera estructurada, que permite el acceso, la gestión y la recuperación de información de manera eficiente y segura, comúnmente están sujetas a un sistema gestor de base de datos (SGBD), que garantiza la administración y control de la información ingresada o extraída (Valverde et al., 2019).

1.1.13. Base de datos relacionales.

Una base de datos de datos relacional o también llamadas bases de datos SQL, es un conjunto de datos organizados que se gestionan en tablas a través de filas y columnas, se rigen bajo un concepto de entidades y relaciones y a diferencia de las no relaciones usan un lenguaje de consultas SQL para interactuar entre sí (Bermúdez, 2020).

1.1.14. Desarrollo web

En la ingeniería de software, una aplicación web se refiere a un tipo de programa que permite a los usuarios interactuar con él a través de un navegador, accediendo a un servidor web a través de internet o una intranet, y que está escrito en un lenguaje que los navegadores web pueden entender y ejecutar (García et al., 2021).

1.1.15. Frontend

El frontend se refiere a la parte de un sistema, sitio web o aplicación a la que el usuario tiene acceso directo e interactúa. En el contexto del desarrollo web, el frontend abarca todas las tecnologías que se ejecutan en el navegador y son responsables de la interacción y la interfaz de usuario (Maldeadora, 2018).

1.1.16. Framework.

Un framework es un conjunto de herramientas, librerías y reglas que se utilizan en el desarrollo de software para facilitar y agilizar el proceso de creación de aplicaciones, proporciona una estructura base y una serie de funcionalidades predefinidas que permiten a los desarrolladores enfocarse en la lógica específica de su aplicación en lugar de tener que escribir todo el código desde cero (Unir, 2023).

1.1.17. Facturación electrónica

La facturación electrónica en Ecuador es un método alternativo para emitir comprobantes de venta, retención y documentos relacionados, que cumple con los estándares legales y reglamentarios establecidos por el SRI, asegurando la autenticidad del emisor y la integridad del contenido, gracias a la inclusión de la firma electrónica del emisor en cada comprobante (SRI, 2024)

1.1.18. Gestor de base de datos

Un gestor de bases de datos (SGBD) es un sistema que posibilita la creación, gestión y supervisión de bases de datos, así como la selección y manipulación de las estructuras necesarias para almacenar y buscar información de la manera más eficiente posible (Rouse, 2019).

1.1.19. JavaScript

JavaScript es un lenguaje de programación versátil y flexible que se ejecuta de manera ligera y eficiente, ya sea mediante interpretación o compilación en tiempo real, aunque se asocia comúnmente con la creación de scripts dinámicos para páginas web, su aplicación se extiende más allá del navegador, siendo utilizado en una variedad de entornos y contextos (MDN, 2024).

1.1.20. Lenguajes de programación

Es un lenguaje diseñado para expresar un conjunto de instrucciones secuenciales que un programador debe proporcionar a una computadora, con el objetivo de controlar el funcionamiento de una máquina (Zamora, 2021).

1.1.21. Modelo Vista Controlador

El Modelo Vista Controlador (MVC) es un patrón en el diseño de software utilizado para la implementación de interfaces de usuario, datos y lógica de control, que propone un enfoque de separación de la lógica del negocio y presentación (Sánchez, 2020).

1.1.22. Patrones de arquitectura

Un patrón arquitectónico es una idea abstracta que resume los elementos de una arquitectura de software. los patrones arquitectónicos suelen ser definidos de manera precisa y estar disponibles de forma común, lo que permite a los desarrolladores de software aprovechar su conocimiento y experiencia para diseñar sistemas más efectivos (Sánchez, 2020).

1.1.23. PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos relacionales, gratuito y accesible que se caracteriza por ser altamente personalizable y cumplir estrictamente con los estándares de lenguaje SQL (Armijos & Fernández, 2022).

1.1.24. Proceso de ventas

Para Barreto & Vargas (2021), se trata de un proceso sistemático y estructurado que sigue el vendedor para interactuar con un posible cliente, con el objetivo de influir en su decisión y lograr una respuesta deseada, como la realización de una compra.

1.1.25. Régimen Simplificado para Emprendedores y Negocios Populares (RIMPE)

Régimen Simplificado para Emprendedores y Negocios Populares (RIMPE) es un mecanismo implementado por el Servicio de Rentas Internas (SRI) en Ecuador, con el objetivo de facilitar y fomentar la formalización de las microempresas en el país y los contribuyentes bajo este régimen están obligados a emitir documentos fiscales (SRI, 2024).

1.1.26. React JS

React es un framework de JavaScript ampliamente utilizado para crear interfaces de usuario dinámicas y eficientes, que se caracteriza por un enfoque modular que permite descomponer la interfaz de usuario en componentes independientes y reutilizables, lo que facilita su desarrollo y mantenimiento (Catalán & Esparza, 2022).

1.1.27. Seguridad del software

La seguridad del software implica la implementación de medidas y mecanismos para proteger el software de posibles amenazas y ataques. Se centra en asegurar que el software sea resistente a vulnerabilidades y que los datos almacenados y procesados por el software estén protegidos de accesos no autorizados. Abarca diferentes aspectos, desde el diseño y desarrollo, hasta su implementación, mantenimiento y la gestión de los datos (Moisset, 2023).

1.1.28. Sistemas de información

Un sistema de información se define como un conjunto de componentes interconectados que recopilan, procesan, almacenan y distribuyen información relevante para los usuarios, y que, además, ofrecen un mecanismo de retroalimentación para garantizar la precisión y eficacia (Gómez, 2020).

1.1.29. Ventas

La venta es el resultado de una serie de actividades coordinadas y ejecutadas en diferentes etapas, donde un vendedor profesional deberá adoptar un enfoque centrado en el cliente, con una estructura de trabajo flexible y personalizada que le permita mantener un seguimiento del progreso en cada etapa y facilitar el avance en el proceso de venta (Barreto & Vargas, 2021).

3.3.1. API REST

Una API (Interfaz de Programación de Aplicaciones) se refiere a una interfaz que conecta aplicaciones móviles o web con la lógica de un software, actuando como un puente de comunicación que permite la interacción asíncrona entre la parte frontal (Frontend) y la parte posterior (Backend) de un sistema (Heredia, 2022). En una API REST las peticiones del cliente y las respuestas del servidor se realiza usando el formato JSON y a través del protocolo HTTP.

3.3.2. Protocolo HTTP

HTTP (Protocolo de Transferencia de Hipertexto) es un estándar global esencial para entender cómo funcionan las API Rest; cualquier sitio web, foro, revista o repositorio virtual en línea necesita del prefijo HTTP para ser accesible a través de un navegador. Cuando una API se comunica con una interfaz de cliente, lo hace a través de direcciones URL (Heredia, 2022).

Legal

La fundamentación legal de este trabajo se basa en la siguiente reglamentación:

"Código Orgánico de Economía Social de los Conocimientos, Creatividad e Innovación."

• **Artículo 142.-Tecnologías libres**. - Se considera software de código abierto aquel que tiene una licencia que garantiza al usuario acceso ilimitado al código fuente y le permite utilizarlo para cualquier fin o propósito.

Especialmente concede a los usuarios una serie de libertades fundamentales, incluyendo las siguientes:

a) La libertad de utilizar el software para cualquier propósito.

b) La libertad de estudiar el funcionamiento del software, y la posibilidad de modificarlo para adaptarlo a las necesidades de cada persona.

Por otra parte, el proyecto se fundamenta en que no existen obstáculos legales que impidan la implementación del sistema para la gestión de ventas en el local comercial lo que permite desarrollar y ejecutar el proyecto sin restricciones jurídicas. La local comercial ISABELITA ha otorgado su consentimiento explícito para la recopilación y utilización de la información necesaria para el desarrollo del sistema, garantizando la colaboración y apoyo necesario para el éxito del proyecto.

Georreferencial

El tema de trabajo de titulación e integración curricular se aplicará en la ciudad de Guaranda en el local comercial ISABELITA, ubicada geográficamente:

Dirección: Calle Azuay y 7 de mayo, frente a la Peluquería Scarlet.

Latitud: 1°35'26"S

Longitud: 79°00'01"W

Elevanción: 2.683,11 m

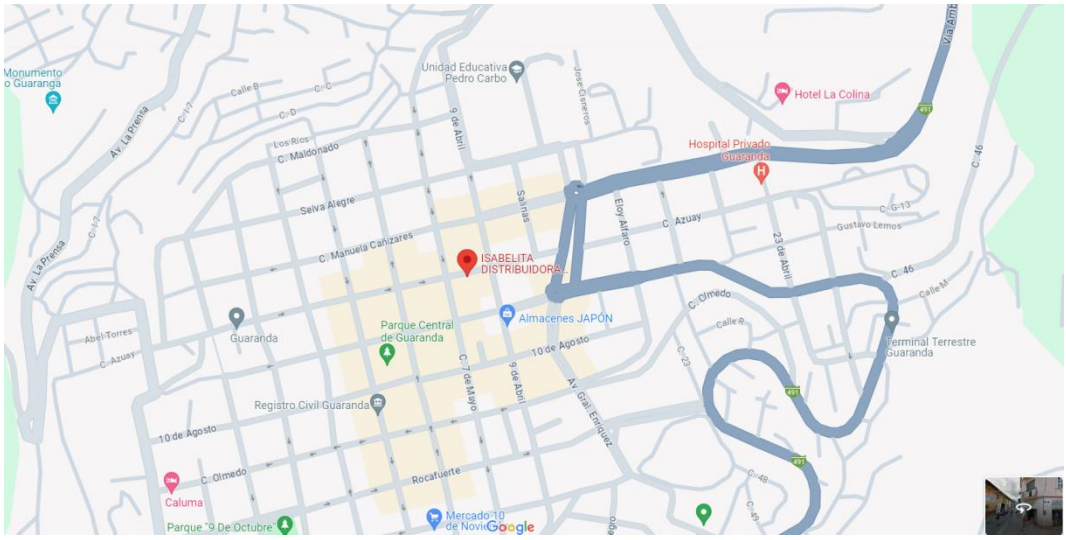


Gráfico 1. Dirección local comercial Isabelita. Fuente: Google Maps.

CAPITULO III

METODOLOGÍA

3.1. Metodología de Desarrollo de Software

3.1.1. Metodologías Agiles

Según Gómez (2020), una de las características clave de cualquier tecnología ágil es la capacidad de proporcionar una plataforma que integre los recursos de desarrollo, facilite la comunicación rápida, flexible y efectiva entre los miembros del equipo y de respuesta a los cambios en los requisitos del proyecto.

3.1.2. Metodología XP

Extreme Programming (XP), se enfoca en crear un producto que se ajuste exactamente a las necesidades del cliente, involucrándolo activamente en el proceso, caracterizándose por una comunicación constante, respuesta rápida a cambios, planificación abierta y flexible, y priorización del software funcional sobre la documentación, donde el éxito del proyecto depende fundamentalmente de los requisitos del cliente y el trabajo del equipo del proyecto (Canive, 2020).

Extreme Programming tiene ventajas como:

1. Flexibilidad y adaptabilidad: permite adaptarse rápidamente a los cambios de requisitos.
2. Enfoque en la calidad: XP enfatiza las pruebas unitarias continuas, la integración continua y la revisión de código para garantizar un software de alta calidad.
3. Comunicación efectiva: promueve la comunicación constante y directa entre el equipo de desarrollo y el cliente (Canive, 2020).
4. Entrega incremental: se basa en la entrega de software en incrementos pequeños y frecuentes para obtener retroalimentación temprana y brindar valor al cliente de manera constante.
5. Mayor productividad: fomenta la colaboración y la responsabilidad compartida, eliminando actividades innecesarias o burocráticas.
6. Reducción de riesgos: ayuda a mitigar los riesgos al detectar y abordar rápidamente los problemas durante el desarrollo del software.

3.1.3. Metodología Scrum

Scrum es un marco de trabajo ágil que permite abordar problemas complejos de manera eficiente y creativa, entregando productos de alto valor a través de la colaboración y el trabajo en equipo, con un enfoque basado en sprints y la mejora continua, utilizando valores, roles y pautas específicas centrados en el cliente (Martins, 2024).

Algunas características de Scrum son las siguientes:

- Control sobre el proceso empírico
- Cada miembro del equipo Scrum asume la responsabilidad de sus tareas y trabajo.
- Trabajo en conjunto durante y después del sprint para obtener los mejores resultados.
- Priorizar el trabajo desde el inicio del proceso Scrum para ofrecer el mayor valor comercial.
- Actividades basadas en el tiempo para establecer una duración limitada para el trabajo y mejorar el trabajo futuro.
- Desarrollo iterativo: Construir de manera iterativa para adaptarse a las necesidades del cliente y modificar el producto y sus resultados en función de la priorización basada en el valor (Martín Gómez, 2020).

3.1.4. Metodología Kanban

Kanban es una forma de ayudar a los equipos a encontrar un equilibrio entre el trabajo que necesitan hacer y la disponibilidad de cada miembro del equipo, se basa en una filosofía centrada en la mejora continua, donde las tareas se "extraen" de una lista de acciones pendientes en un flujo de trabajo constante (Martins, 2024).

Algunos veneficios que presenta Kanban son.

- **Visibilidad del trabajo:** Ofrece una visión general del trabajo del equipo, mostrando el flujo de trabajo y el estado de las tareas.
- **Claridad en equipos remotos:** Facilita la visibilidad del trabajo en equipos remotos, mejorando la comunicación y la coordinación.
- **Flexibilidad:** Promueve la mejora continua y la adaptación a los cambios, haciendo que los equipos sean más ágiles y dinámicos.

3.1.5. Metodología Desarrollo de sistemas dinámicos (DSDM)

DSDM es un marco de trabajo diseñado para ofrecer la solución adecuada en el momento preciso, emplea un enfoque iterativo, dividiendo el proyecto en intervalos cortos con entregables definidos para cada etapa (Tetteh, 2024) . Además, establece roles claros con responsabilidades específicas dentro de estos plazos.

Algunos beneficios de DSDM incluyen:

- Mayor capacidad de adaptación y flexibilidad, lo que permite gestionar mejor los requisitos en constante cambio.
- Entrega regular e incremental de software, acelerando la obtención de beneficios.
- Mejora en la colaboración entre las partes interesadas, los desarrolladores y los usuarios finales.
- Reducción de riesgos gracias a un desarrollo iterativo y ciclos frecuentes de retroalimentación.
- Mejor supervisión y control del proyecto, respaldados por roles y responsabilidades claramente definidos.

3.1.6. Metodología Desarrollo orientado a funcionalidades (FDD)

El Desarrollo de Funcionalidades (FDD) es un enfoque que permite crear software de manera iterativa e incremental, centrándose en las características que aportan valor al cliente. Este método cuenta con la figura de un jefe de proyecto y una fase de arquitectura, y se guía por modelos. También está dirigida por modelos, y es de iteraciones cortas (Fernández et al., 2019).

Algunos veneficios que presenta FDD son.

- Durante las fases de estudio de viabilidad y negocio se evalúa la viabilidad del proyecto y su alineación con los objetivos organizacionales (Tetteh, 2024).
- FDD fomenta un enfoque iterativo e incremental, lo que significa que el producto se desarrolla y mejora gradualmente en lugar de intentar completar todo el proyecto de una vez.

- FDD ayuda a identificar y abordar los riesgos de manera más efectiva al dividir el proyecto en funcionalidades manejables y funcionalidades priorizadas.
- Al concentrarse en la construcción de funcionalidades completas y probadas, FDD fomenta la calidad desde el principio.
- El FDD fomenta la comunicación y colaboración en el equipo al definir roles y responsabilidades claras.
- La estructura iterativa del FDD brinda flexibilidad y capacidad de adaptación, permitiendo ajustar planes y estrategias según los comentarios y necesidades que surjan durante el proceso.

Tabla 1. Metodologías Ágiles.

Característica	**XP**	**Scrum**	**Kanban**	**DSDM**	**FDD**
Define alcance mediante "historias de usuario"	X				
Reuniones diarias de seguimiento	X	X	X		
Programación en parejas	X				
Bloques temporales (iteraciones)		X		X	X
Visualiza el flujo de trabajo			X		
Limita la cantidad de trabajo en proceso			X		
Seguimiento del tiempo			X		
Identifica cuellos de botella			X		X
Colaboración del cliente	X	X		X	
Entregar a tiempo				X	
Desarrollar iterativamente	X	X	X	X	X
Enfoque en funcionalidades					X

División en etapas					X
Modelado y diseños detallados					X
Comunicación continua y clara				X	
Roles y responsabilidades definidos	X	X	X	X	X

Según la tabla de comparación, Scrum ofrece un mayor acercamiento con el cliente en entregas iterativas (sprints), una estructura clara de roles y responsabilidades, lo que permite una adaptación rápida a cambios y un seguimiento continuo del progreso del proyecto, que es ideal para el tipo de sistemas que llevan la gestión de ventas, donde pueden surgir nuevas necesidades.

Carrillo et al. (2021), mencionan que la metodología Scrum es la que destaca como la usada con más frecuencia entre las metodologías ágiles, debido a que ofrece la capacidad de adaptar las formas de trabajo a las necesidades específicas del proyecto, permitiendo respuestas rápidas y flexibles para adaptarse a los cambios del cliente o del entorno.

3.1.7. Metodología de desarrollo Scrum

Para el desarrollo del sistema web se utilizó la metodología Scrum, que se enfoca en la entrega de valor al cliente a través de iteraciones cortas y ciclos de retroalimentación.

Esta metodología favoreció la colaboración y la transparencia en el equipo, lo que permitió adaptarse rápidamente a los cambios y priorizar las características más importantes, de esta manera, se aseguró que el proyecto se desarrolle de manera ágil y flexible, con una reducción de riesgos y costos.

3.1.7.1. Teoría de Scrum

Scrum se basa en el empirismo y el pensamiento Lean involucrando a grupos de personas que colectivamente tienen todas las habilidades y experiencia para hacer el trabajo y compartir o adquirir dichas habilidades según sea necesario. Scrum combina cuatro eventos formales para inspección y adaptación dentro de un evento contenedor, el sprint.

Es una metodología ágil para el desarrollo de software o la gestión de proyectos, más concretamente un marco de trabajo a través del cual las personas pueden abordar problemas complejos adaptativos, a la vez que se entregan productos de forma eficiente y creativa con el máximo valor (Schwaber & Sutherland, 2020)

3.1.7.2. Caracterización de Scrum

Scrum utiliza el progreso real de un proyecto para planificar y concertar los lanzamientos, al final de cada sprint, el cliente y los miembros del equipo se reúnen para evaluar el progreso del proyecto y planear los siguientes pasos a seguir (Fernández, 2018). De esta manera, se pueden realizar ajustes o cambios en la estrategia del proyecto basados en resultados tangibles, en lugar de conjeturas o suposiciones. Filosóficamente, este énfasis continuo de evaluar las tareas finalizadas es el principal causante del éxito que tiene esta metodología entre los directores y desarrolladores (Fernández, 2018)

3.1.7.3. Modelo Scrum para Administración de proyectos de software

Scrum es una variación de la metodología ágil, la más clásica, capaz de soportar la variación de los requerimientos y reaccionar favorablemente a ellos; siempre y cuando sea bien aplicada, ésta permite la flexibilidad en el desarrollo de software y su proceso no se basa en rigurosas reglas de administración (Martins, 2024). Scrum utiliza iteraciones breves llamadas "sprints" de 1-2 semanas, con objetivos específicos y retroalimentación para mejorar el siguiente ciclo.

3.1.7.4. Equipo Scrum

Developers

Las personas del scrum team que se comprometen a crear cualquier aspecto de un increment utilizable en cada Sprint son developers. Las habilidades específicas que necesitan los developers suelen ser amplias y variarán según el ámbito de trabajo.

Product Owner

El product owner garantiza el máximo valor del resultado del trabajo del equipo Scrum, gestionando efectivamente el backlog del producto y adaptándose a las necesidades específicas de la organización y el equipo (Hernandez, 2022).

Tiene la opción de llevar a cabo las tareas asignadas o delegar la responsabilidad a otros. Sin embargo, independientemente de esta elección, sigue siendo el principal

responsable de garantizar que el trabajo se complete satisfactoriamente. Para asegurar es fundamental que toda la organización respete y respalde sus decisiones.

Scrum Master

El scrum master tiene la tarea de impulsar y mantener el scrum a flote esto implica que se encarga de asegurar que el equipo scrum sea efectivo, esto se logra apoyando al equipo para que mejore continuamente dentro del marco de trabajo de Scrum (Hernandez, 2022).

3.1.7.5. Eventos de Scrum

Cada evento en scrum es una oportunidad formal para inspeccionar y adaptar los artefactos scrum (Schwaber & Sutherland, 2020). Los eventos se utilizan en scrum para crear regularidad y minimizar la necesidad de reuniones no definidas en scrum. Lo óptimo es que todos los eventos se celebren al mismo tiempo y en el mismo lugar para reducir la complejidad.

El Sprint

Los sprints son el corazón de scrum, donde las ideas se convierten en valor. Un nuevo sprint comienza inmediatamente después de la conclusión del sprint anterior. Los sprints permiten la previsibilidad al garantizar la inspección y adaptación del progreso hacia un objetivo del producto, si el horizonte de un sprint es demasiado largo, el objetivo del sprint puede volverse inválido, la complejidad puede crecer y el riesgo puede aumentar.

Sprint Planning

El Sprint planning (planificación del sprint) inicia el sprint al establecer el trabajo que se realizará para el sprint, el scrum team también puede invitar a otras personas a asistir al sprint planning para brindar asesoramiento (Schwaber & Sutherland, 2020).

Daily Scrum

El propósito del daily scrum (scrum diario) es inspeccionar el progreso hacia el objetivo del sprint y adaptar el sprint backlog según sea necesario, ajustando el trabajo planificado entrante.

El daily scrum es un evento de 15 minutos para los developers del scrum team. Para reducir la complejidad, se lleva a cabo a la misma hora y en el mismo lugar todos

los días hábiles del Sprint. Si el product owner o scrum master están trabajando activamente en elementos del sprint backlog, participan como developers.

Los daily scrums mejoran la comunicación, identifican impedimentos, promueven la toma rápida de decisiones y, en consecuencia, eliminan la necesidad de otras reuniones.

Sprint Review

El propósito de la sprint review es inspeccionar el resultado del Sprint y determinar futuras adaptaciones. El Scrum team presenta los resultados de su trabajo a los interesados clave y se discute el progreso hacia el Objetivo del Producto.

La sprint review es el penúltimo evento del sprint y tiene un límite de tiempo de máximo cuatro horas para un sprint de un mes. Para sprints más cortos, el evento suele ser de menor duración.

Sprint Retrospective

El propósito de la sprint retrospective es planificar formas de aumentar la calidad y la efectividad.

El scrum team inspecciona cómo fue el último sprint con respecto a las personas, las interacciones, los procesos, las herramientas y su definición de terminado, se analiza qué salió bien durante el Sprint, qué problemas encontró y cómo se resolvieron (o no) esos problemas.

La sprint retrospective concluye el Sprint teniendo un tiempo limitado a máximo tres horas para un Sprint de un mes, para los sprints más cortos, el evento suele ser de menor duración (Fernández, 2018).

3.1.7.6. Artefactos de Scrum

Los artefactos de Scrum, al representar el trabajo y su valor de forma transparente, permiten una adaptación eficiente por parte de todos los involucrados.

Product Backlog

El product backlog es una lista emergente y ordenada de lo que se necesita para mejorar el producto. Es la única fuente del trabajo realizado por el scrum team.

Los elementos del product backlog que el scrum team puede dar por terminados dentro de un sprint se consideran preparados para ser seleccionados en un evento de sprint planning. Los atributos suelen variar según el ámbito del trabajo.

Sprint Backlog

El sprint backlog integra el propósito del sprint, los elementos del product backlog seleccionados para ese ciclo, y un plan de acción para entregar el incremento correspondiente, este backlog es una herramienta de planificación creada por y para los desarrolladores, ofreciendo una visión clara y actualizada del trabajo que se realizará durante el sprint para alcanzar el objetivo establecido (Hernandez, 2022).

3.2. Técnicas e Instrumentos de Recopilación de Datos

Para definir el sistema de gestión de ventas, sus requerimientos y necesidades se utilizaron los siguientes instrumentos y técnicas de recopilación de datos.

3.2.1. Entrevista

Es fundamental para recopilar información valiosa sobre los procesos de ventas actuales, los productos vendidos y las necesidades específicas del negocio.

En este caso, se realizó una entrevista con el propietario del local Wilman Remigio Gómez, para obtener información detallada y precisa sobre los procesos y necesidades del negocio. El objetivo era recopilar datos sobre los procesos de ventas actuales, los productos vendidos y las necesidades específicas del negocio. La entrevista se realizó utilizando un guion de entrevista cuidadosamente diseñado para obtener la información necesaria.

3.2.2. Observación directa

La técnica de observación se utilizó en las tres visitas realizadas al local comercial Isabelita para evidenciar los procesos de ventas, entrega de productos por los proveedores y el registro del inventario de los productos.

CAPITULO IV

INGENIERÍA DEL PROYECTO

5.1. Inicio

5.1.1. Introducción

Definir con precisión los requisitos del software es fundamental para el desarrollo del sistema web de gestión de ventas, asegurando que satisfaga las necesidades y expectativas del usuario y del propietario. La especificación de requerimientos de software proporciona una descripción detallada de los requisitos funcionales, no funcionales y de interfaz para establecer las funcionalidades de este.

5.1.2. Alcance del Producto

El desarrollo del sistema para la gestión de ventas en el local comercial ISABELITA busca aumentar la eficiencia y productividad, reduciendo los tiempos de respuesta y los errores en la gestión del inventario y las ventas. Este sistema proporcionará una herramienta eficiente y segura para gestionar de manera integral diversos aspectos clave del negocio, como el inventario, los proveedores, los pedidos y los clientes. Al optimizar estos procesos, el local podrá mejorar significativamente su competitividad. Además, el sistema mejorará la satisfacción del cliente al brindar una experiencia de compra más rápida, fluida y confiable, mejorando la comunicación con los mismos. Al facilitar un canal de consultas y atención rápido y eficiente, el local podrá responder de manera oportuna a las inquietudes de los compradores. Esto contribuirá a fortalecer los vínculos con los clientes y aumentar su nivel de satisfacción y lealtad hacia el local Isabelita.

El sistema web proporcionará al propietario del local informes y análisis precisos y actualizados sobre las ventas, los productos más demandados y las preferencias de los clientes. Esta información detallada les permitirá tomar decisiones más informadas y estratégicas en cuanto a la gestión del negocio.

El alcance de la aplicación web de gestión de ventas abarcará áreas clave como la gestión del inventario, el registro y seguimiento de las ventas, la generación de informes analíticos y la gestión de usuarios y permisos de acceso.

5.1.3. Limitaciones

- La implementación del sistema web de gestión de ventas e inventario dependerá de la disponibilidad de recursos tecnológicos y financieros.
- La seguridad del sistema web dependerá de la implementación de medidas de seguridad adecuadas para proteger la información del local.
- Para acceder al sistema web es necesario contar con métodos de autenticación y autorización de usuarios para proteger la información e integridad de los datos del local comercial.
- La conexión a Internet debe ser requerida para acceder al sistema.
- La gestión de errores y excepciones debe ser manejada, especialmente si se producen errores en la transmisión de datos o en la conexión a Internet.
- El sistema web debe estar diseñada y optimizada para que funcione al menos en la mayoría de los navegadores.

5.1.4. Definiciones y acrónimos

Sistema web de ventas: Un sistema de gestión de ventas que se accede a través de un navegador web

Control de stock en tiempo real: Un sistema que actualiza automáticamente la información del inventario disponible en tiempo real, a medida que se realizan ventas o se reciben nuevos productos (SafetyCulture, 2024).

VPS: (Virtual Private Server) es un servidor privado virtual, donde se aloja el sistema, permitiendo que funcione en un entorno virtualizado con recursos dedicados.

Docker: Plataforma que permite empaquetar aplicaciones y sus dependencias en contenedores, asegurando que funcionen en cualquier entorno y de forma aislada.

Nginx: Servidor web utilizado como **proxy inverso** para redirigir el tráfico entrante hacia los contenedores de Docker y balancear las peticiones entre diferentes instancias.

Balanceo de Carga: Técnica para distribuir las solicitudes entre múltiples instancias de un servicio para mejorar el rendimiento y la disponibilidad de un sistema.

Certbot: Herramienta que automatiza la creación e instalación de certificados SSL/TLS de Let's Encrypt para asegurar que las conexiones al sistema sean seguras.

SSL: (Secure Sockets Layer) protocolo criptográfico que proporciona seguridad en la comunicación a través de la red. En este caso, asegura las conexiones HTTPS.

Let's Encrypt: Autoridad certificadora gratuita que proporciona certificados SSL para asegurar los sitios web, permitiendo conexiones HTTPS.

Dominio: Nombre que identifica una dirección en Internet. Se utiliza para hacer accesible una aplicación desde una URL personalizada.

Subdominio: Una subdivisión del dominio principal, que se utiliza para alojar diferentes servicios o secciones de una misma aplicación en URLs separadas.

API: Application Programming Interface (Interfaz de Programación de Aplicaciones)

HTTPS: Hypertext Transfer Protocol Secure (Protocolo de Transferencia de Hipertexto Seguro)

SQL: Structured Query Language (Lenguaje de Consulta Estructurado)

CSS: Cascading Style Sheets, lenguaje de hoja de estilos en cascada utilizado para dar formato a páginas web.

JS: JavaScript, lenguaje de programación utilizado para agregar funcionalidad a páginas web.

CRUD: Create, Read, Update, Delete (Crear, Leer, Actualizar, Eliminar), las cuatro operaciones básicas para manejar datos dentro de una aplicación.

5.1.5. Roles

El éxito de un proyecto en la metodología Scrum depende en gran medida de la correcta definición de roles, lo que asegura la asignación eficiente de tareas y responsabilidades a cada miembro del equipo:

Tabla 2. Roles de la metodología Scrum.

Rol	Característica	Especialistas
Product Owner	Es la persona que conoce sobre la idea del desarrollo del sistema On-line para la gestión de ventas y es responsable de definir y priorizar los requisitos del proyecto.	Aldas Aldaz Juan Carlos.
Scrum Master	Es la persona que verifica que el trabajo se esté desarrollando bajo las reglas y	Aldas Aldaz Juan Carlos.

	procesos detallados con la metodología Scrum.	
Scrum Team	Es el equipo que se encarga de desarrollar e implementar el sistema On-line para la gestión de ventas e inventario mediante la utilización de sprints.	García Arteaga Ángelo David, Aldas Aldaz Juan Carlos.
Stakeholders	Son las personas que están interesadas de forma directa o indirecta del proyecto.	Wilman Remigio Gomez, Valeria Aguilar.

5.1.6. Características de los usuarios

Se identificaron cuatro tipos de usuario, el usuario administrador del sistema, propietario, vendedor y el usuario contador.

Tabla 3. Características de los usuarios del sistema.

Tipo de Usuario	Actividades
Administrador del sistema	• Configurar el sistema (parámetros generales, ajustes de seguridad, preferencias de usuario) • Realizar copias de seguridad del sistema y los datos para garantizar la recuperación en caso de fallo. • Realizar revisiones de seguridad para identificar vulnerabilidades y riesgos.

Propietario	• Gestionar usuarios (agregar, editar, eliminar, listar, consultar detalles, asignar roles) • Gestionar ventas (listar ventas, consultar detalles, generar reportes) • Gestionar inventario (agregar, editar, eliminar, listar, consultar detalles, generar reportes) • Gestionar clientes (agregar, editar, eliminar, listar, consultar detalles, generar reportes) • Gestionar proveedores (agregar, editar, eliminar, listar, consultar detalles, generar reportes) • Personalizar y exportar reportes (ventas, inventario, clientes, proveedores) • Ver alertas de stock mínimo, alertas de resúmenes de ventas, alertas de procesos realizados.
Vendedor	• Registrar ventas utilizando el escáner de código de barras • Consultar inventario de productos • Consultar detalles de las ventas • Listar ventas realizadas
Contador	• Consulta de ventas realizadas para facturación • Consulta de ventas facturadas y no facturadas

5.1.7. Requisitos de usuario

3.1.7.7. Historias de usuario

Tabla 4. Historia de usuario de inicio de sesión.

Historia de Usuario	
Número: 1	**Usuario:** Propietario
Nombre de la Historia: Inicio de Sesión	
Prioridad: Alta	**Riesgo en Desarrollo:** Baja
Puntos estimados: 1 día	**Iteración Asignada:** 1
Programador Responsable: Juan Aldas, Angelo García	
Descripción: El sistema permite el ingreso de los usuarios mediante la petición de los datos: número de cedula y contraseña.	
Observaciones: Para el inicio de sesión el usuario debe estar registrado en el sistema por el usuario propietario.	

Tabla 5. Historia de usuario creación del usuario administrador.

Historia de Usuario	
Número: 2	**Usuario:** Administrador
Nombre de la Historia: Usuario administrador	
Prioridad: Alta	**Riesgo en Desarrollo:** Baja
Puntos estimados: 1 día	**Iteración Asignada:** 1
Programador Responsable: Juan Aldas, Angelo García	
Descripción: El sistema requiere un usuario administrador que se encargue de realizar tareas como configuración del sistema y la realización de las copias de seguridad para la recuperación.	
Observaciones:	

Tabla 6. Historia de usuario de registro y gestión de usuarios.

Historia de Usuario

Número: 3	**Usuario:** Propietario
Nombre de la Historia: Registro y gestión de usuarios	
Prioridad: Alta **Puntos estimados:** 1 día	**Riesgo en Desarrollo:** Media **Iteración Asignada:** 1
Programador Responsable: Juan Aldas, Angelo García	
Descripción: Se requiere crear nuevos usuarios y gestionar usuarios para los empleados del local, con información como nombres, email, contraseña, teléfono, residencia, tipo de empleo, cedula, cargo o rol, palabra segura.	
Observaciones: La gestión de los usuarios incluye agregar, editar y dar de baja a usuarios.	

Tabla 7. Historia de usuario de registro de productos en inventario.

Historia de Usuario	
Número: 4	**Usuario:** Propietario
Nombre de la Historia: Registro de Productos	
Prioridad: Alta **Puntos estimados:** 2 días	**Riesgo en Desarrollo:** Media **Iteración Asignada:** 1
Programador Responsable: Juan Aldas, Angelo García	
Descripción: El sistema permite registrar y gestionar productos, incluyendo información como nombre, estado, código primario, código secundario, código de barras, cantidad, descripción, precio, descuento.	
Observaciones: La gestión de los productos incluye agregar, editar y la baja productos.	

Tabla 8. Historia de usuario de consulta de inventario de productos.

Historia de Usuario	
Número: 5	**Usuario:** Vendedor
Nombre de la Historia: Consulta de inventario de productos	
Prioridad: Alta **Puntos estimados:** 2 días	**Riesgo en Desarrollo:** Alta **Iteración Asignada:** 1

Programador Responsable: Juan Aldas, Angelo García
Descripción: El sistema permite consultar el inventario de productos para que el vendedor pueda verificar la disponibilidad y realizar las ventas.
Observaciones: Para la consulta los productos deben ser registrados previamente y poder consultarse a través del código de barras o el nombre.

Tabla 9. Historia de usuario de registro de ventas.

Historia de Usuario	
Número: 6	**Usuario:** Vendedor
Nombre de la Historia: Registro de Ventas	
Prioridad: Alta	**Riesgo en Desarrollo:** Alta
Puntos estimados: 2 días	**Iteración Asignada:** 1
Programador Responsable: Juan Aldas, Angelo García	
Descripción: El sistema permite registrar y gestionar ventas, incluyendo información como fecha, hora, producto vendido, cantidad, precio, etc.	
Observaciones: Durante la venta debe permitir consultar el detalle de la venta y la modificación de parámetros como la cantidad de los productos y otros. Además de la agregación de productos a través del lector de código de barras	

Tabla 10. Historia de usuario de registro de clientes.

Historia de Usuario	
Número: 7	**Usuario:** Propietario, vendedor
Nombre de la Historia: Registro de Clientes	
Prioridad: Alta	**Riesgo en Desarrollo:** Media
Puntos estimados: 2 días	**Iteración Asignada:** 1
Programador Responsable: Juan Aldas, Angelo García	
Descripción: El sistema permite registrar y gestionar clientes, incluyendo información como nombre, apellido, email, dirección, teléfono.	
Observaciones: La gestión de los clientes incluye agregar, editar y eliminar los clientes.	

Tabla 11. Historia de usuario de gestión de usuario.

Historia de Usuario	
Número: 8	**Usuario:** Propietario
Nombre de la Historia: Gestión de Inventario	
Prioridad: Alta	**Riesgo en Desarrollo:** Media
Puntos estimados: 2 días	**Iteración Asignada:** 1
Programador Responsable: Juan Aldas, Angelo García	
Descripción: El sistema permite gestionar el inventario de productos, generar reportes y ver información como cantidad en stock, etc.	
Observaciones: La gestión de productos incluye la edición y la baja de estos.	

Tabla 12. Historia de usuario de generación de reportes.

Historia de Usuario	
Número: 9	**Usuario:** Propietario
Nombre de la Historia: Generación de Reportes	
Prioridad: Alta	**Riesgo en Desarrollo:** Alta
Puntos estimados: 3 días	**Iteración Asignada:** 2
Programador Responsable: Juan Aldas, Angelo García	
Descripción: El sistema permite generar reportes y estadísticas sobre ventas, productos, clientes y proveedores.	
Observaciones:	

Tabla 13. Historia de usuario de búsqueda y consulta.

Historia de Usuario	
Número: 10	**Usuario:** Vendedor, Propietario
Nombre de la Historia: Búsqueda y Consulta	
Prioridad: Media	**Riesgo en Desarrollo:** Baja
Puntos estimados: 1 días	**Iteración Asignada:** 2
Programador Responsable: Juan Aldas, Angelo García	
Descripción: El sistema permite buscar y consultar ventas, productos, clientes y proveedores por diferentes filtros.	

Observaciones:

Tabla 14. Historia de usuario de análisis de ventas.

Historia de Usuario	
Número: 11	**Usuario:** Propietario
Nombre de la Historia: Análisis de Ventas	
Prioridad: Media	**Riesgo en Desarrollo:** Media
Puntos estimados: 1 días	**Iteración Asignada:** 2
Programador Responsable: Juan Aldas, Angelo García	
Descripción: El sistema permite generar estadísticas de productos más vendidos y gráficos de barras sobre las ventas mensuales, para las tendencias del local comercial.	
Observaciones:	

Tabla 15.Historia de usuario de consulta de ventas realizadas, facturadas y no facturadas.

Historia de Usuario	
Número: 12	**Usuario:** Contador
Nombre de la Historia: Consulta de ventas realizadas, facturas y no facturadas	
Prioridad: Media	**Riesgo en Desarrollo:** Media
Puntos estimados: 1 días	**Iteración Asignada:** 2
Programador Responsable: Juan Aldas, Angelo García	
Descripción: El sistema permite la visualización y consulta de las ventas facturadas y no facturadas.	
Observaciones:	

Tabla 16. Historia de usuario de visualización de alertas.

Historia de Usuario	
Número: 13	**Usuario:** Propietario
Nombre de la Historia: Visualización de alertas	

Prioridad: Media	**Riesgo en Desarrollo:** Media
Puntos estimados: 1 días	**Iteración Asignada:** 2
Programador Responsable: Juan Aldas, Angelo García	
Descripción: El propietario debe visualizar alertas de stock mínimo, o medio para prevenir la falta de los productos, alertas de resúmenes de ventas, alertas de los procesos realizados en el sistema como el registro de un cliente, registro de inventario, registro de proveedores, entre otros.	
Observaciones:	

3.1.7.8. Requisitos no funcionales

En el contexto de la implementación de un sistema web para la gestión de ventas en el local comercial Isabelita de la ciudad de Guaranda, los requerimientos no funcionales son fundamentales para determinar la eficiencia y calidad del sistema. Los requerimientos funcionales por considerar son los siguientes:

- RNF1: El sistema debe ser seguro y proteger la información de los clientes y el local comercial contra acceso no autorizado.
- RNF2: Debe ser escalable y capaz de manejar un aumento en la cantidad de usuarios y transacciones.
- RNF3: El sistema debe ser compatible con diferentes dispositivos y navegadores para permitir el acceso desde diferentes plataformas.
- RNF4: Las funcionalidades del sistema debe ser fácil de usar e intuitivo para los usuarios del local, minimizando la necesidad de capacitación.
- RNF5: La aplicación web debe ser confiable y disponible durante las 24 horas del día, con un tiempo de respuesta promedio de 2 segundos.
- RNF7: El sistema debe cumplir con los estándares de documentación y registro del local.
- RNF8: El sistema debe proporcionar un mecanismo de respuesta a errores y excepciones para minimizar la pérdida de datos y tiempo.
- RNF9: Las funcionalidades del sistema deben ser testeadas y validadas para asegurar que cumpla con los requisitos funcionales y no funcionales.

3.1.7.9. Requisitos de interfaz

5.1.7.1.1. Interfaz de Hardware

Los requisitos de hardware necesarios para la funcionalidad del sistema son:

- Dispositivos de entrada
  - Teclado
  - Mouse
- Dispositivos de salida
  - Monitor (Pantalla)
  - Internet
  - Impresora
- Otros requisitos de hardware
  - Procesador: Intel Core 2 o superior
  - Memoria RAM: 2 GB o superior
  - Disco duro: 64 GB o superior

5.1.7.1.2. Interfaz de Software

El sistema web debe interactuar con un cliente web como:

- Google Chrome versión 61+
- Microsoft Edge versión 16+
- Mozilla Firefox versión 60+
- Safari on iOS versión 10.3+
- Otros navegadores web compatibles con **HTML5 y CSS3**

5.1.7.1.3. Interfaz de Comunicación

El sistema web debe utilizar protocolos de comunicación seguros y estándar, como:

- HTTPS (Hypertext Transfer Protocol Secure)
- JSON (JavaScript Object Notation) para el intercambio de datos entre el cliente y el servidor
- API RESTful (Representational State of Resource) para la comunicación entre el sistema web y otros sistemas

5.2. Planificación y estimación

5.2.1. Product Backlog

Para documentar cada una de las tareas, se utilizarán los siguientes campos:

- Descripción: una breve explicación del requisito.
- Complejidad o estimación del esfuerzo requerido: una evaluación del nivel de dificultad y del tiempo necesario para completar la tarea.
- Prioridad: la importancia relativa del requisito en comparación con otros.
- Número de sprint: la iteración en la que se abordará la tarea.
- Duración: el tiempo estimado para completar la tarea.

Es importante destacar que la duración de cada sprint puede variar según la complejidad de la tarea y la cantidad de procesos involucrados. El product backlog se crea a partir de las historias de usuario, que se agrupan en categorías para presentar de manera más clara y organizada.

Tabla 17. Product Backlog.

Descripción	**Prioridad**	**Complejidad**	**Sprint**	**Duración (días)**
Diseño de alto nivel.	4	4	1	6
Diseño detallado.	5	5	2	6
Programación Backend.	5	5	3	30
Programación Frontend.	5	5	4	35
Análisis y planificación de pruebas.	4	4	5	5
Ejecución de las pruebas.	4	4	6	14
Total				**96**

La prioridad y la complejidad se evaluaron utilizando una escala de 1 a 5, donde:

- La prioridad se clasificó de la siguiente manera: 1 (menos urgente) a 5 (más urgente), lo que indica la importancia relativa de cada tarea.
- La complejidad se valoró según su nivel de dificultad, con 1 (menor complejidad) y 5 (mayor complejidad), lo que refleja el esfuerzo y los recursos necesarios para completar cada requisito.

5.2.2. Estimación del Product Backlog

Evaluar las tareas del product backlog es esencial para comprender claramente el esfuerzo y el tiempo necesarios para completar cada una. Esto facilita una planificación precisa y una ejecución eficiente en las fases posteriores del proyecto.

Tabla 18. Estimación del Product Backlog.

Id	Tareas	Tiempo estimado (días)	Estimación en caso de retrasos (días)	Sprint
1	Diseño de alto nivel.	6	1	1
2	Diseño detallado.	6	1	2
3	Programación Backend.	30	5	3
4	Programación Frontend.	35	5	4
5	Análisis y planificación de pruebas.	5	1	5
6	Ejecución de las pruebas.	14	2	6
	Total	**96**	**15**	

Con una estimación de 96 días y un tiempo de 15 días adicionales en caso de retraso, se completaron las tareas del product backlog con un total de 111 días.

5.3. Implementación

5.3.1. Desarrollo del sprint 1. Diseño de alto nivel

3.1.7.10. Planificación del sprint 1

En el desarrollo del sprint 1 se realizó la construcción de la arquitectura del proyecto para la implementación del sistema On-line para la gestión de ventas en el local Isabelita. El sprint duró 6 días, con un promedio de 5 horas de trabajo diarias.

El objetivo de este sprint fue definir la arquitectura de la aplicación web, el diseño de la navegación, el diseño de la base de datos, diseñar la vista principal de la aplicación y configurar el marco de trabajo.

Para el diseño de la base de datos se tomó en cuenta la información proporcionada por el propietario durante la entrevista realizada, la observación directa y el análisis de documentos existentes como hojas electrónicas de Excel y registros en hojas de papel. Además del análisis de las historias de usuario para identificar las entidades y sus relaciones.

Durante este sprint, se diseñó el modelo de entidad-relación, que describe las entidades, sus propiedades y relaciones. También se definió la estructura de la base de datos, incluyendo entidades para la gestión de ventas, clientes, productos, proveedores, entre otros.

Una vez finalizado el diseño de la base de datos, se realizó una revisión con la colaboración del *Scrum Master*. La revisión permitió identificar posibles incoherencias y errores en el diseño de la base, y se realizaron las correcciones necesarias.

3.1.7.11. Tareas del sprint 1

Tabla 19. Tareas del sprint 1 - Diseño de alto nivel.

ID	Rol	Tareas	Tiempo de estimación (horas)
1	Desarrollador.	Definir la arquitectura del sistema.	**2**
2	Desarrollador.	Diseñar el modelo entidad relación de la base de datos.	**24**
3	Desarrollador, Scrum Master.	Revisar el modelo entidad relación.	**4**
4	Desarrollador.	Configurar el SGBD.	**1**
5	Desarrollador.	Crear la base de datos.	**4**
6	Desarrollador	Realizar pruebas de funcionamiento de la base de datos.	**3**

7	Desarrollador	Diseñar la navegación del sistema On-line.	**6**
8	Desarrollador	Diseñar la estructura de la vista principal del sistema.	**2**
		Total	**46**

3.1.7.12.Ejecución del sprint 1 – Scrum diario

Tabla 20. Scrum diario del sprint 1 - Diseño de alto nivel.

Día	Miembro del Equipo	Tarea Realizada	Impedimentos	Acciones a Seguir
1	Juan Aldaz, Angelo García	Se realizo la entrevista con el propietario del local para determinar los requerimientos del sistema y para establecer las bases del proyecto y el diseño funcional del sistema.	Es necesario más información.	Continuar con la obtención de información, a través de la observación directa y el análisis de documentos existentes.
1	Juan Aldaz, Angelo García	Se definió la arquitectura del sistema con la que se va a trabajar.		Realizar una reunión con el equipo para socializar la arquitectura del sistema y las actividades que se van a realizar.

1	Juan Aldaz, Angelo García	Se diseño el modelo entidad-relación de la base de datos.	Necesidad de aclaración de requisitos.	Reunión con el cliente para aclarar dudas.
1	Angelo García	Se configuró el SGBD.		Verificar configuraciones y preparar la base de datos.
2	Juan Aldaz, Angelo García	Se finalizó la arquitectura del sistema.		Documentar la arquitectura definida.
2	Juan Aldaz	Se revisó el modelo entidad-relación.		Esperar feedback y realizar ajustes.
2	Juan Aldaz, Angelo García	Se creo el modelo físico la base de datos.	Retrasos en la configuración del SGBD.	Configurar correctamente el SGBD y crear la base de datos.
3	Juan Aldaz	Se documentó del diseño arquitectónico.		Iniciar la documentación técnica.
3	Juan Aldaz	Se ajustó el modelo entidad-relación.		Implementar ajustes según feedback.
3	Angelo García	Se continuó con la creación de la base de datos.		Finalizar la estructura y empezar pruebas con la base de datos.
4	Juan Aldaz	Se diseñó la navegación de la		Continuar con el diseño de navegación.

		del sistema On-line.		
4	Angelo García	Se realizó pruebas de funcionamiento de la base de datos.	Problemas menores detectados.	Corregir los problemas y retomar las pruebas.
5	Angelo García	Se optimizó la base de datos.		Finalizar optimizaciones y preparar la base de datos para su uso final.
5	Angelo García	Se diseñó la estructura de la vista principal del sistema.		Finalizar el diseño de la vista principal.
6	Juan Aldaz, Angelo García	Se realizó una revisión final del diseño de navegación y estructura principal.		Realizar las correcciones del diseño de la navegación.
6	Juan Aldaz, Angelo García	Se documentó la base de datos.		Revisar la documentación y finalizar el sprint.

3.1.7.13.Resultados

Los resultados del *sprint 1* incluyen la arquitectura de la aplicación web, el diseño de la navegación, el modelo entidad – relación de la base de datos, el diseño estructural de la vista principal de la aplicación.

5.3.1.1.1. Arquitectura del software

Patrón MVC

Para la implementación del sistema web de gestión de ventas para el local comercial ISABELITA, se ha optado por utilizar la arquitectura de software basada en el patrón MVC (Modelo-Vista-Controlador).

Esta elección permite una clara separación de las responsabilidades, separando la lógica de la aplicación y la lógica de la presentación, facilitando el desarrollo.

- **Modelo (Model)**

  El componente Modelo es responsable de la lógica de negocio y la gestión de datos del sistema. En este caso, el modelo interactúa directamente con la base de datos y contiene las definiciones de las entidades principales del sistema de gestión de ventas, tales como productos, clientes, ventas, proveedores y usuarios. Los principales elementos del Modelo son:

- **Vista (View)**

  El componente Vista se encarga de la presentación de los datos al usuario y la interacción con él. Las vistas son las interfaces de usuario que muestran la información y reciben las entradas del usuario. En este sistema web, las vistas están compuestas por páginas web realizadas con la librería ReactJS que es una de las librerías más populares JavaScript, que utiliza HTML y CSS para la presentación dinámica de los datos.

- **Controlador (Controller)**

  El componente Controlador actúa como intermediario entre el Modelo y la Vista. Recibe las solicitudes del usuario a través de la Vista, procesa estas solicitudes mediante la lógica del Modelo, y devuelve los resultados a la Vista para ser presentados. Los controladores contienen la lógica de aplicación y control de flujo.

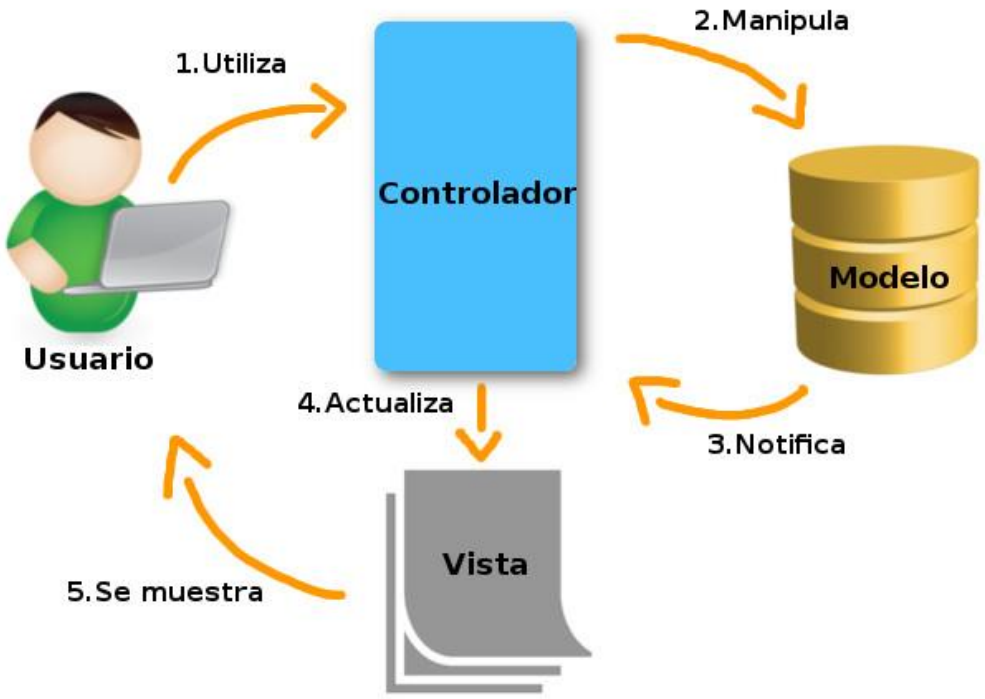


Gráfico 2. Funcionamiento de la aplicación según el modelo MVC. Fuente: Mikaelson (2016)

Esta arquitectura basada en MVC proporcionará una estructura sólida y flexible para el desarrollo del sistema, permitiendo un desarrollo ágil y eficiente, además de facilitar el mantenimiento y futuras expansiones del sistema.

5.3.1.1.2. Diseño de la navegación del sistema

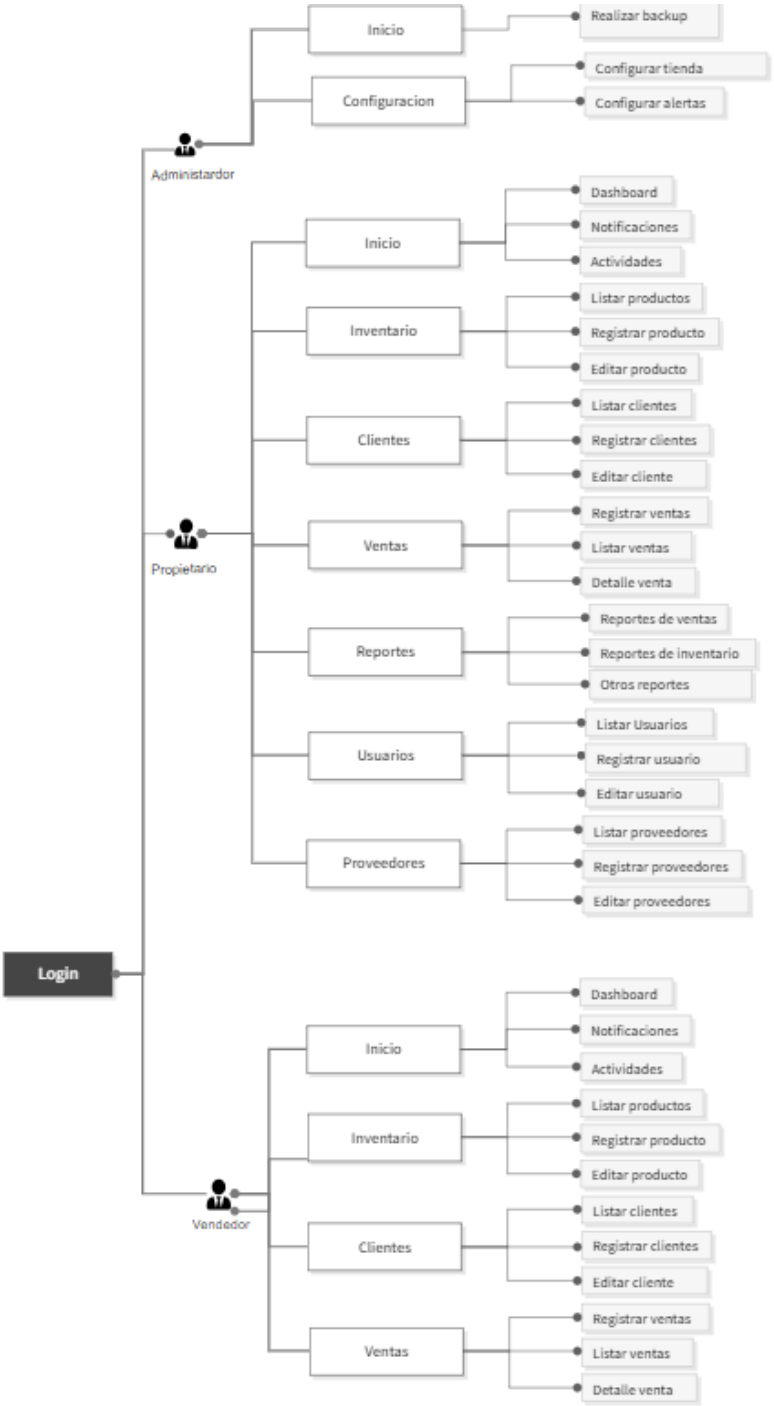


Gráfico 3. Diseño de la navegación web del sistema.

5.3.1.1.3. Modelo entidad relación de la base de datos

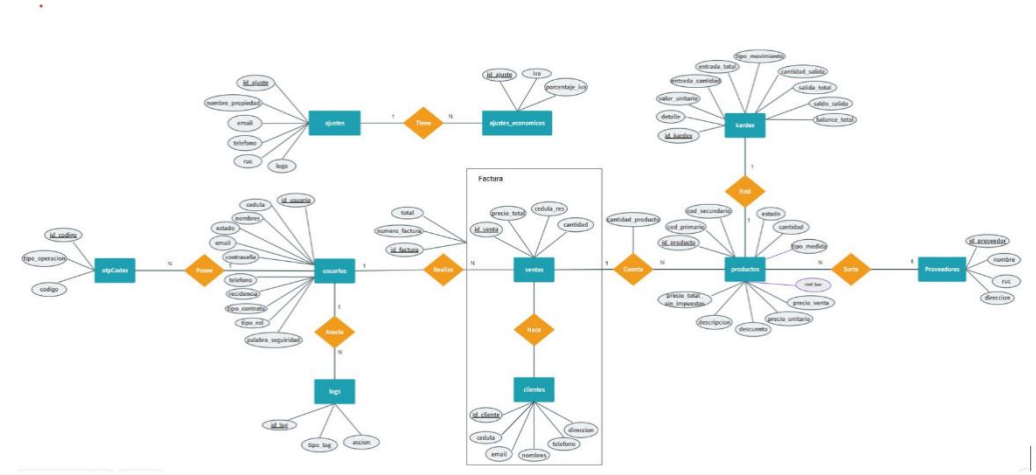


Gráfico 4. Diagrama Entidad - Relación Lógico.

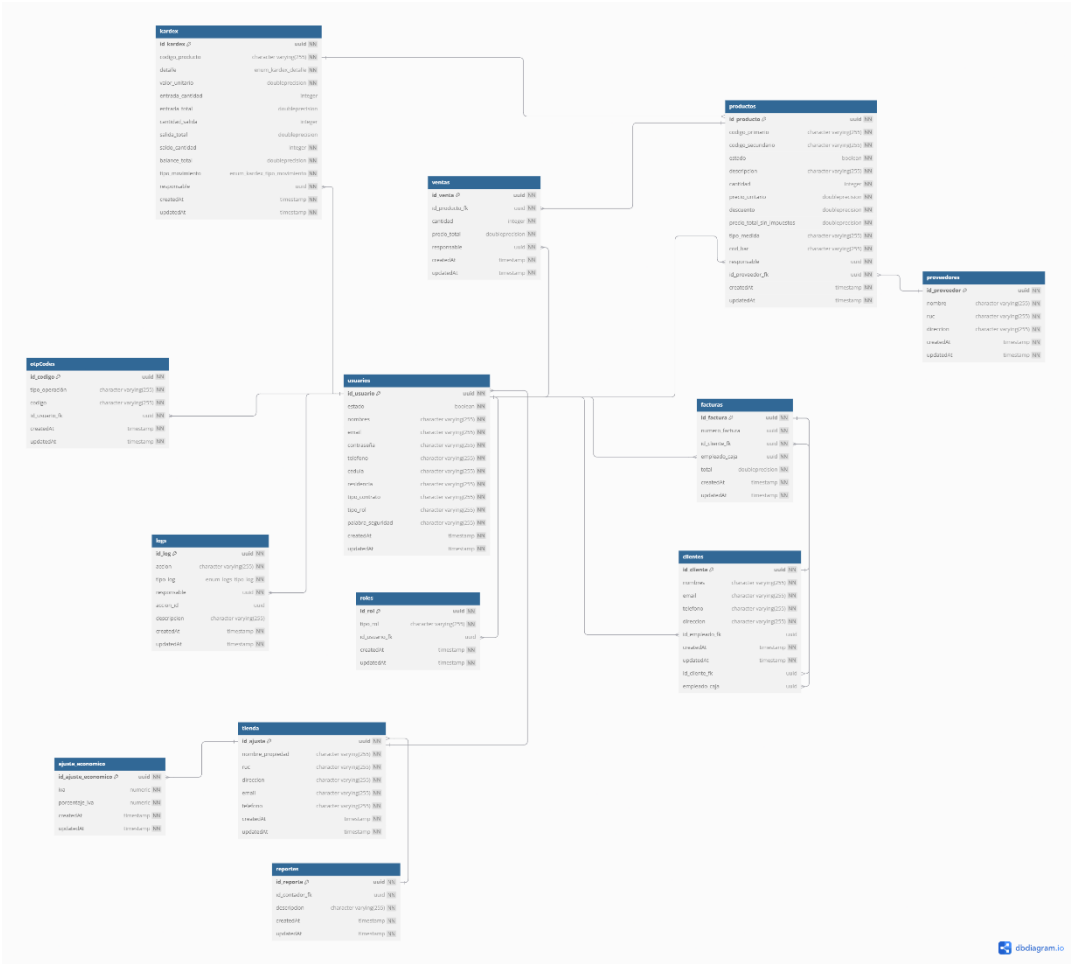


Gráfico 5. Modelo Entidad - relación de la base de datos.

5.3.2. Desarrollo del sprint 2. Diseño detallado

3.1.7.14. Planificación del sprint 2

En el sprint 2 se completaron las actividades esenciales para el diseño detallado del sistema. Este sprint se enfocó en la creación de diagramas UML y el diseño de la interfaz de usuario (UI) utilizando la arquitectura MVC (Modelo-Vista-Controlador).

Los diagramas UML elaborados incluyen el diagrama de casos de uso, que identifica y describe las funcionalidades principales del sistema desde la perspectiva del usuario y el diagrama de secuencia, que muestra la interacción entre los objetos del sistema para realizar casos de uso específicos.

Además, se diseñaron wireframes y prototipos para las vistas principales del sistema, lo que permitió visualizar el flujo de la aplicación y asegurar una experiencia de usuario coherente y efectiva. La guía de estilo fue definida para garantizar la consistencia visual en todo el sistema, abarcando la selección de paleta de colores, tipografía y componentes UI.

Se realizaron sesiones de validación y revisión con las partes interesadas, recopilando y documentando su retroalimentación. Los ajustes necesarios fueron implementados para alinear el diseño con los requisitos y expectativas del propietario del local, asegurando así un diseño detallado y robusto listo para la fase de desarrollo.

3.1.7.15. Tareas del sprint 2

Tabla 21. Tareas del sprint 2 - Diseño detallado.

ID	Rol	Tareas	Tiempo de estimación (horas)
1	Desarrollador.	Identificar los actores principales.	2
2	Desarrollador.	Definir los casos de uso principales.	3
3	Desarrollador.	Crear el diagrama de casos de uso.	3
4	Desarrollador, Product Owner.	Seleccionar casos de uso clave.	2

5	Desarrollador.	Identificar los objetos involucrados en cada caso de uso.	3
6	Desarrollador.	Describir la secuencia de mensajes entre los objetos.	3
7	Desarrollador.	Crear los diagramas de secuencia.	3
8	Desarrollador, Scrum Master.	Identificar los componentes del Modelo, Vista y Controlador.	2
9	Desarrollador.	Definir las interacciones entre los componentes.	3
10	Desarrollador.	Diseñar wireframes para las vistas principales.	5
11	Desarrollador.	Seleccionar paleta de colores, tipografía y componentes UI.	3
12	Desarrollador.	Documentar las reglas de diseño y uso de componentes.	3
13	Product Owner.	Organizar una reunión de revisión del diseño con las partes interesadas.	2
14	Product Owner, Desarrollador.	Recopilar y documentar la retroalimentación.	2
15	Desarrollador, Product Owner.	Realizar ajustes en los diagramas y diseños de interfaz según sea necesario.	4
		Total	**43**

3.1.7.16.Ejecución del sprint 2 – Scrum diario

Tabla 22. Scrum diario del sprint 2 - Diseño detallado.

Día	Miembro del Equipo	Tarea Realizada	Impedimentos	Acciones a Seguir

1	Angelo García, Juan Aldas.	Se identificó los actores principales, definir sus roles y responsabilidades.	Continuar con la definición de casos de uso principales.
1	Juan Aldas.	Se creo el diagrama de casos de uso, incluyendo la descripción de cada caso de uso.	Revisar y refinar el diagrama con el Product Owner.
2	Juan Aldas.	Se definió los casos de uso principales, los actores involucrados.	Revisar y refinar los casos de uso con el Desarrollador.
3	Juan Aldas.	Se identificó los objetos involucrados en cada caso de uso, definir sus roles y responsabilidades	Continuar con la definición de las interacciones entre los componentes.
4	Angelo García.	Se describió la secuencia de mensajes entre los objetos, crear el diagrama de secuencia	Continuar con la creación de los diagramas de secuencia.
5	Angelo García.	Se definió los diagramas de secuencia, incluyendo la descripción de cada mensaje y su flujo.	Revisar y refinar los diagramas con el Scrum Master.
6	Juan Aldas.	Se definió las interacciones entre los componentes, e	Organizar una reunión de revisión del

		identificar los patrones de diseño.		diseño con las partes interesadas.
6	Angelo García.	Se creo los wireframes para las vistas principales, y se seleccionó la paleta de colores, tipografía y componentes UI.	Nuevos requisitos y cambios en el diseño de la interfaz del sistema web.	Continuar con la documentación de las reglas de diseño y uso de componentes.

3.1.7.17. Resultados

En este sprint se realizó el diseño detallado del sistema, que incluye los diagramas UML como diagramas de caso de uso, diagramas de secuencia y el diseño de la interfaz de usuario.

5.3.2.1.1. Diagrama de casos de uso

Diagrama de casos de uso de las funciones de Administrador

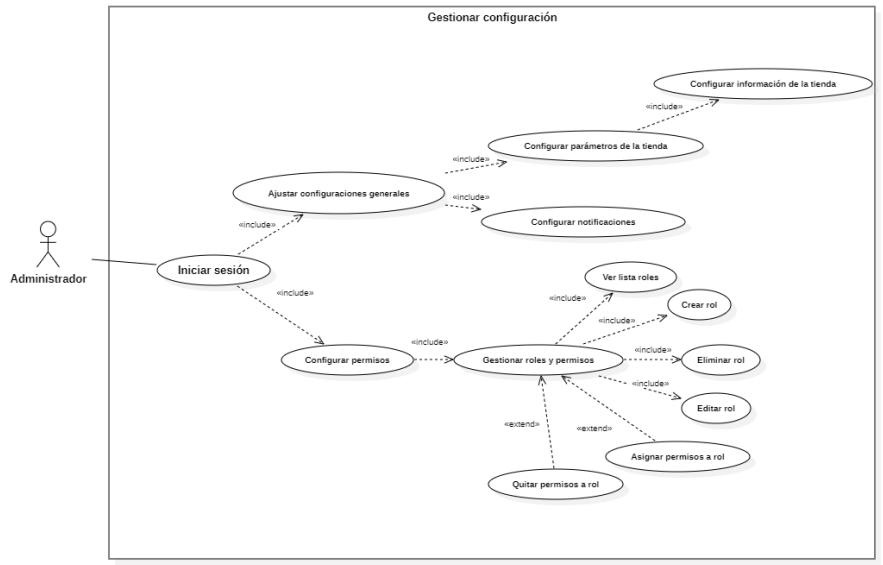


Gráfico 6. Caso de uso - Gestión de la configuración – Usuario Administrador.

Diagrama de casos de uso de las funciones del Propietario.

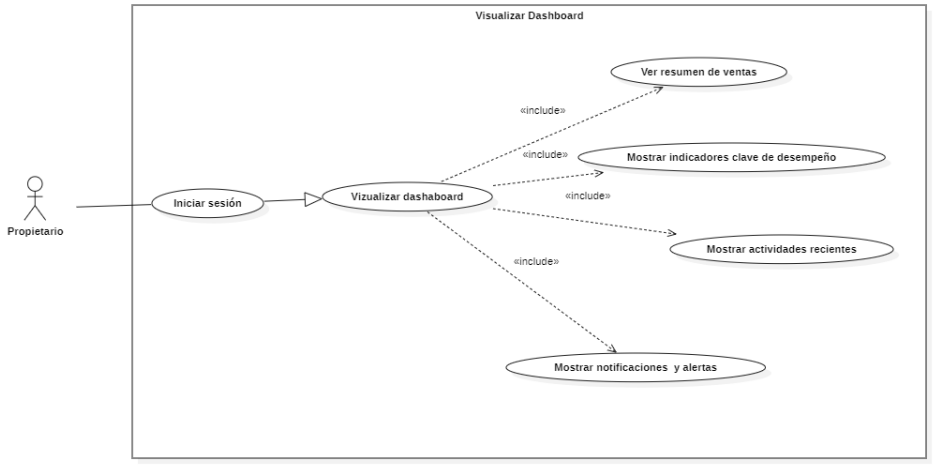


Gráfico 7. Caso de uso - Visualización del Dashboard – Usuario Propietario.

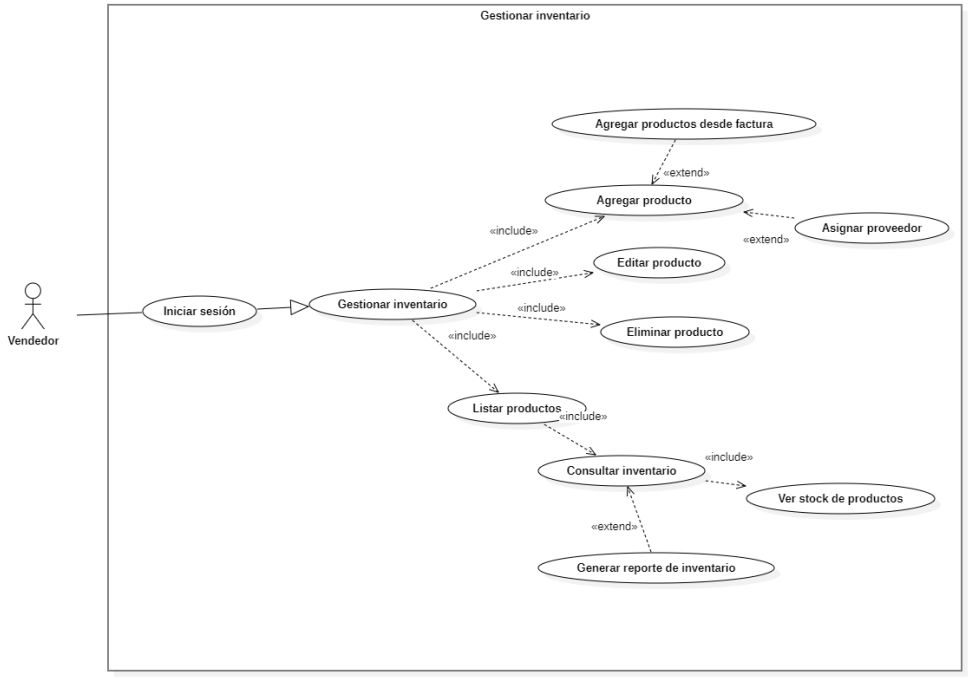


Gráfico 8. Caso de uso - Gestión de inventario – Usuario Propietario.

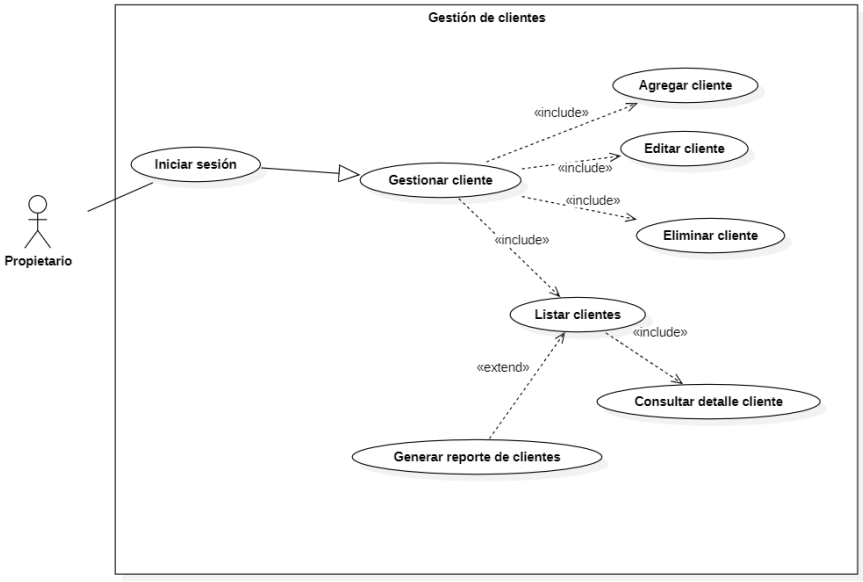


Gráfico 9. Caso de uso - Gestión de clientes – Usuario Propietario.

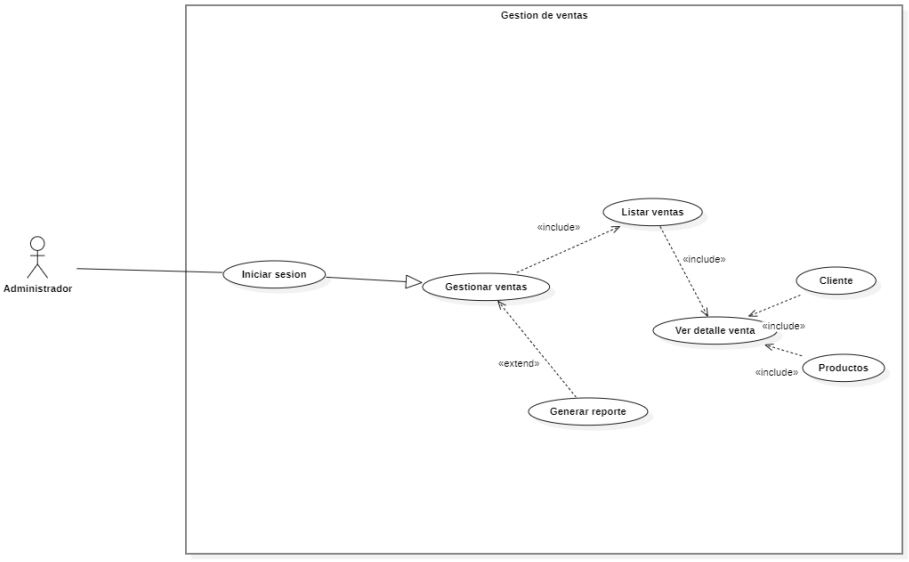


Gráfico 10. Caso de uso - Gestión de ventas – Usuario Propietario.

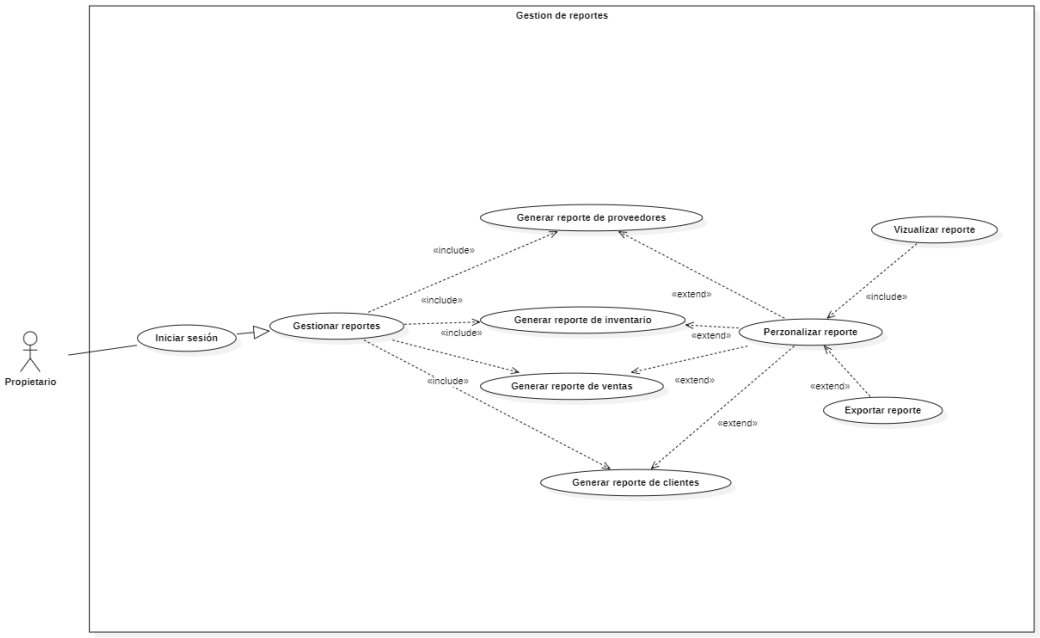


Gráfico 11. Caso de uso - Gestión de reportes – Usuario Propietario.

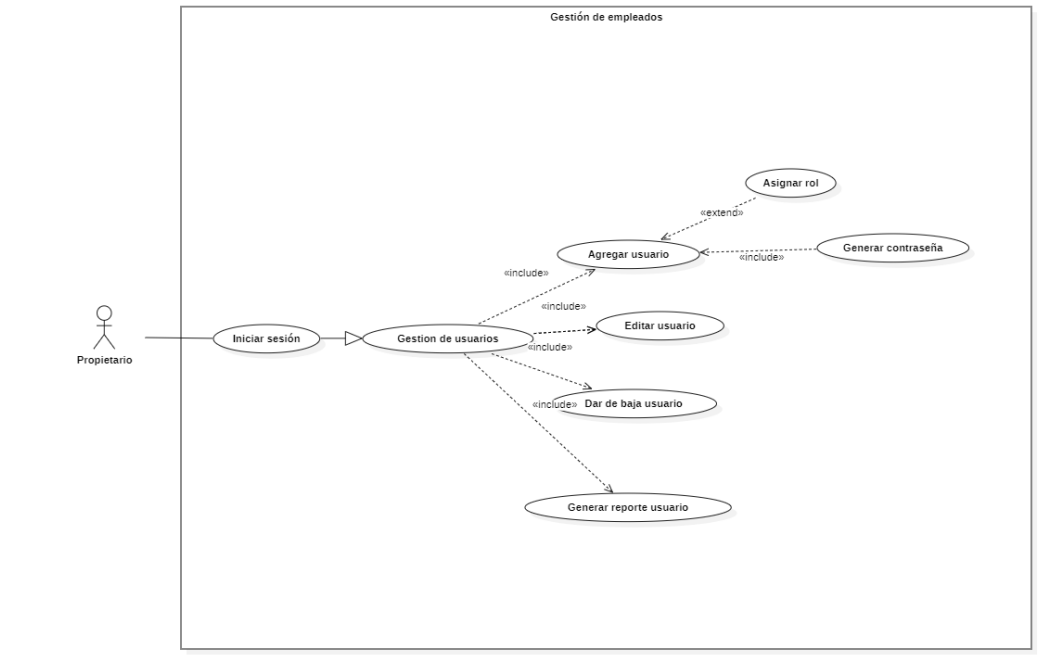


Gráfico 12. Caso de uso - Gestión de usuarios – Usuario Propietario.

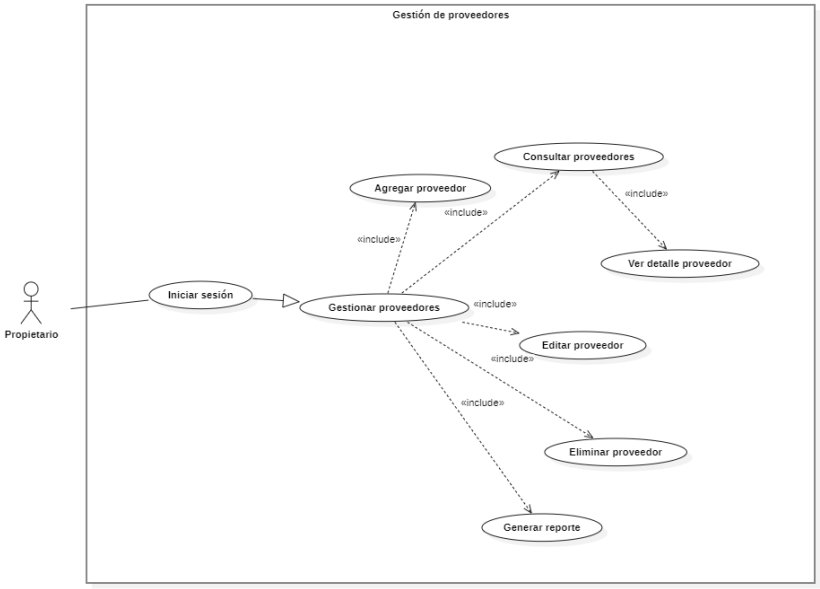


Gráfico 13. Caso de uso - Gestión de proveedores – Usuario Propietario.

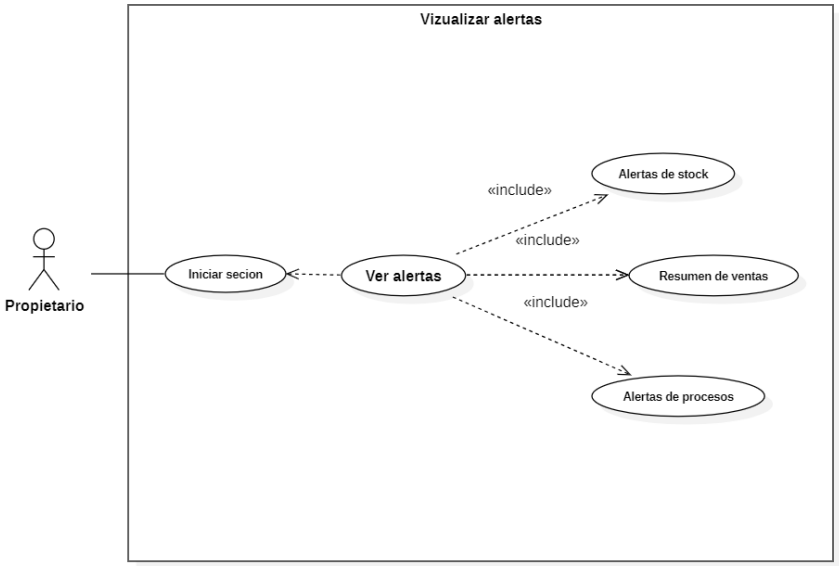


Gráfico 14. Caso de uso - Visualización de alertas - Usuario Propietario

Diagrama de casos de uso de las funciones del Contador

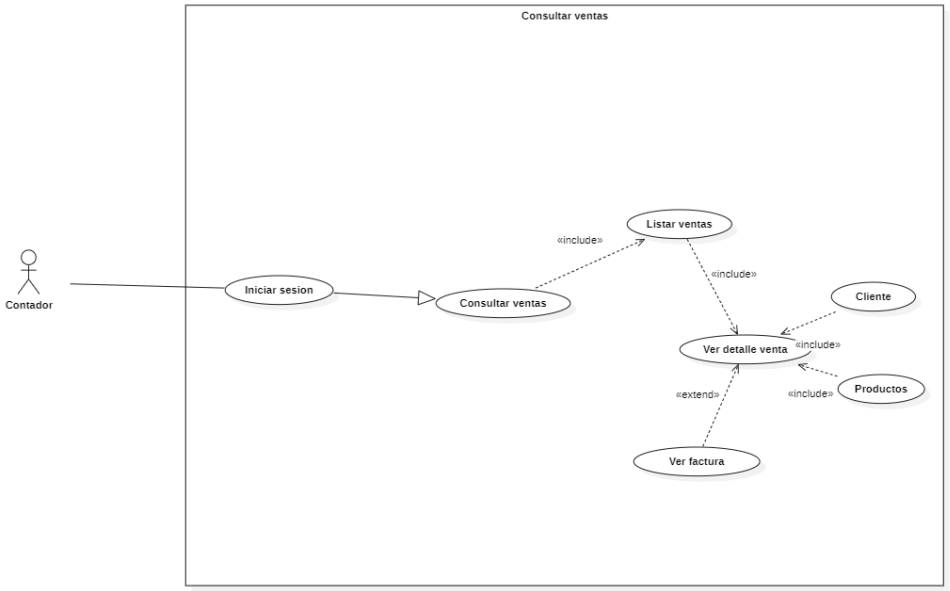


Gráfico 15. Caso de uso - Consulta de ventas realizas – Usuario Contador.

Diagrama de casos de uso de las funciones del Vendedor

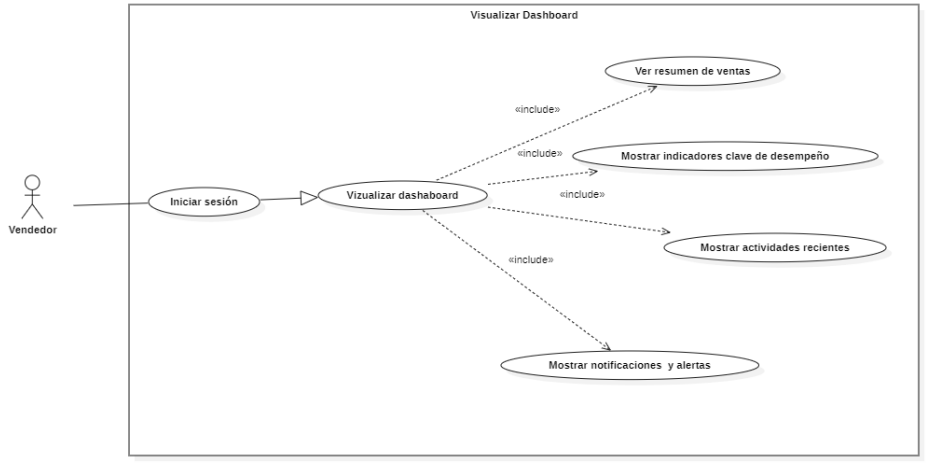


Gráfico 16. Caso de uso - Visualizar dashboard – Vendedor.

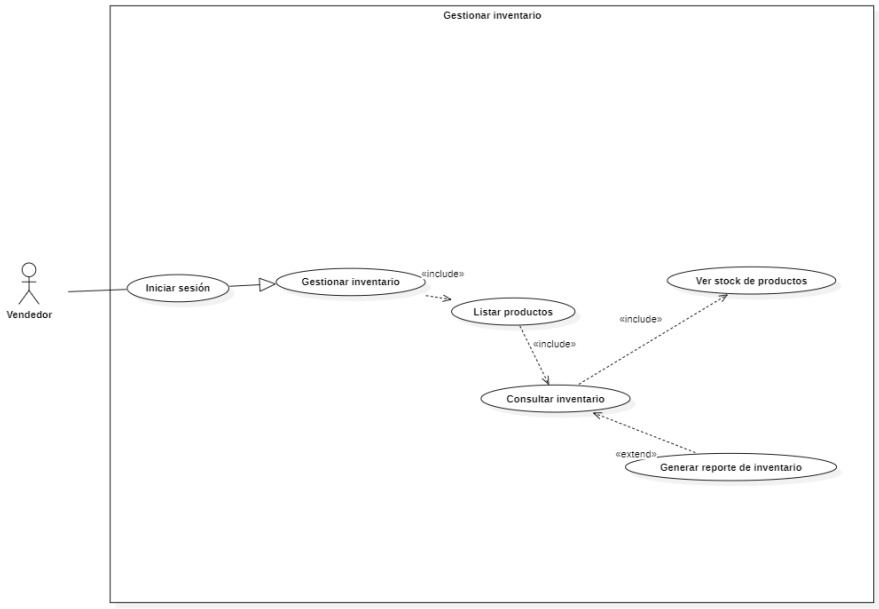


Gráfico 17. Caso de uso -Gestión de inventario - Vendedor

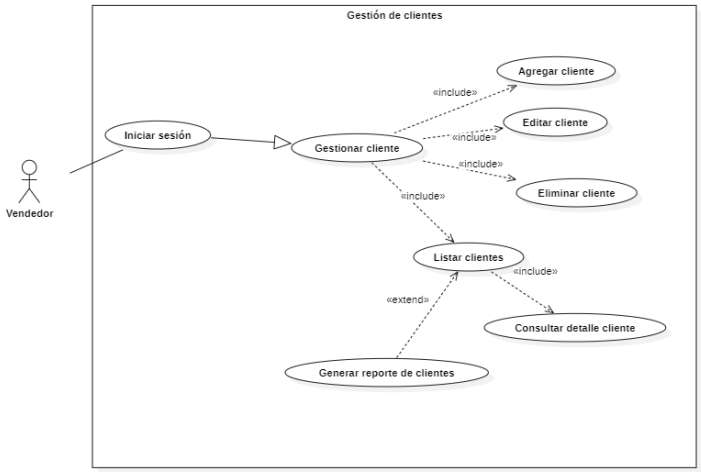


Gráfico 18. Caso de uso - Gestión de clientes - Vendedor.

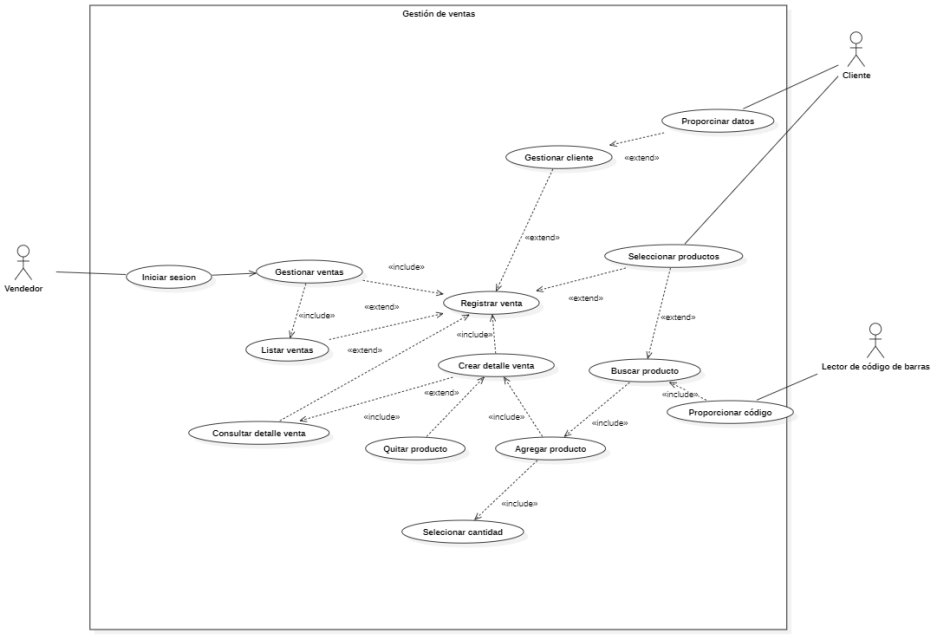


Gráfico 19. Caso de uso - Gestión de ventas - Usuario Vendedor.

5.3.2.1.2. Diagrama de secuencia

Diagrama de secuencia de inicio de sesión

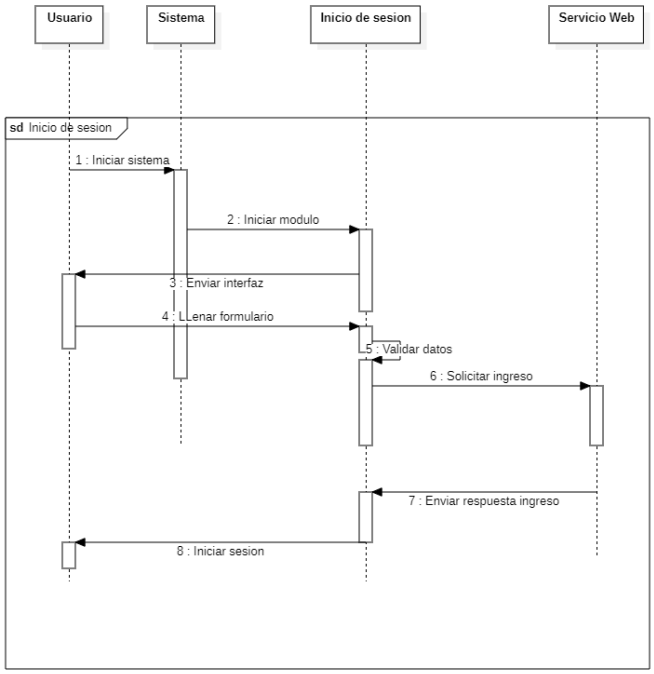


Gráfico 20. Diagrama de secuencia de inicio de sesión.

Diagramas de secuencia usuario Administrador

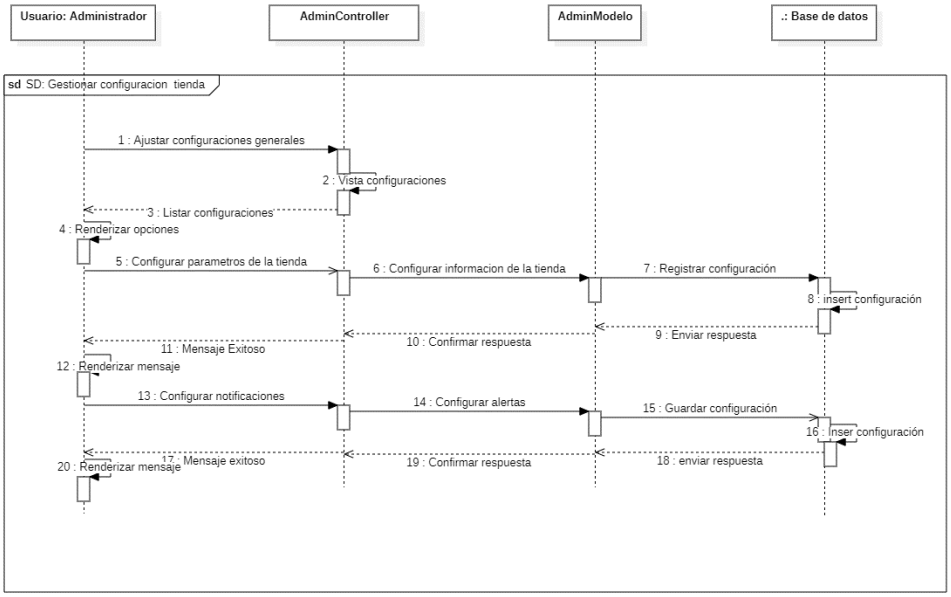


Gráfico 21. Diagrama de secuencia - Configuración del local comercial.

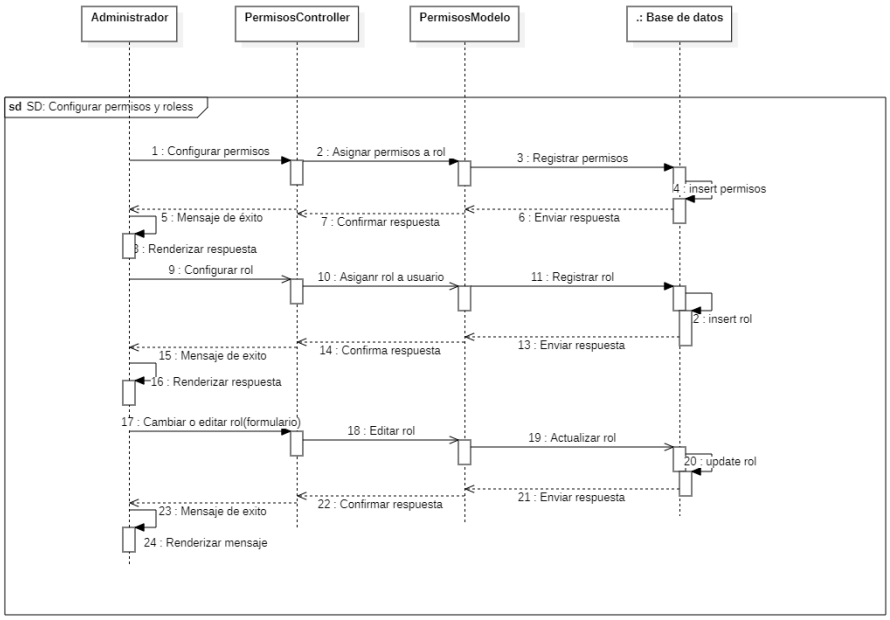


Gráfico 22. Diagrama de secuencia - Configuración de permisos y roles.

Diagramas de secuencia del usuario Propietario

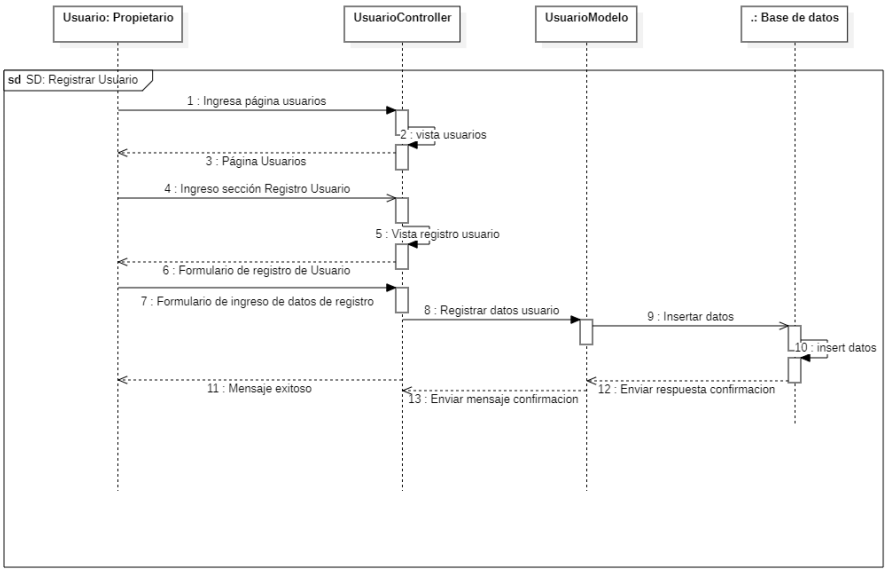


Gráfico 23. Diagrama de caso de uso - Registrar usuario

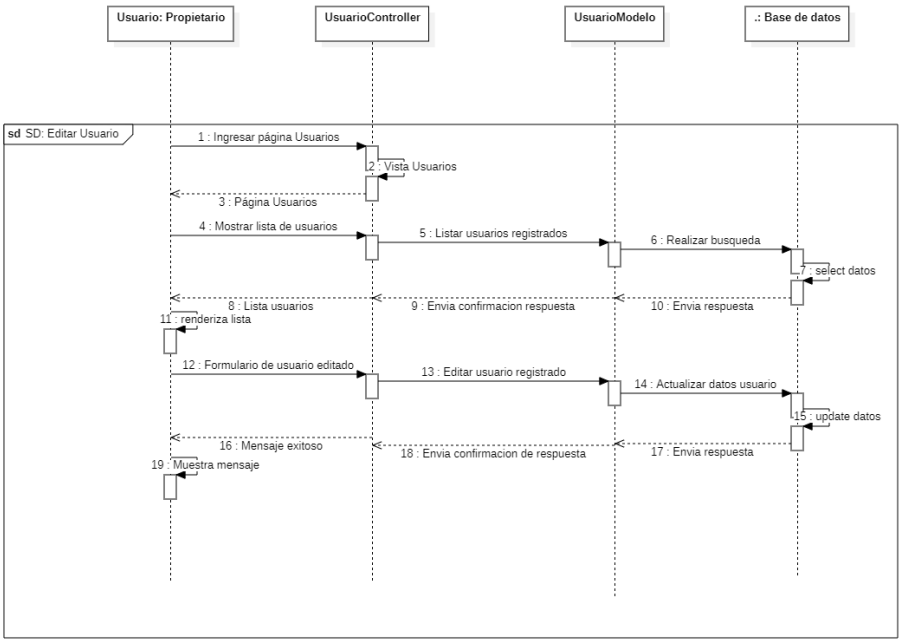


Gráfico 24. Diagrama de secuencia - Editar usuario.

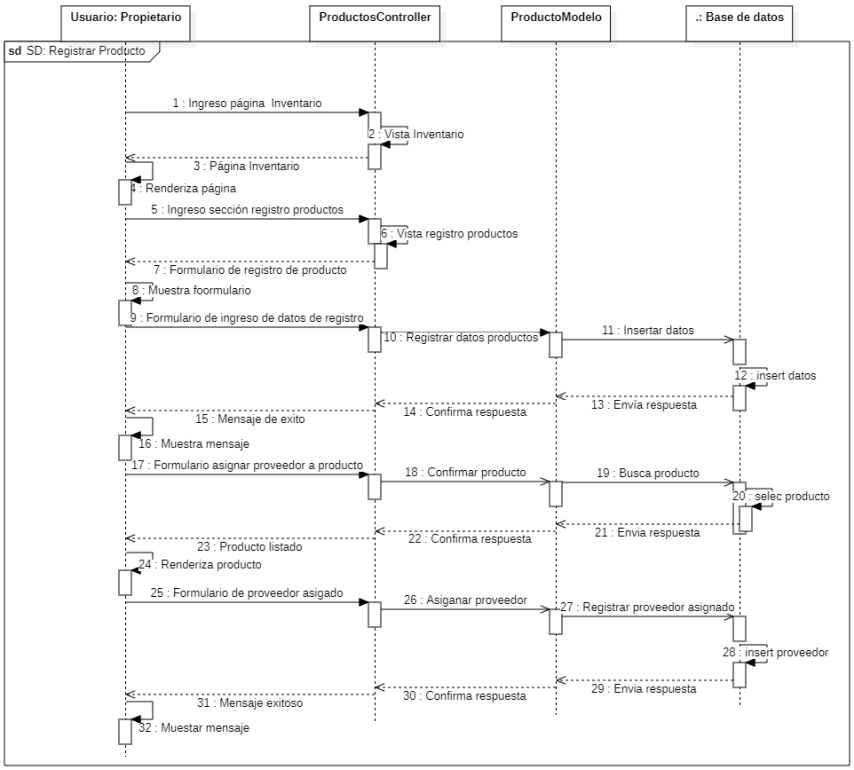


Gráfico 25. Diagrama de secuencia - Registrar producto en inventario.

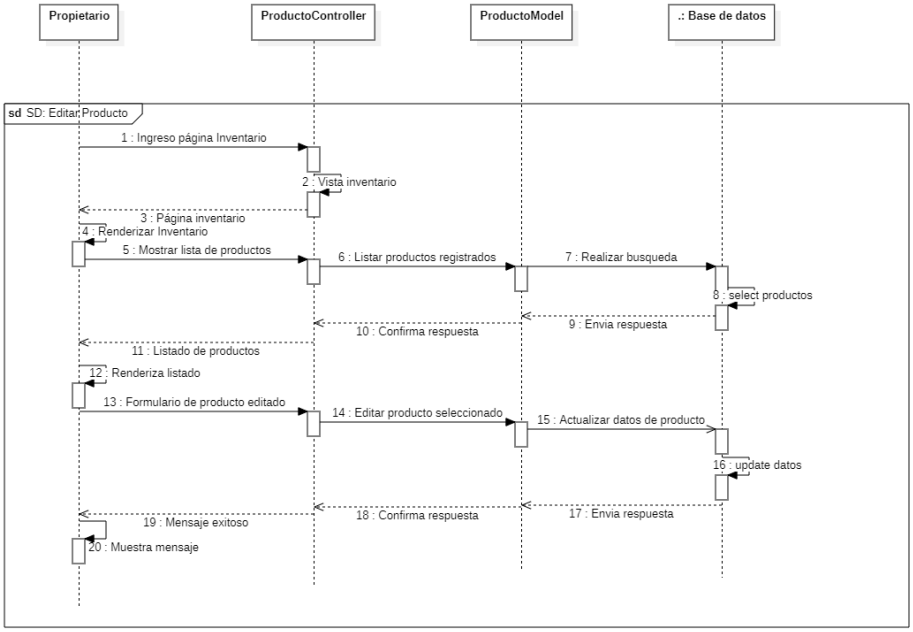


Gráfico 26. Diagrama de secuencia - Editar producto de inventario.

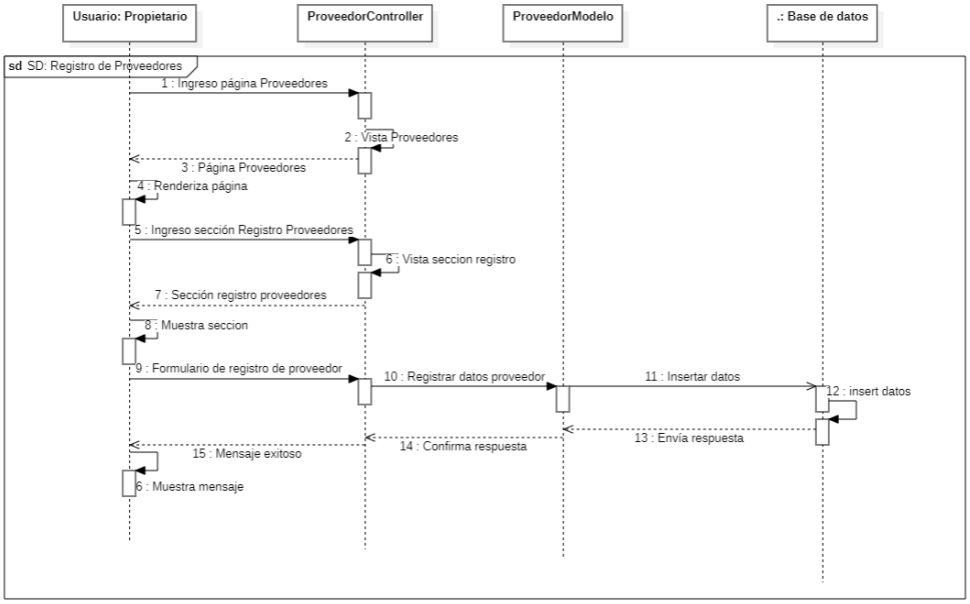


Gráfico 27. Diagrama de secuencia - Registrar proveedor.

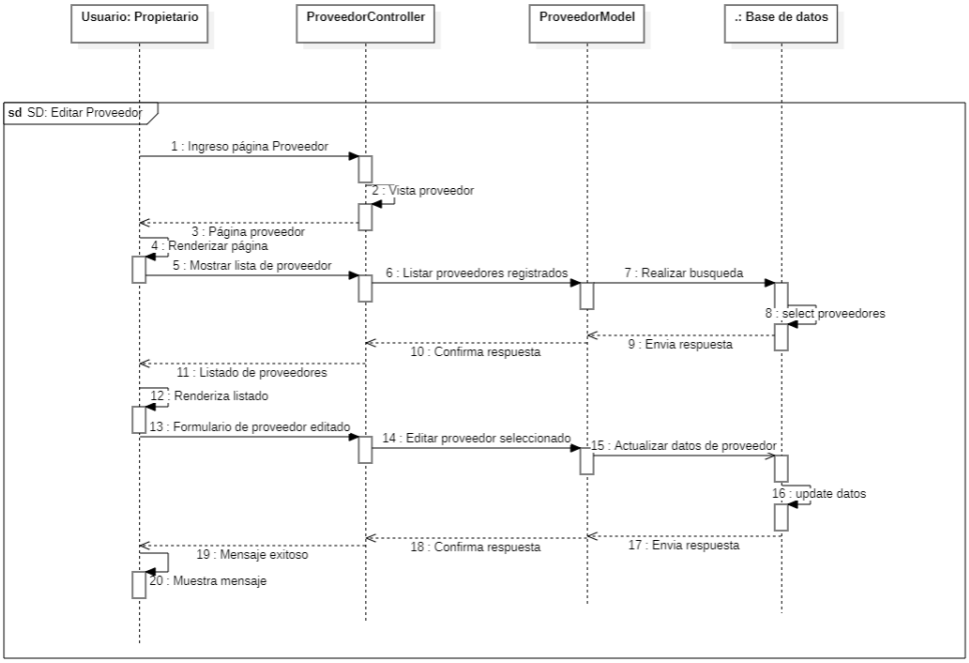


Gráfico 28. Diagrama de secuencia -Editar Proveedor.

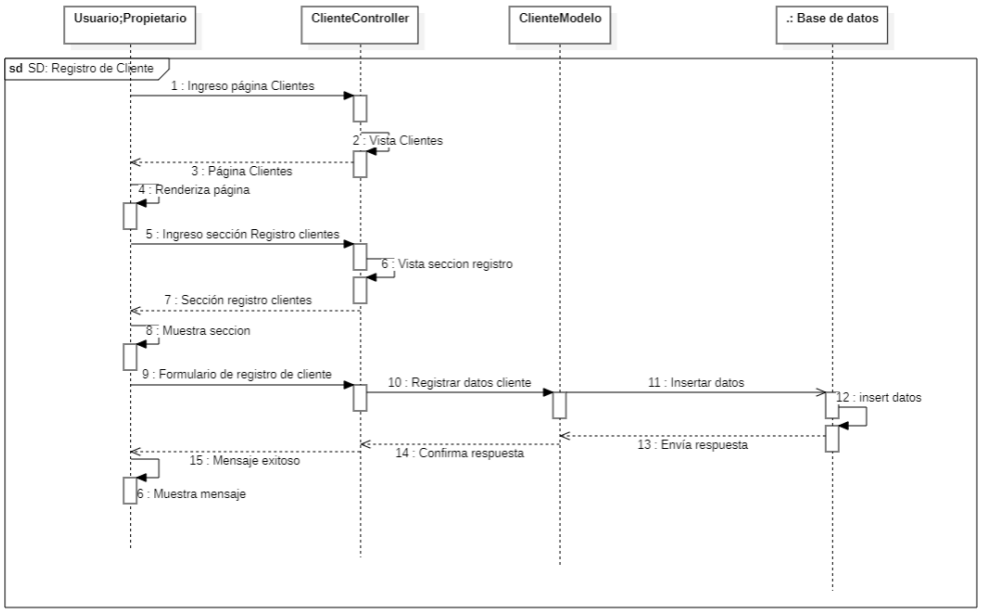


Gráfico 29. Diagrama de secuencia - Registrar cliente.

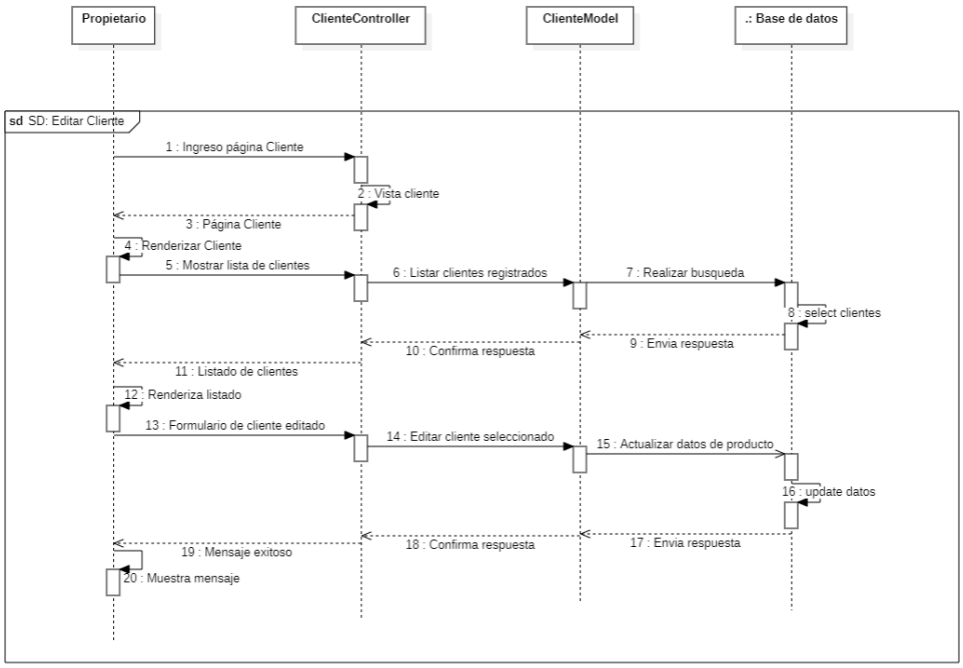


Gráfico 30. Diagrama de secuencia - Editar cliente.

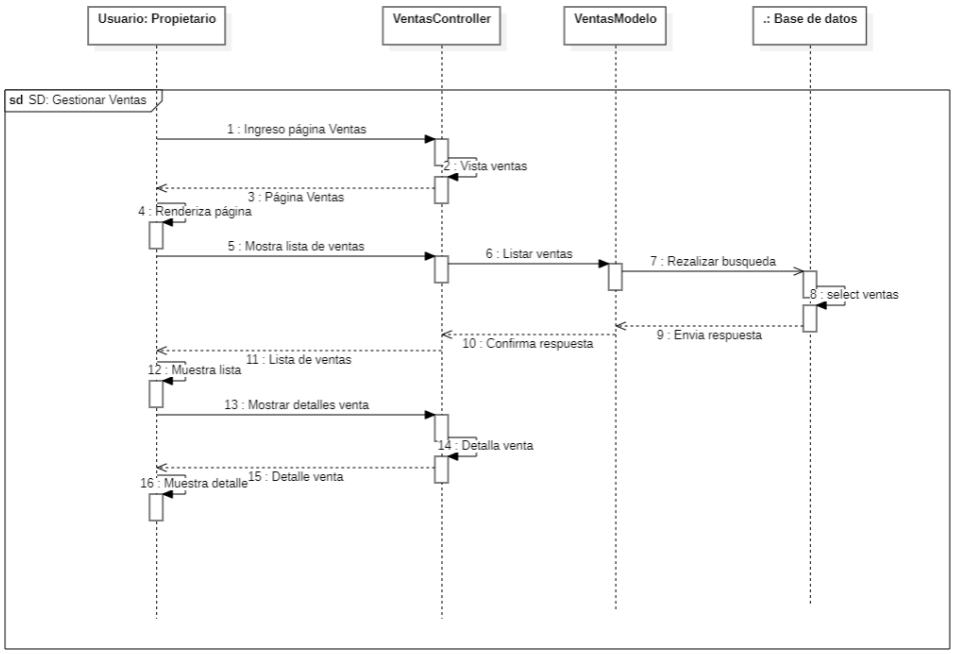


Gráfico 31. Diagrama de secuencia - Gestionar ventas.

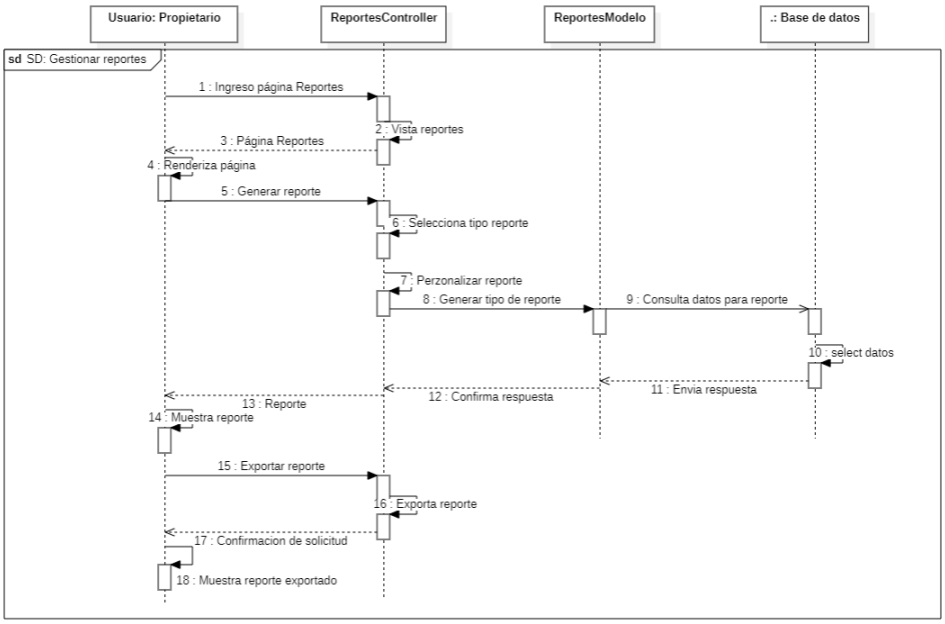


Gráfico 32. Diagrama de secuencia - Gestionar reportes.

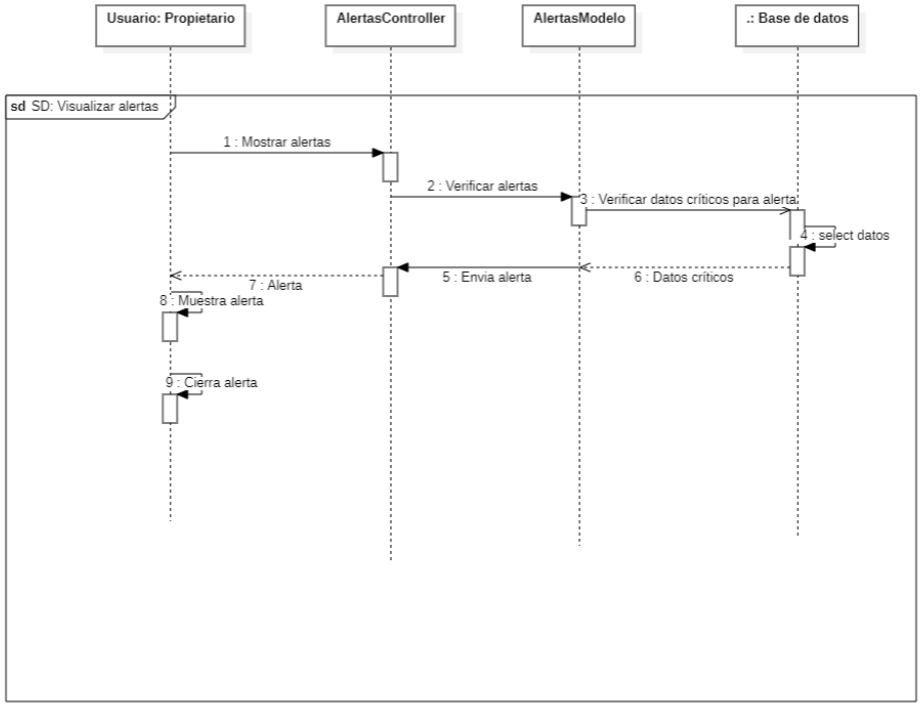


Gráfico 33. Diagrama de secuencia - Visualizar alertas.

Diagramas de secuencia del usuario Contador

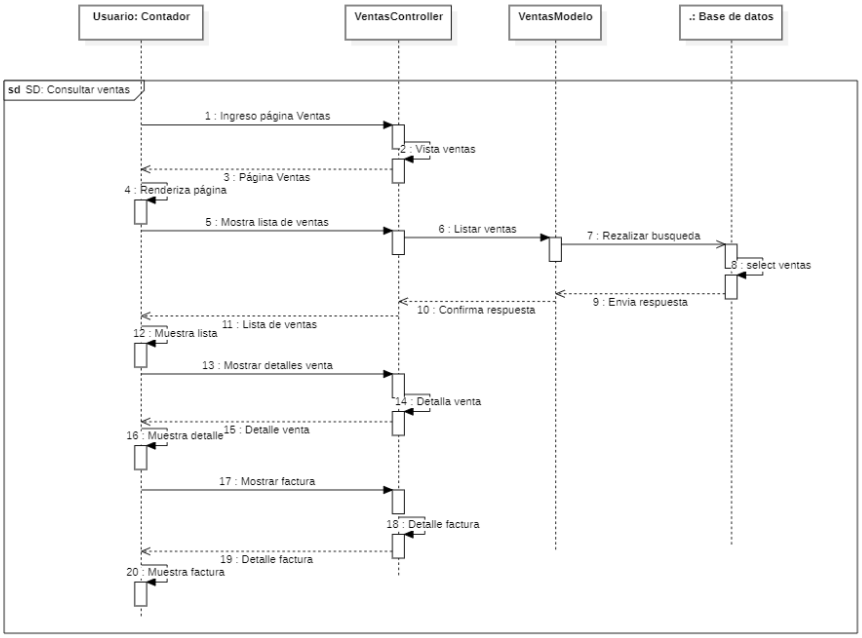


Gráfico 34. Diagrama de secuencia – Consulta de ventas realizadas.

Diagramas de secuencia del usuario Vendedor

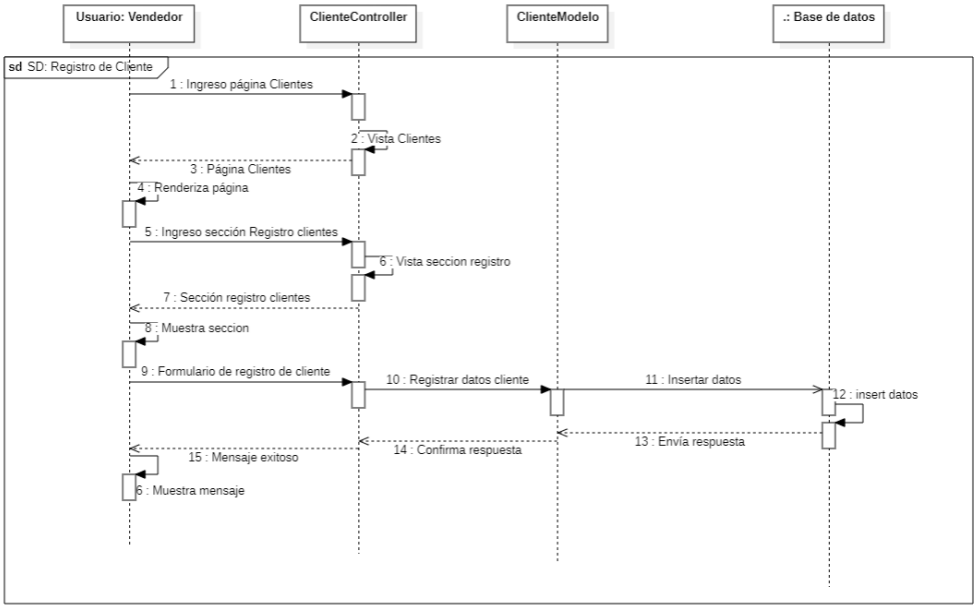


Gráfico 35. Diagrama de secuencia - Registro de cliente por vendedor.

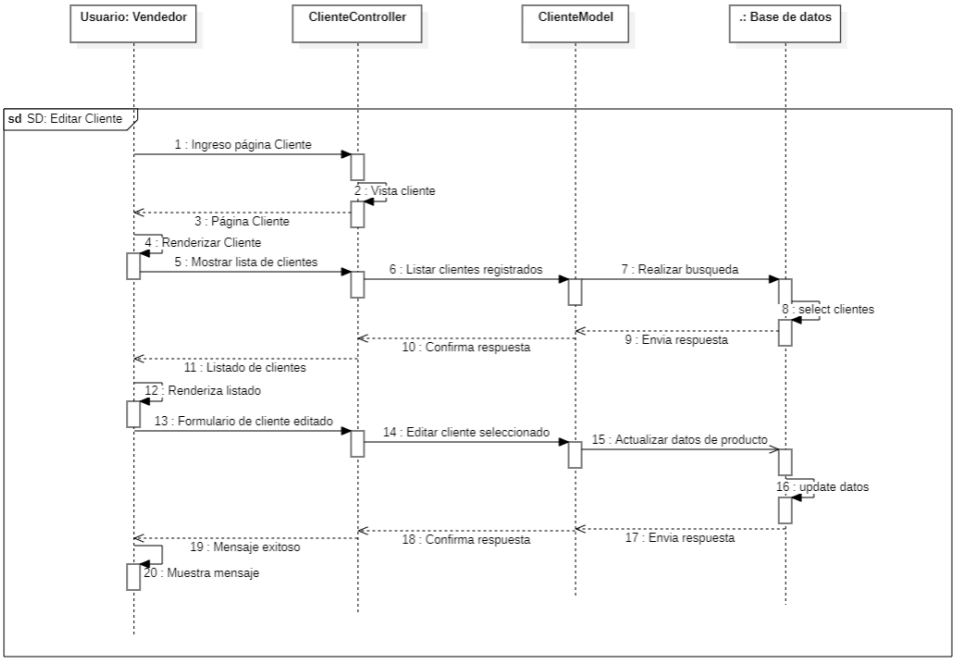


Gráfico 36. Diagrama de secuencia - Editar cliente por vendedor.

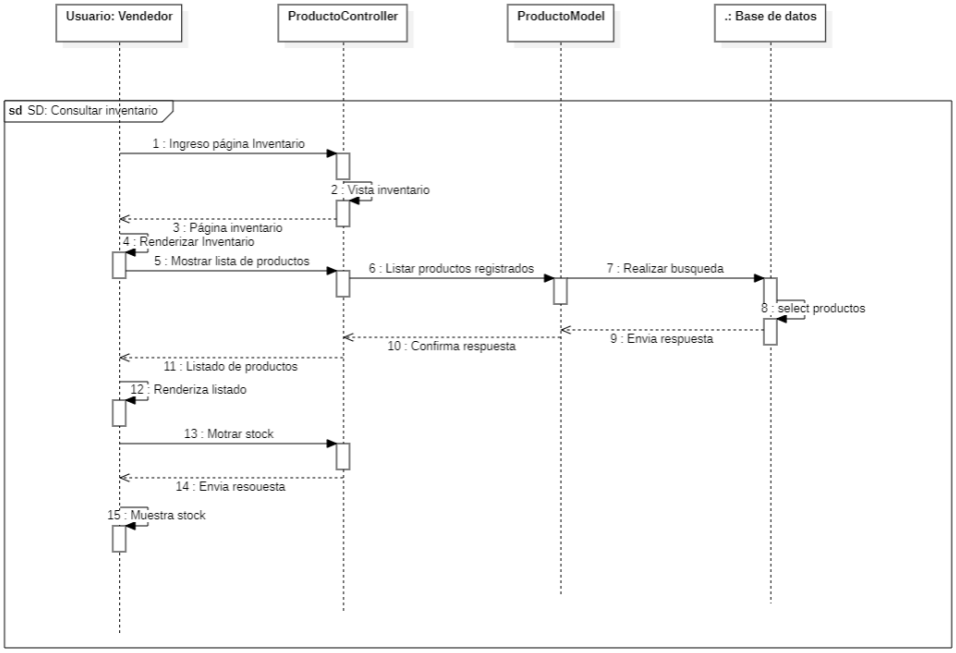


Gráfico 37. Diagrama de secuencia - Consultar inventario.

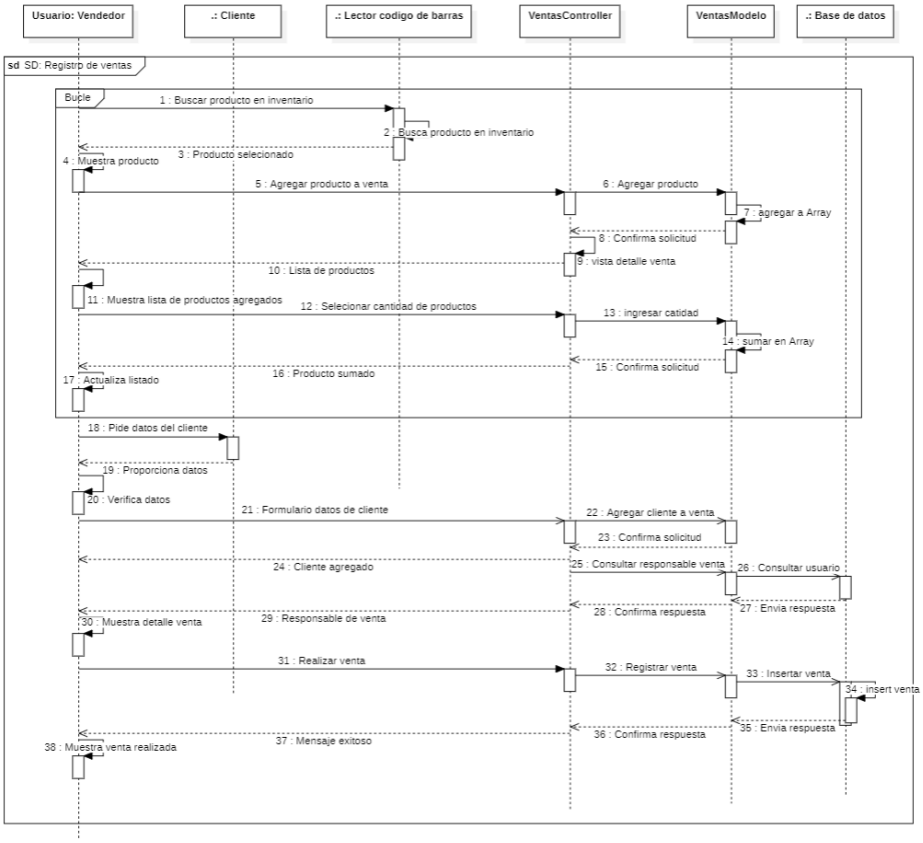


Gráfico 38. Diagrama de secuencia - Registro de venta.

5.3.2.1.3. Diseño de la interfaz de usuario

Diseño de interfaz de la página de inicio de sesión

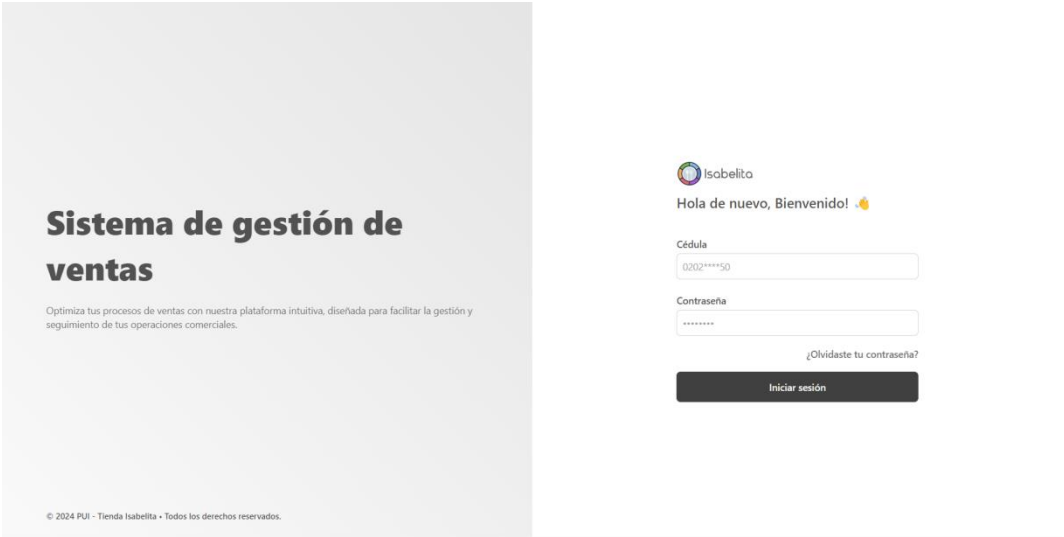


Gráfico 39. Diseño de interfaz de inicio de sesión.

Interfaces de usuario del Propietario.

Diseño de la vista principal del sistema

La vista principal de la aplicación web (Inicio) está diseñada para proporcionar una visión general rápida y accesible del sistema de gestión de ventas, permitiendo a los usuarios interactuar fácilmente con las funcionalidades más importantes.

Componentes de la Vista Principal:

1. Encabezado (Header)
    - Logo del local comercial: Ubicado en la esquina superior izquierda, proporcionando una identificación visual inmediata del local Isabelita.
    - Barra de navegación principal: Menú horizontal que contiene enlaces a las secciones clave del sistema (Inicio, Productos, Clientes, Ventas, Reportes, Usuarios, Proveedores).
    - Perfil del Usuario: Icono del usuario con un menú desplegable que incluye opciones como "Perfil", "Configuración" y "Cerrar Sesión".
2. Panel de Control (Dashboard)
    - Resumen de Ventas: Gráficas y estadísticas que muestran el rendimiento de las ventas diarias, semanales y mensuales.
    - Indicadores Clave de Desempeño (KPIs): Metrías importantes como el total de ventas, número de clientes activos, productos en stock, y otros.
    - Últimas Ventas: Lista de las ventas más recientes realizadas, con detalles básicos como el cliente, producto y monto total.
    - Notificaciones y Alertas: Mensajes importantes para los usuarios, como recordatorios de reabastecimiento de productos o eventos próximos.
3. Sección de Actividades Recientes
    - Registro de Actividades: Historial de las acciones recientes realizadas por los usuarios, como ventas registradas, productos añadidos, o cambios en la configuración del sistema.

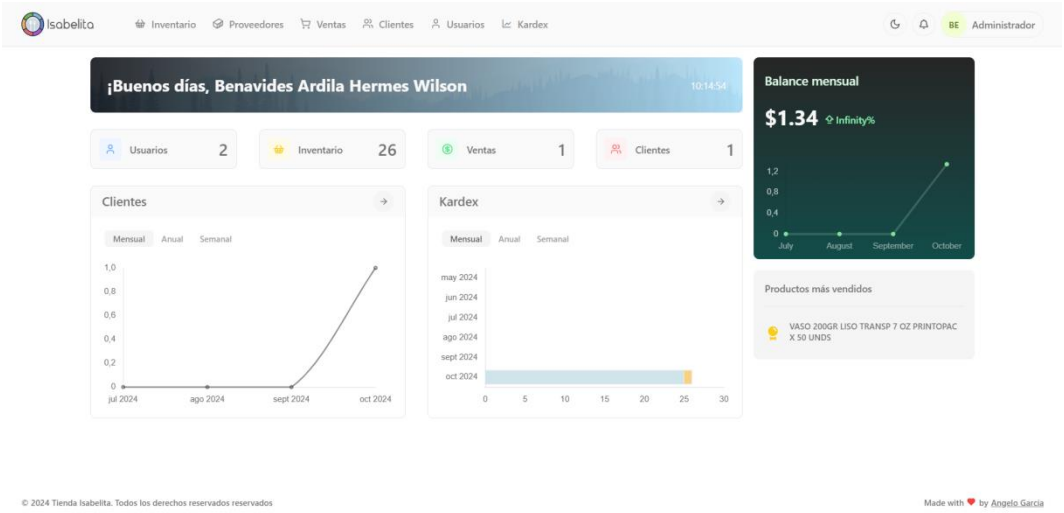


Gráfico 40. Estructura de la vista principal del sistema.

Diseño de interfaz de la página de Inventario

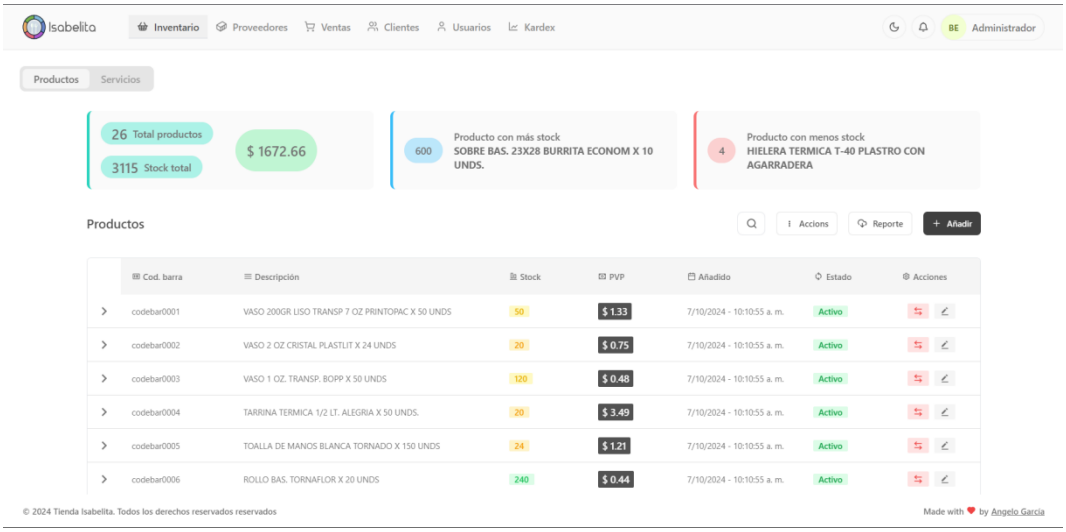


Gráfico 41. Diseño de interfaz del inventario.

Diseño de la interfaz de usuario de la página usuarios

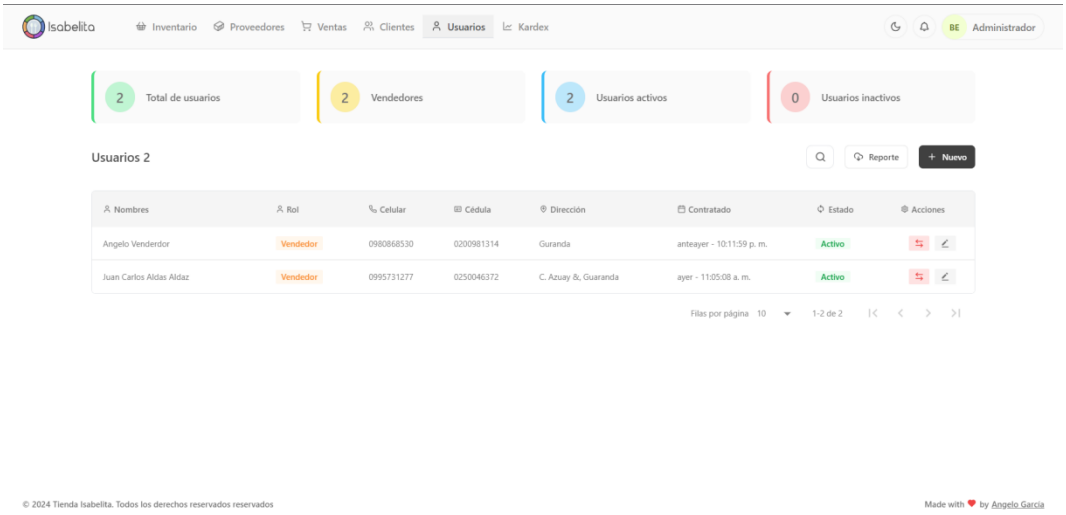


Gráfico 42. Diseño de interfaz de la página usuarios.

Diseño de la interfaz de usuario de la página clientes

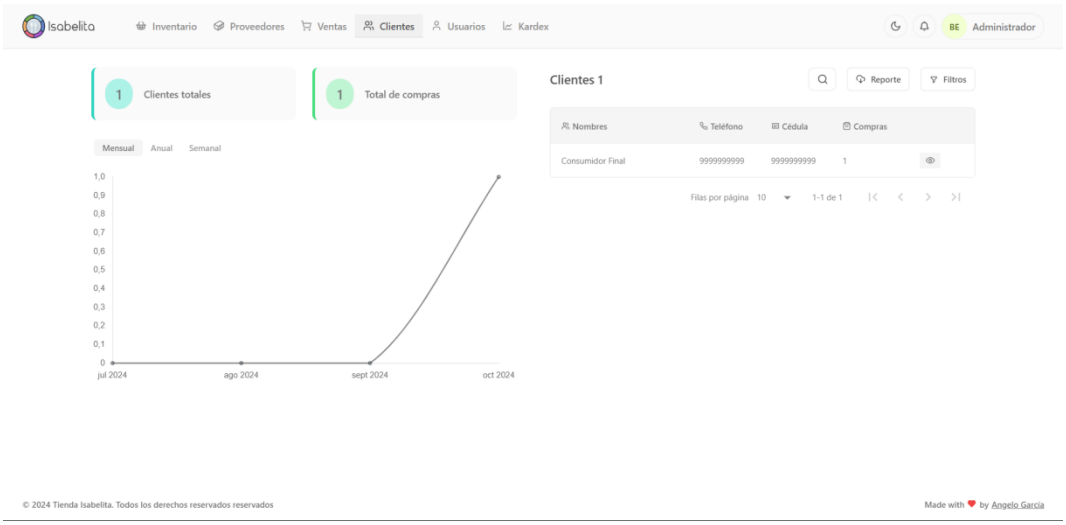


Gráfico 43. Diseño de interfaz de la página clientes.

Diseño de la interfaz de usuario de la página ventas

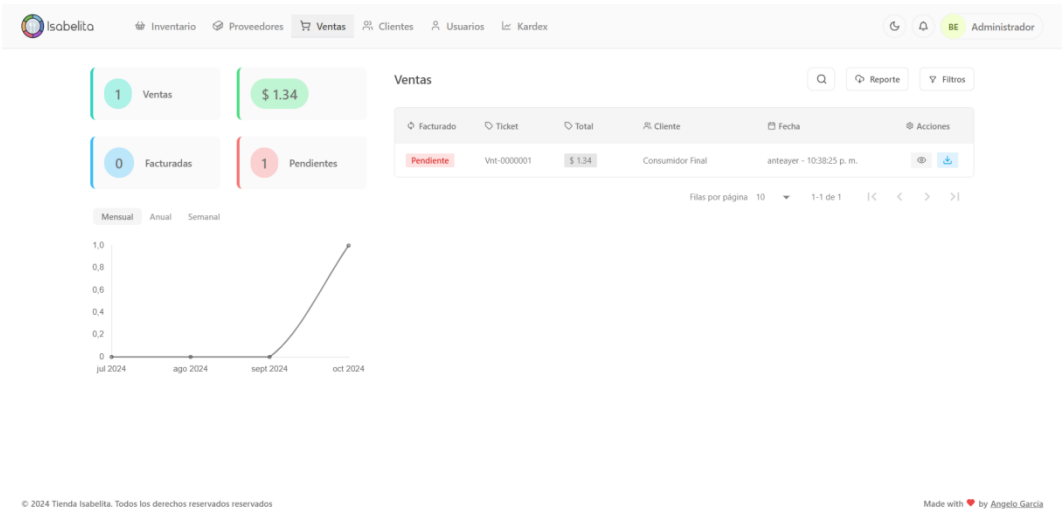


Gráfico 44. Diseño de interfaz de la página ventas.

Diseño de la interfaz de usuario de la página de configuraciones

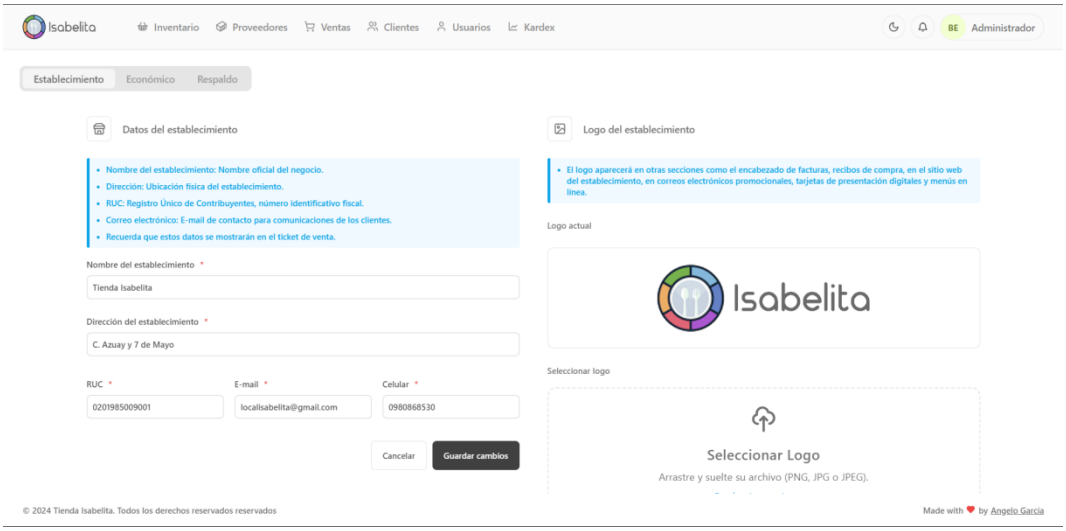


Gráfico 45. Diseño de interfaz de las configuraciones del sistema.

Interfaces de usuario del Vendedor.

Diseño de la interfaz de usuario de la página principal del vendedor.

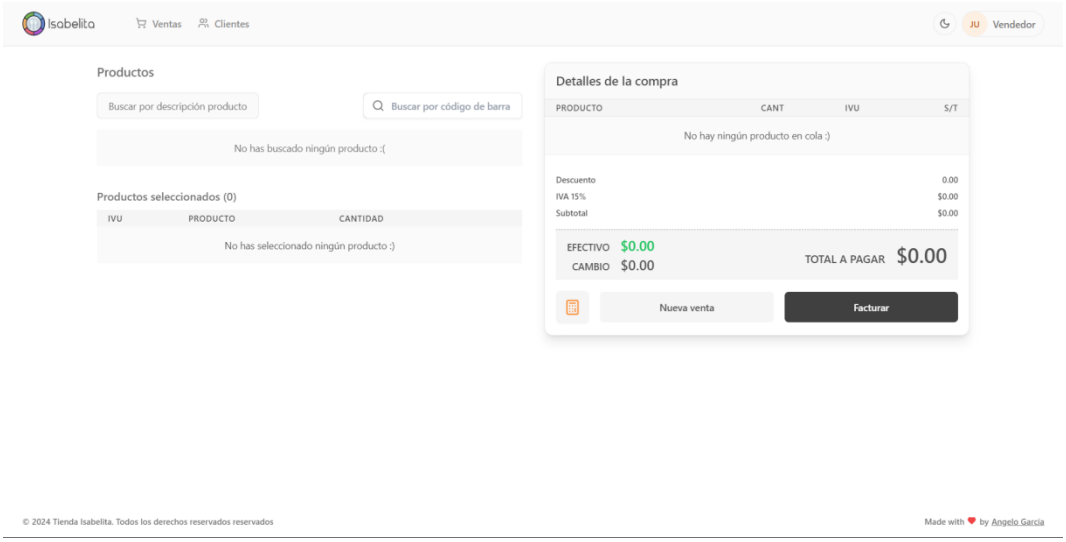


Gráfico 46. Diseño de interfaz de la página principal del vendedor.

Diseño de la interfaz de usuario de la página de ventas

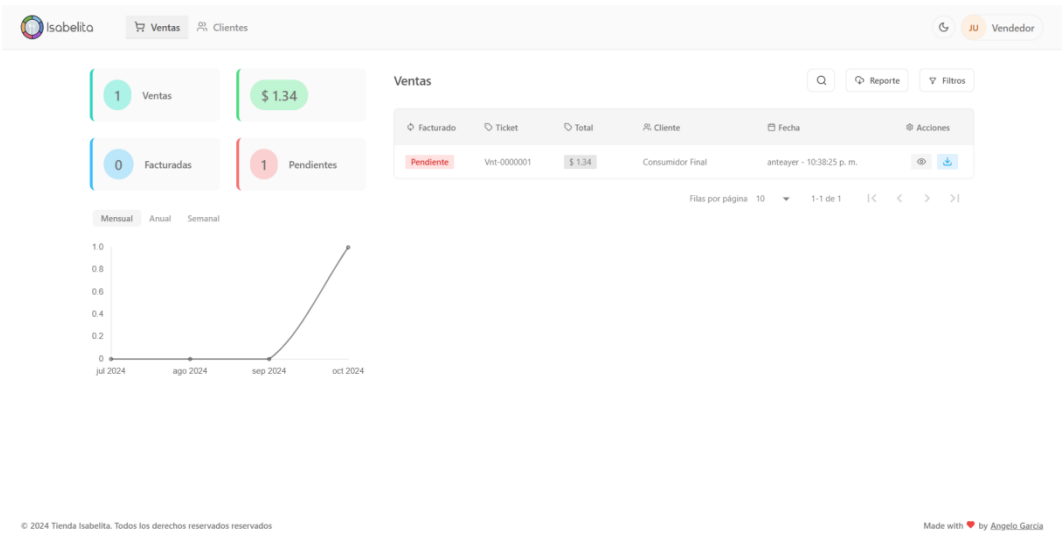


Gráfico 47. Diseño de la interfaz de usuario de la página de ventas.

Diseño de la interfaz de usuario de la página de clientes

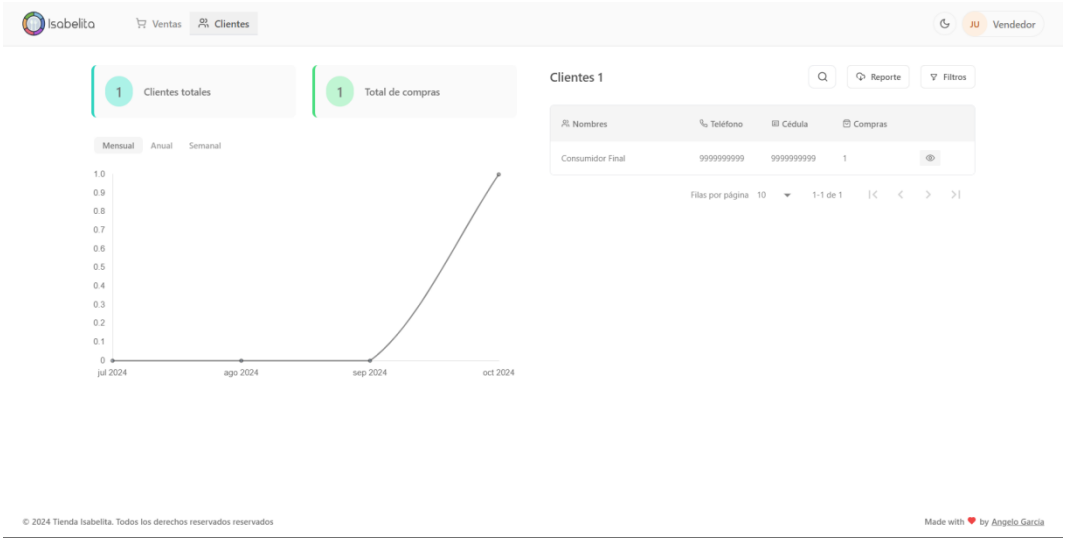


Gráfico 48. Diseño de la interfaz de usuario de la página de clientes.

Interfaces de usuario del Contador.

Interfaz de usuario de la página principal del contador fiscal.

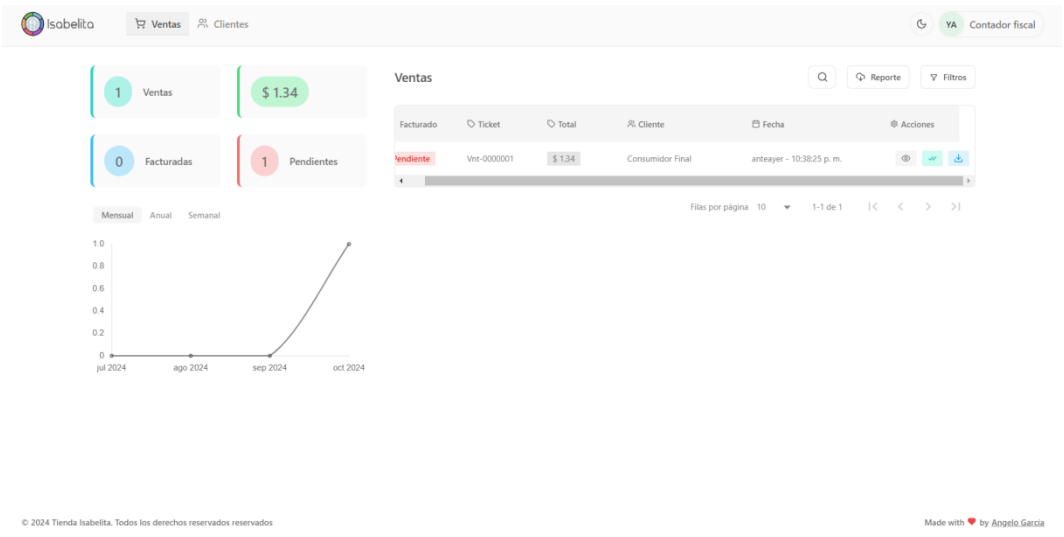


Gráfico 49. Diseño de la interfaz de usuario de la página principal del contador fiscal.

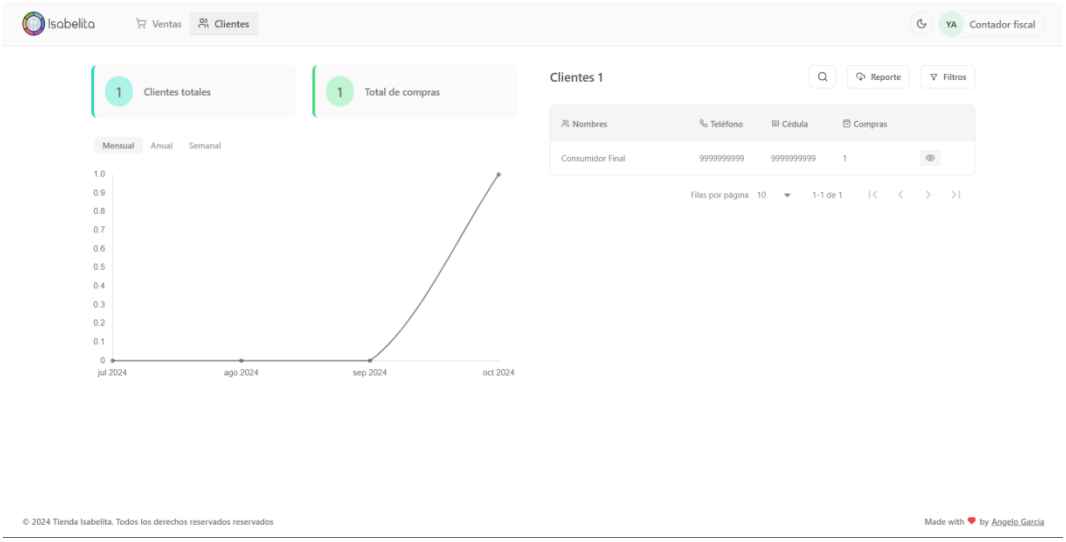


Gráfico 50. Diseño de la interfaz de la página principal del clientes registrados.

5.3.3. Desarrollo del sprint 3. Programación Backend

3.1.7.18. Planificación del sprint 3

En el desarrollo del sprint 3, se realizó la programación del backend del sistema online para la gestión de ventas del local comercial Isabelita, el sprint tiene una duración de 35 días, con un promedio de 7 horas de trabajo diarias.

El objetivo de este sprint es desarrollar la funcionalidad de gestión de usuarios, productos, ventas, inventario, reportes y análisis de ventas, así como la implementación de la lógica de negocio para el sistema en general.

Se identificaron 19 tareas que se debían desarrollar durante el sprint. Estas tareas incluyen la implementación de todas las funcionalidades del sistema a nivel de backend, como la funcionalidad de inicio de sesión (HU1), la creación de un usuario administrador (HU2), el registro y gestión de usuarios (HU3), el registro de productos (HU4), la consulta de inventario de productos (HU5), el registro de ventas (HU6), el registro de clientes (HU7), la gestión de inventario (HU8), la generación de reportes (HU9), la búsqueda y consulta (HU10), el análisis de ventas (HU11), consulta de ventas realizadas, facturas y no facturadas (HU12) y la visualización de alertas (HU13).

Se priorizarán las tareas con mayor riesgo en desarrollo y mayor prioridad, como la gestión de inventario que incluye registro, edición y baja de productos, el registro de las ventas y el registro de clientes. Se utilizó una estrategia de desarrollo

incremental, donde se desarrollaron las tareas de manera secuencial, pero con una revisión y refactorización continua para asegurar la calidad del código.

Para desarrollar estas funcionalidades se toma en cuenta la creación de una API REST para interactuar con la base de datos y realizar operaciones CRUD sobre los datos, además de los modelos y controladores del MVC, donde estos pueden ser reutilizados para las funcionalidades de más de un usuario.

Para asegurar el correcto funcionamiento del backend se usó extensiones del editor de código como REST Client, para verificar las peticiones y respuestas de la API sin la necesidad de un frontend durante este sprint.

3.1.7.19. Tareas del sprint 3

Tabla 23. Tareas del sprint 3 -Desarrollo Backend.

ID	Rol	Tareas	Tiempo de estimación (horas)
1	Desarrollador.	Configurar el entorno de desarrollo y crear la estructura del sistema según la arquitectura MVC.	6
2	Desarrollador.	Crear el esquema de datos para acceder a la base de datos.	12
3	Desarrollador.	Crear API REST para interactuar con la base de datos y crear las funcionalidades del sistema.	24
4	Desarrollador.	Crear modelos y controladores para manejar las solicitudes HTTP.	18
5	Desarrollador.	Desarrollar la lógica para los controladores y modelos del inicio de sesión.	8
6	Desarrollador.	Crear usuario administrador.	4
7	Desarrollador.	Realizar el registro y gestión de usuarios.	14

8	Desarrollador.	Desarrollar el registro de productos para el inventario.	12
9	Desarrollador.	Desarrollar la lógica para las funcionalidades de consulta de inventario de productos.	10
10	Desarrollador.	Realizar el registro de venta.	14
11	Desarrollador.	Crear el registro y gestión de clientes.	12
12	Desarrollador.	Realizar la gestión de inventario (registro, edición y baja de productos)	20
13	Desarrollador.	Desarrollar la lógica para las funcionalidades búsqueda y consulta de clientes, proveedores, ventas y registros.	14
14	Desarrollador.	Desarrollar la lógica para el análisis de ventas.	16
15	Desarrollador.	Crear la funcionalidad de generación de reportes.	14
16	Desarrollador.	Realizar la funcionalidad consulta de consulta de ventas realizadas, facturas y no facturadas para el usuario contador.	8
17	Desarrollador.	Desarrollar la visualización de alertas para el sistema.	8
18	Desarrollador, Product Owner.	Realizar comprobaciones de funcionamiento de la API (petición, respuesta) en el backend.	18
19	Desarrollador.	Refactorizar y optimizar el código para mejorar el rendimiento.	14
		Total	**246**

3.1.7.20. Ejecución del sprint 3 – Scrum diario

Tabla 24. Scrum diario del sprint 3 - Desarrollo Backend.

Día	Miembro del Equipo	Tarea Realizada	Impedimentos	Acciones a Seguir
1	Juan Aldaz, Angelo García	S estructuró el proyecto según la arquitectura MVC.		Identificar la lógica del negocio del sistema para crear los modelos y controladores.
2	Juan Aldaz, Angelo García	Se creó la API REST para las acciones CRUD que realizará el modelo y serán manejadas por el controlador.	Falta de claridad en la lógica del negocio del sistema para crear los modelos y controladores.	Revisar la documentación del proyecto y realizar una reunión con el equipo para aclarar la lógica del negocio.
3, 4	Angelo García, Juan Aldaz	Se creó el esquema de datos para la base de datos usando el ORM Sequelize.	Necesidad de aclaración de requisitos.	Reunión con el propietario para aclarar dudas.
5	Angelo García	Se creó el modelo para la gestión de los usuarios y el manejo de acceso al sistema por inicio de sesión.		
6	Angelo García	Se creó el usuario Administrador.		Comprobar la creación del usuario

				Administrador en la base de datos.
7, 8	Juan Aldaz	Se realizó el modelo para la gestión de los productos del inventario.	Dependencia del modelo para la gestión de los proveedores.	Verificar que todos los productos se ingresen correctamente desde una factura en formato xml. Esperar la creación del modelo para los proveedores.
9	Juan Aldaz	Se realizó el modelo para la gestión de los proveedores de los productos.	Problemas en la relación entre productos y proveedores debido a la relación múltiple.	Modificar el modelo de los productos para la relación múltiple entre proveedores y productos.
10, 11	Angelo García	Se creo el modelo de datos para la gestión de las ventas.	Dependencia con el modelo de la gestión de los clientes.	Esperar la creación del modelo del cliente con el que tiene dependencia.
12	Juan Aldaz	Se realizó el modelo para la gestión de los clientes.		
13	Angelo García	Se construyó el modelo para realizar las configuraciones del sistema (datos del local comercial)		

14	Angelo García	Se desarrolló el modelo para realizar los ajustes económicos (porcentajes de ganancia y otros)		
	Juan Aldaz	Se creó el modelo para el registro de las acciones de los usuarios.		
15	Angelo García	Se creó el controlador para manejar la gestión de los usuarios.		Integrar con el modelo de usuarios.
16	Angelo García	Se creó el controlador de inicio de sesión: implementar la lógica de inicio de sesión.		Continuar con la implementación de la lógica de acceso a las funcionalidades según el rol del usuario y los permisos.
17	Juan Aldaz	Se creó el controlador para el manejo de los productos.		Integrar con el modelo de los productos.
18	Juan Aldaz	Se desarrolló el controlador para el manejo de los proveedores.		Conectar con el modelo de los proveedores.
19, 20	Angelo García	Se desarrolló el controlador de	Falta de claridad en la lógica de	Consultar con el propietario los

		consulta de inventario de productos.	consulta y los resultados esperados por el cliente.	resultados que esperaría y posteriormente aplicar filtros de búsqueda según los resultados esperados.
21	Angelo García	Se creó el controlador para la gestión de las ventas.	Dependencia del controlador para el cliente.	Esperar la creación del controlador para el cliente. Integrar con el modelo de ventas.
22	Juan Aldaz	Se desarrolló el controlador para la administración del cliente.		Integrar con el modelo del cliente.
23	Angelo García	Se desarrolló la lógica del modelo para las configuraciones del sistema.		Conectar con el modelo para las configuraciones del sistema.
24	Angelo García	Se creó el controlador para los ajustes económico.		Integrar con el modelo de los ajustes económicos.
25	Juan Aldaz	Se realizó el controlador para el registro de acciones de los usuarios.		Implementar con el modelo para los registros de acciones.

26, 27	Juan Aldaz	Se creó el controlador de generación de reportes.	Falta de claridad con los tipos de reportes que desea el cliente y la lógica para su implementación. Dependencia con el controlador de análisis de ventas para ese reporte.	Conectar con el modelo de reportes. Reunión con el propietario para determinar los tipos de reportes que desearía obtener. Esperar a que se cree el controlador de análisis de ventas.
28	Angelo García	Se realizó el controlador de búsqueda y consulta para los clientes, proveedores y ventas.		Verificar la creación del controlador para la consulta en cada modelo.
29	Angelo García	Se desarrolló el controlador de análisis de ventas.		Integrar con el controlador de reportes.
30	Angelo García	Se creó el controlador de consulta de consulta de ventas realizadas, facturas y no facturadas para el usuario contador.	Dudas con la forma en la que accede a la información de las ventas el usuario contador.	Definir la forma en la que accede a la información de las ventas el usuario contador.

31	Juan Aldaz	Se creó el controlador de visualización de alertas: implementar la lógica para alertas y las notificaciones.		Continuar con la creación del controlador.
32	Juan Aldaz, Angelo García	Se realizó la integración de los controladores con la API.		Integrar los controladores en las rutas de la API.
33	Juan Aldaz, Angelo García	Se comprobó el funcionamiento de la API.	No todos las rutas de la API funcionan correctamente.	Revisar la lógica de los controladores y modelos.
34, 35	Juan Aldaz, Angelo García	Se realizaron pruebas de petición y respuestas a todas las rutas para determinar el correcto funcionamiento del backend.	No todas las rutas funcionan correctamente.	Continuar con las correcciones y refactorización del código según sea necesario para asegurar los datos de entrada y salida correctos.

3.1.7.21. Resultados

Como resultados del sprint 3 se han completado 31 actividades de las tareas planteadas para el desarrollo del backend del sistema que incluyen la creación de la API REST, la creación del esquema de datos para la base de datos, la creación de modelos y controladores para la gestión de usuarios, productos, proveedores,

ventas, clientes, configuraciones del sistema, ajustes económicos, registro de acciones de los usuarios y alertas y notificaciones.

5.3.3.1.1. Desarrollo del Backend

Creación de entorno backend e instalación de paquetes

Para el desarrollo en el backend se instalaron las dependencias necesarias para el proyecto utilizando el administrador de paquetes npm (Node Package Manager) de Node.js. Se ejecutó el comando npm install seguido de los nombres de las dependencias requeridas, que son:

- express: un framework de aplicaciones web para Node.js que ayuda con la creación y control de la funciones del servidor.
- sequelize: un ORM (Object-Relational Mapping) para interactuar con bases de datos relacionales.
- zod: una biblioteca para validar y parsear datos en Node.js.
- cors: un middleware para habilitar el acceso a recursos desde orígenes cruzados (Cross-Origin Resource Sharing)
- pg: un driver para conectar a bases de datos PostgreSQL desde Node.js.
- dotenv: una biblioteca para cargar variables de entorno desde un archivo. env.
- bcrypt: una biblioteca para hashear y verificar contraseñas
- jsonwebtoken: una biblioteca para generar y verificar tokens de autenticación JSON Web Tokens
- cookie-parser: un middleware para parsear cookies en solicitudes HTTP
- nodemon: un paquete de desarrollo que se utiliza para reiniciar automáticamente el servidor cuando se realizan cambios en el código

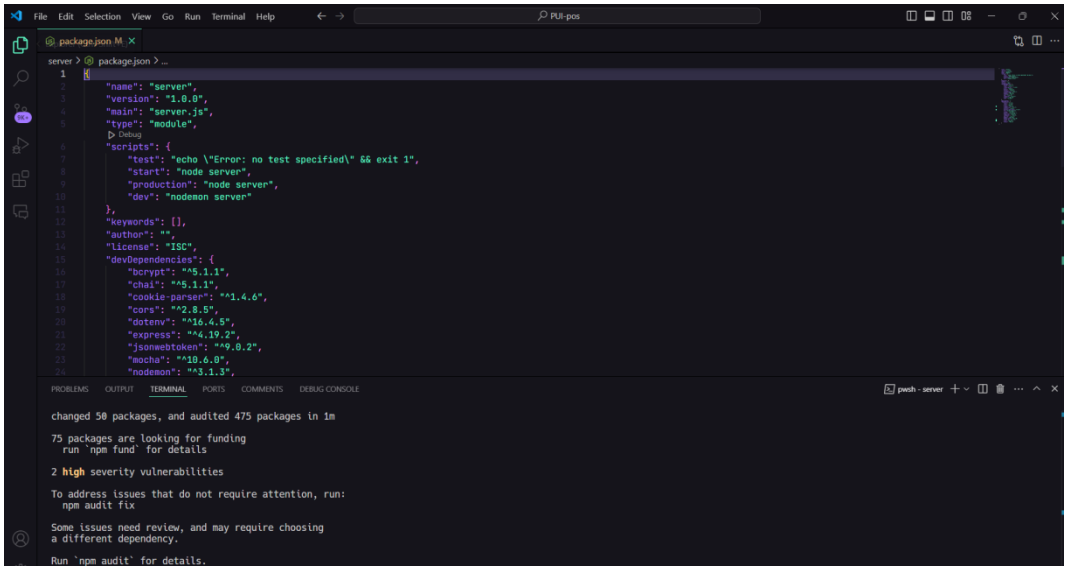


Gráfico 51. Inicialización de dependencias para el entorno Backend.

Conexión con la base de datos de PostgresQL

Se configuró la conexión a la base de datos de PostgreSQL utilizando el driver *pg*, creando un archivo de configuración que establece la conexión a la base de datos usando un archivo de variables de entorno que contiene datos como la dirección del servidor, el nombre de la base de datos, el usuario y la contraseña ya establecidas. Para establecer la conexión es necesario importar el módulo *sequelize* y crear una instancia de la conexión a la base de datos utilizando la configuración previamente establecida.

Algoritmo de conexión a la base de datos

```
import dotenv from 'dotenv'
import { Sequelize } from 'sequelize'
dotenv.config()
const sequelize = new Sequelize(process.env.DB_NAME, process.env.DB_USER,
process.env.DB_PASSWORD, {
    host: process.env.DB_HOST,
    port: process.env.DB_PORT,
    dialect: 'postgres',
    logging: false,
})
```

```
export { sequelize }
```

Creación de la API REST y el servidor

Se creó y configuró un servidor web utilizando el framework *express* para escuchar solicitudes en un puerto específico y se creó una API REST para interactuar con la base de datos. La API REST se creó en el archivo **server.js** y se diseñó para manejar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) sobre los datos almacenados en la base de datos.

Algoritmo de crear el servidor y la API REST

```
import http from 'http'
import dotenv from 'dotenv'
import express from 'express'
import routes from './routes.js'
import cookieParser from 'cookie-parser'
import { sequelize } from './config/db.config.js'
import { setupCors } from './config/cors.config.js'
import { startServer } from './utils/server.utils.js'
import { setupMorgan } from './middlewares/morgan.middleware.js'
import { startServerStatusCron } from './services/serverStatusCron.js'
import { limiterRequest } from './middlewares/rateLimit.middleware.js'
import { setupSocketIO } from './config/socket.config.js'

dotenv.config()
const app = express()
const server = http.createServer(app)
// Configuración de middleware
setupCors(app)
setupMorgan(app)
// Middleware para parsing de cookies y JSON
app.use(cookieParser())
app.use(express.json())

// Rutas de la API
```

```
app.use('/api/', limiterRequest, routes)
// Configuración de Socket.IO
setupSocketIO(server)

// Sincronizar base de datos y arrancar el servidor
sequelize
   .sync({ force: false })
   .then(() => {
      startServer(server)
      startServerStatusCron()
   })
   .catch(err => {
      console.error('Error syncing with database:', err)
   })
```

Creación del esquema de datos para la base de datos

Se diseñó y creó el esquema de datos para la base de datos utilizando el ORM sequelize. El esquema de datos define la estructura de las tablas de todas las entidades y las relaciones entre ellas, y maneja directamente las operaciones de lectura y escritura en la base de datos.

Algoritmo del esquema para la base de datos

```
import { DataTypes } from 'sequelize'
import { sequelize } from '../config/db.config.js'

/* ------------------------------ TABLE USERS ------------------------------*/

const userScheme = sequelize.define(
   'user',
   {
      id_user: {
         type: DataTypes.UUID,
         defaultValue: DataTypes.UUIDV4,
         primaryKey: true,
```

```
},
active: {
    type: DataTypes.BOOLEAN,
    allowNull: false,
    defaultValue: true,
},
username: {
    type: DataTypes.STRING,
    allowNull: false,
},
email: {
    type: DataTypes.STRING,
    allowNull: false,
    unique: true,
    validate: {
        isEmail: true,
    },
},
password: {
    type: DataTypes.STRING,
    allowNull: false,
},
phone: {
    type: DataTypes.STRING,
    allowNull: false,
},
identification_card: {
    type: DataTypes.STRING,
    allowNull: false,
    unique: true,
},
residence: {
```

```
            type: DataTypes.STRING,
            allowNull: false,
        },
        securityWord: {
            type: DataTypes.STRING,
            allowNull: false,
        },
    },
    {
        tableName: 'users',
        timestamps: true,
    }
)

/* ------------------------------ TABLE ROLES -----------------------------*/

const rolScheme = sequelize.define(
    'role',
    {
        id_rol: {
            type: DataTypes.UUID,
            defaultValue: DataTypes.UUIDV4,
            primaryKey: true,
        },

        type_rol: {
            type: DataTypes.STRING,
            allowNull: false,
            defaultValue: 'employee',
            validate: {
                isIn: [['admin', 'employee']],
            },
```

```
        },

        id_user_fk: {
            type: DataTypes.UUID,
            defaultValue: DataTypes.UUIDV4,
        },
    },
    {
        tableName: 'roles',
        timestamps: true,
    }
)

userScheme.hasOne(rolScheme, {
    foreignKey: 'id_user_fk',
    sourceKey: 'id_user',
})

/* ------------------------------ TABLE CODE OTP -----------------------------*/

const otpCodeScheme = sequelize.define(
    'otpCode',
    {
        id_code: {
            type: DataTypes.UUID,
            defaultValue: DataTypes.UUIDV4,
            primaryKey: true,
        },
        operation_type: {
            type: DataTypes.STRING,
            allowNull: false,
            defaultValue: 'login',
```

```
            validate: {
                isIn: [['login', 'update']],
            },
        },
        code: {
            type: DataTypes.STRING,
            allowNull: false,
        },
        id_user_fk: {
            type: DataTypes.UUID,
            allowNull: false,
            references: {
                model: userScheme,
                key: 'id_user',
            },
        },
    },
    {
        tableName: 'otpCodes',
        timestamps: true,
    }
)

otpCodeScheme.belongsTo(userScheme, {
    foreignKey: 'id_user_fk',
    targetKey: 'id_user',
})

/* ------------------------------ TABLE PRODUCT ------------------------------*/

const productScheme = sequelize.define(
    'product',
```

```
{
  id_product: {
    type: DataTypes.UUID,
    defaultValue: DataTypes.UUIDV4,
    primaryKey: true,
  },
  code_product: {
    type: DataTypes.STRING,
    allowNull: false,
  },
  code_bar: {
    type: DataTypes.STRING,
    allowNull: true,
  },
  code_primary: {
    type: DataTypes.STRING,
    allowNull: true,
  },
  code_secondary: {
    type: DataTypes.STRING,
    allowNull: true,
  },
  active: {
    type: DataTypes.BOOLEAN,
    allowNull: false,
    defaultValue: true,
  },
  description: {
    type: DataTypes.STRING,
    allowNull: false,
  },
  amount: {
```

```
            type: DataTypes.FLOAT,
            allowNull: false,
        },
        price_unit: {
            type: DataTypes.FLOAT,
            allowNull: false,
        },
        discount: {
            type: DataTypes.FLOAT,
            allowNull: false,
        },
        total_price_without_tax: {
            type: DataTypes.FLOAT,
            allowNull: false,
        },
    },
    {
        tableName: 'products',
        timestamps: true,
    }
)

/* ------------------------------ TABLE PRODUCT - ADDITIONAL ------------------
-----------*/

const additionalDetailProductScheme = sequelize.define(
    'additional_detail_product',
    {
        id_additional_detail_product: {
            type: DataTypes.UUID,
            defaultValue: DataTypes.UUIDV4,
            primaryKey: true,
```

```
        },
        id_product_fk: {
            type: DataTypes.UUID,
            allowNull: false,
            references: {
                model: productScheme,
                key: 'id_product',
            },
        },
        name: {
            type: DataTypes.STRING,
            allowNull: false,
        },
        value: {
            type: DataTypes.STRING,
            allowNull: false,
        },
    },
    {
        tableName: 'additional_detail_product',
        timestamps: false,
    }
)

additionalDetailProductScheme.belongsTo(productScheme, {
    foreignKey: 'id_product_fk',
    targetKey: 'id_product',
})

/* ------------------------------ TABLE SERVICES - DIGITALS ----------------------
------*/
```

```
const serviceDigitalScheme = sequelize.define(
    'service_digital',
    {
        id_service_product: {
            type: DataTypes.UUID,
            defaultValue: DataTypes.UUIDV4,
            primaryKey: true,
        },
        code_primary: {
            type: DataTypes.STRING,
            allowNull: true,
        },
        code_secondary: {
            type: DataTypes.STRING,
            allowNull: true,
        },
        amount: {
            type: DataTypes.FLOAT,
            allowNull: false,
        },
        description: {
            type: DataTypes.STRING,
            allowNull: false,
        },
        price_unit: {
            type: DataTypes.FLOAT,
            allowNull: false,
        },
        iva: {
            type: DataTypes.FLOAT,
            allowNull: false,
        },
```

```
discount: {
    type: DataTypes.FLOAT,
    allowNull: false,
},
total_price: {
    type: DataTypes.FLOAT,
    allowNull: false,
},
iva_product: {
    type: DataTypes.FLOAT,
    allowNull: false,
},
ice: {
    type: DataTypes.FLOAT,
    allowNull: false,
},
irbpnr: {
    type: DataTypes.FLOAT,
    allowNull: false,
},
ruc: {
    type: DataTypes.STRING,
    allowNull: false,
},
serie: {
    type: DataTypes.STRING,
    allowNull: false,
},
name: {
    type: DataTypes.STRING,
    allowNull: false,
},
```

```
        document: {
            type: DataTypes.STRING,
            allowNull: false,
        },
        document: {
            type: DataTypes.STRING,
            allowNull: false,
        },
        id_recipient: {
            type: DataTypes.STRING,
            allowNull: false,
        },
    },
    {
        tableName: 'service_digital',
        timestamps: true,
    }
)

/* ------------------------------ TABLE SALES ------------------------------*/

const salesScheme = sequelize.define(
    'sales',
    {
        id_sale: {
            type: DataTypes.UUID,
            defaultValue: DataTypes.UUIDV4,
            primaryKey: true,
        },

        id_product_fk: {
            type: DataTypes.UUID,
```

```
            allowNull: false,
            references: {
                model: 'products',
                key: 'id_product',
            },
        },

        quantity: {
            type: DataTypes.INTEGER,
            allowNull: false,
        },

        total_price: {
            type: DataTypes.FLOAT,
            allowNull: false,
        },
    },
    {
        tableName: 'sales',
        timestamps: true,
    }
)

productScheme.hasMany(salesScheme, {
    foreignKey: 'id_product_fk',
    sourceKey: 'id_product',
})

salesScheme.belongsTo(productScheme, {
    foreignKey: 'id_product_fk',
    targetKey: 'id_product',
})
```

```
/* ------------------------------ TABLE SUPPLIERS ------------------------------*/

const suppliersScheme = sequelize.define(
   'suppliers',
   {
      id_suppliers: {
         type: DataTypes.UUID,
         defaultValue: DataTypes.UUIDV4,
         primaryKey: true,
      },
      company_name: {
         type: DataTypes.STRING,
         allowNull: false,
      },
      commercial_name: {
         type: DataTypes.STRING,
         allowNull: false,
      },
      ruc: {
         type: DataTypes.STRING,
         allowNull: false,
      },
      address: {
         type: DataTypes.STRING,
         allowNull: false,
      },
   },
   {
      tableName: 'suppliers',
      timestamps: true,
   }
```

```
)

/* ------------------------------ TABLE BILL PRODUCT ----------------------------*/

const billProductScheme = sequelize.define(
   'bill_product',
   {
      id_bill_product: {
         type: DataTypes.UUID,
         defaultValue: DataTypes.UUIDV4,
         primaryKey: true,
      },
      sequential_code: {
         type: DataTypes.STRING,
         allowNull: false,
      },
      responsible: {
         type: DataTypes.UUID,
         allowNull: false,
         references: {
            model: userScheme,
            key: 'id_user',
         },
      },
   },
   {
      tableName: 'bill_product',
      timestamps: true,
   }
)

billProductScheme.belongsTo(userScheme, {
```

```
    foreignKey: 'responsible',
    targetKey: 'id_user',
})

/* ------------------------------ TABLE SETTINGS ----------------------------*/

const settingsScheme = sequelize.define(
    'settings',
    {
        id_setting: {
            type: DataTypes.UUID,
            defaultValue: DataTypes.UUIDV4,
            primaryKey: true,
        },

        property_name: {
            type: DataTypes.STRING,
            allowNull: false,
        },

        ruc: {
            type: DataTypes.STRING,
            allowNull: false,
        },

        address: {
            type: DataTypes.STRING,
            allowNull: false,
        },

        email: {
            type: DataTypes.STRING,
```

```
        allowNull: false,
      },

      phone: {
        type: DataTypes.STRING,
        allowNull: false,
      },
    },
    {
      tableName: 'settings',
      timestamps: true,
    }
)

/* ------------------------------ TABLE KARDEX ----------------------------*/

const kardexScheme = sequelize.define(
    'kardex',
    {
      id_kardex: {
        type: DataTypes.UUID,
        defaultValue: DataTypes.UUIDV4,
        primaryKey: true,
      },

      cod_product: {
        type: DataTypes.STRING,
        allowNull: false,
      },

      detail: {
        type: DataTypes.ENUM('Entrada', 'Venta', 'Devolución', 'Ajuste'),
```

```
    allowNull: false,
},

unit_value: {
    type: DataTypes.FLOAT,
    allowNull: false,
},

quantity_entry: {
    type: DataTypes.INTEGER,
    allowNull: true,
},

total_entry: {
    type: DataTypes.FLOAT,
    allowNull: true,
},

quantity_exit: {
    type: DataTypes.INTEGER,
    allowNull: true,
},

total_exit: {
    type: DataTypes.FLOAT,
    allowNull: true,
},

quantity_balance: {
    type: DataTypes.INTEGER,
    allowNull: false,
},
```

```
        total_balance: {
            type: DataTypes.FLOAT,
            allowNull: false,
        },

        movement_type: {
            type: DataTypes.ENUM('Entrada', 'Salida'),
            allowNull: false,
        },
    },
    {
        tableName: 'kardex',
        timestamps: true,
    }
)

/* ------------------------------ TABLE ECONOMIC ----------------------------*/

const economicScheme = sequelize.define(
    'economic',
    {
        id_economic: {
            type: DataTypes.UUID,
            defaultValue: DataTypes.UUIDV4,
            primaryKey: true,
        },
        iva: {
            type: DataTypes.DECIMAL,
            allowNull: false,
        },
        profit_percentage: {
```

```
            type: DataTypes.DECIMAL,
            allowNull: false,
        },
    },
    {
        tableName: 'economics',
        timestamps: true,
    }
)

/* ------------------------------ TABLE LOGS ------------------------------*/

const logsScheme = sequelize.define(
    'logs',
    {
        id_log: {
            type: DataTypes.UUID,
            defaultValue: DataTypes.UUIDV4,
            primaryKey: true,
        },

        action: {
            type: DataTypes.STRING,
            allowNull: false,
        },

        targetType: {
            type: DataTypes.ENUM('usuario', 'producto', 'venta'),
            allowNull: false,
        },

        id_user_fk: {
```

```
            type: DataTypes.UUID,
            allowNull: false,
            references: {
                model: userScheme,
                key: 'id_user',
            },
        },

        target_id: {
            type: DataTypes.UUID,
            allowNull: true,
        },

        description: {
            type: DataTypes.TEXT,
            allowNull: true,
        },
    },
    {
        tableName: 'logs',
        timestamps: true,
    }
)

logsScheme.belongsTo(userScheme, {
    foreignKey: 'id_user_fk',
    targetKey: 'id_user',
})

logsScheme.belongsTo(userScheme, {
    foreignKey: 'target_id',
    constraints: false,
```

```
    scope: {
        targetType: 'usuario',
    },
})

logsScheme.belongsTo(productScheme, {
    foreignKey: 'target_id',
    constraints: false,
    scope: {
        targetType: 'producto',
    },
})

export {
    userScheme,
    rolScheme,
    otpCodeScheme,
    productScheme,
    additionalDetailProductScheme,
    serviceDigitalScheme,
    salesScheme,
    logsScheme,
    settingsScheme,
    economicScheme,
    kardexScheme,
    suppliersScheme,
    billProductScheme,
}
```

Construcción de los modelos

Se crearon los modelos para cada una de las entidades del dominio del negocio, como ventas, productos, clientes, etc. Estos modelos se encargan de proporcionar

una capa de abstracción y de interactuar con la base de datos, cada modelo usa el esquema de datos para cada entidad a la que pertenece o maneja y realizar operaciones CRUD.

Algoritmo del modelo para la gestión del inventario

```
import { productScheme } from '../schema/schemes.js'

export class ProductModel {
   static async getAllProducts() {
      try {
         const products = await productScheme.findAll({
            order: [['createdAt', 'DESC']],
         })
         return products
      } catch (error) {
         throw new Error(error)
      }
   }

   static async getAllProductsActive() {
      try {
         const products = await productScheme.findAll({
            where: {
               active: true,
            },
            order: [['createdAt', 'DESC']],
         })
         return products
      } catch (error) {
         throw new Error(error)
      }
   }
```

```
static async getByIdProductsActive(codeBar) {
    try {
        const products = await productScheme.findAll({
            where: {
                code_bar: codeBar,
                active: true,
            },
        })
        return products
    } catch (error) {
        throw new Error(error.message)
    }
}

static async getProductById(id) {
    try {
        const product = await productScheme.findByPk(id)
        return product
    } catch (error) {
        throw new Error(error)
    }
}

static async createProduct(newData) {
    try {
        const newProduct = await productScheme.create(newData)
        return newProduct
    } catch (error) {
        throw new Error(error)
    }
}
```

```
static async updateProduct(id, newData) {
    try {
        const product = await productScheme.findByPk(id)
        await product.update(newData)
        return product
    } catch (error) {
        throw new Error(error)
    }
}

static async changeActiveProduct(id, active = false) {
    try {
        const product = await productScheme.findByPk(id)

        // Solo actualizar si el estado es diferente al actual
        if (product.active !== active) {
            product.active = active
            await product.save() // Guardar solo si se realizó un cambio
        }

        return true
    } catch (error) {
        // Lanzar error para manejar en la capa superior
        throw new Error(`Error actualizando producto: ${error.message}`)
    }
}

static async deleteProduct(id) {
    try {
        const product = await productScheme.findByPk(id)
        await product.destroy()
    } catch (error) {
```

```
            throw new Error(error)
        }
    }

    static async findByCodeProduct(code_product) {
        try {
            const product = await productScheme.findOne({ where: { code_product } })
            return product
        } catch (error) {
            throw new Error(error)
        }
    }

    static async findByCodePrimary(code_primary) {
        try {
            const product = await productScheme.findOne({ where: { code_primary } })
            return product
        } catch (error) {
            throw new Error(error)
        }
    }

    static async findByCodeSecondary(code_secondary) {
        try {
            const product = await productScheme.findOne({ where: { code_secondary } })
            return product
        } catch (error) {
            throw new Error(error)
        }
```

```
}

static async findByCodeBar(code_bar) {
    try {
        const product = await productScheme.findOne({ where: { code_bar } })
        return product
    } catch (error) {
        throw new Error(error)
    }
}

static async findLastProductCode() {
    try {
        const lastProduct = await productScheme.findOne({
            order: [['createdAt', 'DESC']],
            attributes: ['code_product'],
        })
        return lastProduct ? lastProduct.code_product : null
    } catch (error) {
        throw error
    }
}

static async getProductAmount(productId) {
    try {
        const product = await productScheme.findOne({
            where: { id_product: productId },
            attributes: ['amount'],
        })

        return product.amount
    } catch (error) {
```

```
            throw error
        }
    }
}
```

Controladores de los modelos

Los controladores se crearon para cada uno de los modelos, se encargan de recibir las solicitudes de la API REST y de interactuar con los modelos para realizar operaciones CRUD ya establecidas, definiendo métodos para cada una de las operaciones e implementando la lógica de negocio necesaria. Cada controlador recibe dos parámetros de tipo Request y Response para manejar correctamente los errores y las respuestas del servidor.

Algoritmo del controlador del Kardex

```
import { KardexModel } from '../models/kardex.model.js'
import { ProductModel } from '../models/products.models.js'
import { handleError } from '../utils/errorHandler.utils.js'
import { SuppliersModel } from '../models/suppliers.models.js'

const KardexController = {
    async getAllKardex(req, res) {
        try {
            const kardex = await KardexModel.getAllKardex()

            const latestKardexMap = new Map()

            kardex.forEach(kardexItem => {
                const productFk = kardexItem.product_fk
                if (
                    !latestKardexMap.has(productFk) ||
                    new Date(kardexItem.createdAt) > new
Date(latestKardexMap.get(productFk).createdAt)
                ) {
                    latestKardexMap.set(productFk, kardexItem)
```

```
            }
        })

        // Convierte el mapa en un array de los registros más recientes
        const latestKardexArray = Array.from(latestKardexMap.values())

        // Obtén los detalles del producto para cada registro más reciente
        const kardexWithProducts = await Promise.all(
            latestKardexArray.map(async kardexItem => {
                const product = await
ProductModel.getProductById(kardexItem.product_fk)
                return {
                    ...kardexItem.toJSON(),
                    product: product ? product.toJSON() : null,
                }
            })
        )

        // Envía la respuesta con los registros más recientes y sus productos
        res.status(200).json(kardexWithProducts)
    } catch (error) {
        handleError(res)
    }
},

async getKardexById(req, res) {
    try {
        const { id } = req.params
        console.log(id)

        const kardexRecords = await KardexModel.getByProductId(id)
        if (!kardexRecords)
```

```
            return res.status(404).json({ message: 'No se encontraron registros de Kardex para el producto.' })

        // Obtén los detalles del producto y el proveedor para cada registro de Kardex
        const kardexWithDetails = await Promise.all(
          kardexRecords.map(async kardexItem => {
            const product = await ProductModel.getProductById(kardexItem.product_fk)
            const supplier = await SuppliersModel.getSuppliersById(product.supplier_fk)
            return {
              ...kardexItem.toJSON(),
              product: product ? product.toJSON() : null,
              supplier: supplier ? supplier.toJSON() : null,
            }
          })
        )

        console.log(kardexWithDetails)
        res.status(200).json(kardexWithDetails)
      } catch (error) {
        console.log(error)
        handleError(res)
      }
    },
}

export { KardexController }
```

Creación de rutas para la API REST

Las rutas para la API REST se crearon utilizando el framework *express*, estas rutas se crearon para cada una de las operaciones CRUD y se asociaron con los controladores correspondientes. Todas las rutas usan un método HTTP y se contienen en un paquete denominado **routes** y es exportado a un archivo general **routes.js** que maneja las rutas generales y las sub-rutas según las peticiones al servidor a través de los métodos HTTP.

Algoritmo de creación de rutas para la entidad Productos

```
import multer from 'multer'
import { Router } from 'express'
import { Auth } from '../middlewares/auth.middleware.js'
import { ProductController } from '../controllers/product.controller.js'
import { isAdmin, isAdminEmployee } from '../middlewares/admin.middleware.js'

const router = Router()
const uploadXML = multer({ dest: 'uploads/xml/' })
const uploadXLSX = multer({ dest: 'uploads/xlsx/' })

router.get('/', Auth, isAdmin, ProductController.getAllProducts)
router.get('/active',Auth,isAdminEmployee,
ProductController.getAllProductsActive)
router.get('/:id', Auth, isAdmin, ProductController.getProductsById)
router.delete('/:id', Auth, isAdmin, ProductController.deleteProduct)
router.put('/active/:id', Auth, isAdmin, ProductController.activeProduct)
router.put('/update/:id', Auth, isAdmin, ProductController.updateProduct)
router.put('/desactive/:id', Auth, isAdmin, ProductController.desactiveProduct)
router.post('/uploadXML', uploadXML.single('file'), Auth, isAdmin,
ProductController.uploadFile)
router.post('/upload-code-bar', Auth, isAdmin, uploadXLSX.single('file'),
ProductController.uploadCodeBar)

export default router
```

En el archivo server.js se implementa las rutas con el sistema importando el archivo routes.js que maneja el paquete de rutas.

Algoritmo de implementacion de rutas

```
import express from 'express'

import authRoutes from './routes/auth.routes.js'

import usersRoutes from './routes/users.routes.js'

import profileRoutes from './routes/profile.routes.js'

import productRoutes from './routes/product.routes.js'

import settingsRoutes from './routes/settings.routes.js'

import economicRoutes from './routes/economic.routes.js'

import suppliersRoutes from './routes/suppliers.routes.js'

import bill_productsRoutes from './routes/bill_products.routes.js'


const router = express.Router()

// Endpoints públicos - autenticación

router.use('/auth/', authRoutes)

// Endpoints privados - autenticación

router.use('/users/', usersRoutes)

router.use('/profile/', profileRoutes)

router.use('/settings/', settingsRoutes)

router.use('/economic/', economicRoutes)

router.use('/products/', productRoutes)

router.use('/suppliers/', suppliersRoutes)
```

```
router.use('/bill-product', bill_productsRoutes)

// Endpoint para verificar el estado del servidor

router.get('/status', (req, res) => {

   res.json()

})

export default router
```

5.3.4. Desarrollo del sprint 4. Programación Frontend

3.1.7.22. Planificación del sprint 4

El desarrollo del sprint 4 se basa en la programación del frontend del sistema online para la gestión de ventas del local comercial Isabelita. El sprint tiene una duración de 40 días, con un promedio de 7 horas de trabajo diarias.

El alcance de este sprint es desarrollar la interfaz de usuario de todo el sistema, así como la implementación de la lógica de negocio a nivel frontend (vistas del MVC) y la integración con la API del sistema y todo el backend, realizado durante el sprint 3.

Durante este sprint se identificaron tareas que se debían desarrollar para la creación de las diferentes interfaces de usuario: inicio de sesión (HU1), gestión de usuarios (HU2, HU3, HU8), registro de productos y gestión de inventario (HU4), consulta de inventario de productos (HU5), registro y gestión de ventas (HU6), registro y gestión de clientes de clientes (HU7), generación de reportes (HU9), búsqueda y consulta de registros (HU10), análisis de ventas (HU11), consulta de ventas realizas, facturadas y no facturadas (HU12) y visualización de alertas (HU13), separando las interfaces a mostrar según el tipo de usuario, los permisos y roles a los que tiene acceso.

Al igual que en el backend se priorizarán las tareas con mayor riesgo en desarrollo y mayor prioridad, como la creación de la interfaz de usuario para la gestión de ventas y la gestión de usuarios.

Para desarrollar estas interfaces y sus funcionalidades se tomó en cuenta la programación basada en la creación de componentes reutilizables para la aplicación,

evitando la creación de código repetitivo que se traduce en el rendimiento del sistema.

3.1.7.23. Tareas del sprint 4

Tabla 25. Tareas del sprint 4 - Desarrollo Frontend.

ID	Rol	Tareas	Tiempo de estimación (horas)
1	Desarrollador.	Configuración de entorno de programación e instalación de dependencias.	2
2	Desarrollador.	Crear la estructura de carpetas para las vistas del MVC.	1
3	Desarrollador.	Desarrollar la vista de inicio de sesión.	10
4	Desarrollador	Crear la vista principal del sistema On-line para el usuario “propietario”.	15
5	Desarrollador.	Realizar la vista de gestión de usuarios para el usuario “propietario”.	17
6	Desarrollador.	Crear la vista principal del sistema On-line para el usuario “vendedor”.	15
7	Desarrollador.	Crear la vista para el registro de productos en el inventario del usuario “propietario”.	19
8	Desarrollador.	Desarrollar componentes para la consulta de los productos del inventario del usuario “propietario”.	10
9	Desarrollador.	Crear componentes de consulta de inventario para el usuario “vendedor”.	13
10	Desarrollador.	Crear la vista de registro y gestión de ventas para el usuario “vendedor”.	20

11	Desarrollador.	Realizar la vista de componentes para el registro y gestión de clientes del usuario “vendedor”.	15
12	Desarrollador.	Desarrollar los componentes para la visualización de reportes de ventas del usuario “propietario”.	20
13	Desarrollador.	Crear componentes de búsqueda y consulta de registros de clientes, proveedores y ventas del usuario “propietario”.	20
14	Desarrollador.	Realizar la vista y los componentes para representar el análisis de ventas del usuario “propietario”.	20
15	Desarrollador	Desarrollar la vista de los ajustes del sistema online del usuario “propietario”.	15
16	Desarrollador.	Crear componente de consulta y visualización de ventas realizadas para el usuario “contador”.	7
17	Desarrollador.	Crear componentes de visualización de alertas y notificaciones del sistema.	8
18	Desarrollador.	Realizar la vista y los componentes para las configuraciones del sistema del usuario “administrador”.	12
19	Desarrollador.	Realizar la integración de las vistas de páginas del sistema On-line con la API REST.	10
20	Desarrollador.	Configurar el servidor web para la integración del frontend con el backend.	10

21	Desarrollador, Product Owner.	Realizar revisiones de la estructura del código, el diseño de las interfaces y la integración con el backend.	8
22	Desarrollador.	Refactorizar el código si es necesario para asegurar un mejor rendimiento del sistema On-line.	10
23	Desarrollador, Product Owner, Stakeholder Propietario.	Realizar una retroalimentación del sistema y sus funcionalidades con el propietario para asegurarse de que el producto se ajuste a sus necesidades y expectativas.	3
		Total	**280**

3.1.7.24.Ejecución del sprint 4

Tabla 26. Scrum diario del sprint 4 - Desarrollo frontend.

Día	Miembro del Equipo	Tarea Realizada	Impedimentos	Acciones a seguir
1	Juan Aldaz, Angelo García	Se configuró el entorno de programación usando el framework node js.		Configurar la biblioteca React JS para crear las vistas de las páginas del sistema On-line.
1	Juan Aldaz, Angelo García	Se instalaron las dependencias necesarias para el proyecto.	Algunas dependencias no son compatibles con la versión de node js instalada.	Instalar la versión compatible de las dependencias con la versión actual de node js.
2	Juan Aldaz,	Se creó la estructura de carpetas para		Verificar las organización de

	Angelo García	organizar las vistas del sistema, siguiendo el patrón MVC.		las carpetas separando las vistas de los diferentes usuarios existentes en el sistema.
2	Angelo García	Se diseñó el formulario de inicio de sesión con campos para ingresar las credenciales del usuario.		Crear una vista y un componente de formulario para recuperar la contraseña del usuario.
3	Angelo García	Se desarrolló la lógica de inicio de sesión, validando los datos del usuario en los campos de entrada y redirigiendo a la página correspondiente.		Implementar la vista de la página de inicio de sesión con la API.
4	Juan Aldas	Se implementó la vista y el componente formulario para recuperar la contraseña a través de un botón de acceso tipo enlace.	La lógica de negocio a nivel del backend no existía para esta funcionalidad.	Crear la lógica de negocio para la recuperación de contraseña e integrarla en la API del backend.
5	Angelo García	Se creo la vista de la página Inicio del usuario “propietario”.		Diseñar el layout de la página Inicio asegurando la

				experiencia de usuario.
6	Angelo García	Se creó el componente del menú principal para el usuario "propietario", incluyendo las opciones de gestión de usuarios, inventario, ventas, proveedores, clientes, Kardex y ajustes.		Implementar el componente menú en todas las páginas del usuario “propietario”.
7, 8	Angelo García	Se creo componentes de resumen de Kardex y resúmenes de ventas para la vista principal o dashboard del usuario "propietario.	Falta de consenso sobre la herramienta de visualización. Complejidad en la lógica de negocio para la integración de datos con las gráficas.	Seleccionar la herramienta de visualización de gráficos y tablas. Implementar los componentes en un área de contenido de la página principal.
9	Juan Aldas	Se creo el componente pie de página (footer) para las vistas del usuario “propietario”		Implementar el componente footer en todas las páginas del usuario “propietario”.
10	Juan Aldas	Se diseñó la vista de gestión de usuarios, incluyendo opciones para agregar, editar,		Crear componentes para la vista usuarios.

		inhabilitar y consultar usuarios.	
11	Juan Aldas	Se creó el componente botón para crear un nuevo usuario y el componente tabla para listar los usuarios existentes.	Crear componentes botones para inhabilitar y editar cada usuario de existente.
12	Juan Aldas	Se desarrolló el componente formulario para el registro de los datos del usuario, el mismo que también es reutilizado para la edición de los datos.	Integrar con la API del backend.
13	Angelo García	Se desarrolló la vista la gestión y registro del inventario del usuario “propietario”.	Diseñar los componentes de consulta de productos y servicios.
14	Angelo García	Se creó el componente de ingreso de la factura xml para el registro de los productos en el inventario, el componente tabla para el listado de los productos y servicios,	Validar el formato de la factura sea de tipo xml.

		y el componente para la búsqueda y consulta.		
15	Angelo García	Se creo el componente formulario para la edición del producto o servicio.	Errores en la validación de datos al intentar editar.	Corregir las validaciones tanta a nivel del backend como fronted.
16, 17	Angelo García	Se creo la vista principal del usuario “vendedor”, en este caso la página para registrar las ventas.	No existían los componentes menú y footer para las páginas del usuario “vendedor”.	Crear componentes menú y footer para las páginas del usuario “vendedor”.
18	Juan Aldas	Se creó el componente menú para el usuario "vendedor", incluyendo las opciones de registro de ventas, gestión de clientes, gestión de ventas y notificaciones.		Implementar los componentes menú y footer en las páginas del usuario “vendedor”.
19	Juan Aldas	Se desarrolló los componentes de búsqueda y selección de productos o servicios para generar la venta, el componente para el listado de productos		Implementar los componentes en la vista de registro de ventas.

		de seleccionados y el componente para los detalles de la compra.	
20	Juan Aldas	Se creó el componente de registro del cliente para la vista de ventas.	Diseñar el componente de visualización de la factura o comprobante de la compra realizada.
21	Angelo García	Se diseñó la vista de gestión de ventas del usuario “vendedor”, incluyendo componentes para un listado de ventas realizadas y opciones para ver detalles y descargar el comprobante.	Definir el flujo de usuario para la gestión de ventas. Implementar la funcionalidad de búsqueda en la vista de gestión de ventas.
22	Juan Aldas	Se desarrollo la vista para la gestión de los clientes para el usuario “vendedor”, los componentes para el listado y opciones para consultar sus detalles.	Implementar la funcionalidad de búsqueda en la vista de gestión de clientes.
23	Juan Aldas	Se desarrollo la vista para la gestión de los clientes para el usuario “propietario”,	Reutilizar componentes de los clientes del

		los componentes para el listado y opciones para consultar sus detalles.		usuario “propietario”.
24	Angelo García	Se desarrolló la vista para la gestión de proveedores del usuario “propietario”, así como los componentes para la visualización del listado y búsqueda y consulta.		
25	Juan Aldas	Se implementó un componente gráfico de líneas para representar la cantidad de clientes mensuales en las dos vistas de los dos usuarios.		Reutilizar el componente grafico de line del usuario “propietario”.
26	Angelo García	Se creó la vista para la representación del Kardex y los componentes para el listado y opciones de visualización de detalles para el usuario “propietario”.		Crear el componente para la visualización de detalles del Kardex.
27	Juan Aldas	Se diseñó la vista de ajustes del sistema online, incluyendo	Falta de aclaración del diseño de la vista	Definir el diseño de la vista de

		opciones para cambiar la contraseña, actualizar la información del usuario y realizar ajustes.	de los ajustes del sistema.	ajustes del sistema online.
28	Angelo García	Se implementó una vista para el usuario contador.		Crear componentes de consulta, visualización de ventas realizadas y componentes para visualizar la factura.
29, 30	Juan Aldas	Se diseñaron los componentes para la visualización de alertas y notificaciones del sistema.	Problemas con la visualización de las notificaciones y alertas en tiempo real debido a problemas con la integración en el backend.	Definir el diseño de las alertas y notificaciones. Verificar el problema en la API del backend para corregir las alertas y notificaciones.
31, 32	Angelo García	Se desarrollo la vista para el usuario "administrador" con opciones para realizar las copias de seguridad del sistema y algunas configuraciones adicionales.		

33 - 40	Angelo García, Juan Aldas.	Se verificó que todas las vistas y sus componentes se integren correctamente con el backend del sistema.	Errores en la integración del backend y el frontend de las notificaciones y alertas en tiempo real por problemas de sockets.	Realizar correcciones y optimización de código para garantizar la integración de todas las vistas con el backend.

5.3.4.1.1. Resultados

Los resultados del sprint 4 abarca el desarrollo del frontend de todo el sistema para la gestión de ventas, las vistas y componentes de las páginas de los usuarios "vendedor" como registro de ventas, gestión de ventas y gestión de clientes, "propietario" como como inicio, inventario, proveedores, clientes, usuarios, Kardex y "administrador" que maneja las copias de seguridad y configuraciones del sistema.

5.3.4.1.2. Desarrollo del frontend

Para el desarrollo del frontend del sistema se utilizaron Node JS, React JS, y Tailwind CSS. React JS, la biblioteca de JavaScript permitió construir interfaces de usuario interactivas con su enfoque de componetización, mientras que Node JS, como entorno de ejecución, facilitó la creación de una aplicación web completa. Tailwind CSS, el framework de diseño de CSS, se utlizo para el estilo visual, moderno y consistente, aprovechando su enfoque de utilidad y sus clases predefinidas.

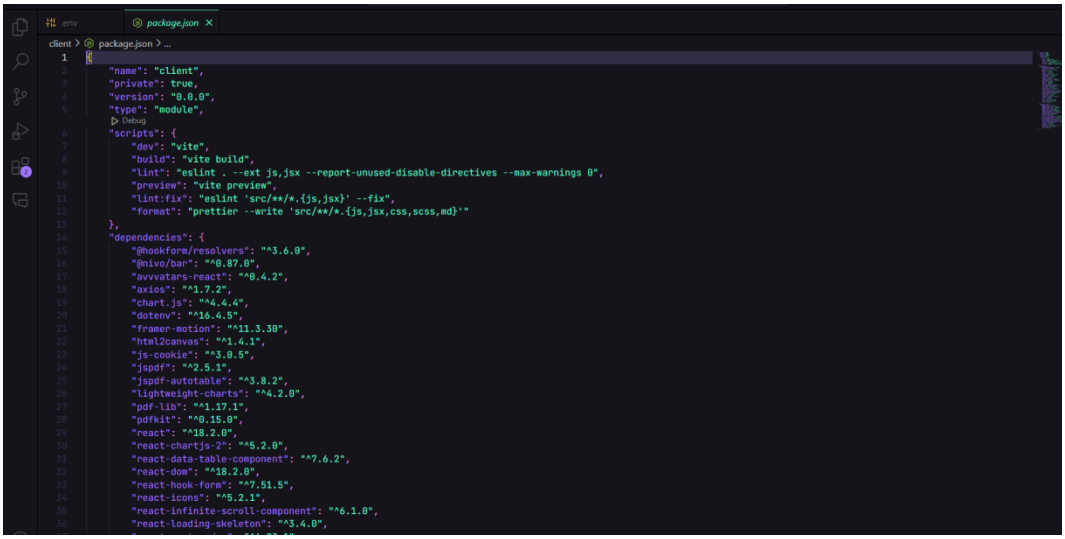


Gráfico 52. Instalación de dependencias de node.js para el desarrollo del frontend.

Desarrollo de la página principal del usuario "vendedor" – página de registro de ventas

La página principal del usuario "vendedor" fue diseñada y desarrollada con el objetivo de facilitar el proceso de registro de ventas. Esta página se convirtió en el punto de acceso principal para el vendedor, brindándole una interfaz intuitiva y eficiente para gestionar las transacciones comerciales. La página integra diferentes componentes que permiten al vendedor realizar las siguientes acciones:

- **Búsqueda y selección de productos o servicios:** El vendedor puede buscar y seleccionar los productos o servicios que desea incluir en la venta, utilizando un sistema de búsqueda y filtrado por código de barras y nombre que facilita la localización rápida de los productos.
- **Listado de productos seleccionados:** La página muestra un listado de los productos o servicios que el vendedor ha seleccionado para la venta, incluyendo información relevante como el nombre, la descripción, el precio y la cantidad.
- **Detalles de la compra:** Se proporciona información detallada sobre la compra, incluyendo el total de la venta, el IVA aplicado y otros.
- **Registro del cliente:** El vendedor puede registrar los datos del cliente para cada venta.

- **Visualización de la factura o comprobante:** La página genera una factura o comprobante de compra que se puede visualizar y descargar, proporcionando un registro formal de la transacción.

Algoritmo de llamada al componente de la vista principal del usuario "vendedor"

```
import React from 'react'
import { DashboardLayout } from '../../layouts/employee/DashboardLayout'

function DashboardEmployeePage() {
   return <DashboardLayout />
}

export default DashboardEmployeePage
```

Desarrollo de la página principal del usuario "propietario"

La página principal del usuario "propietario" es un tablero o dashboard que ofrece una visión general del negocio. Presenta tarjetas con información resumida sobre usuarios, inventario, ventas y clientes, incluyendo porcentajes de cambio. También incluye componentes para visualizar el Kardex, el balance y los productos más vendidos. La página está diseñada para proporcionar una visión rápida y fácil de entender del estado del negocio.

Algoritmo de maquetación de la página principal del usuario "propietario"

```
import { Toaster } from 'sonner'
import { Card } from '../../components/ui/cards/Card'
import { useUsersAll } from '../../redux/hooks/useUsers'
import { Kardex } from '../../components/ui/cards/Kardex'
import { Balance } from '../../components/ui/cards/Balance'
import { Navbar } from '../../components/ui/navbars/Navbar'
import { Spinner } from '../../components/ui/loaders/spinner'
import { useProductsAll } from '../../redux/hooks/useProducts'
import { TopProducts } from '../../components/ui/cards/TopProduct'
import { Footer_alt } from '../../components/ui/footers/Footer_alt'
```

```
import { LuCircleDollarSign, LuShoppingBasket, LuUser, LuUsers } from 'react-icons/lu'

const DashboardLayout = () => {
  const { users, loading, error } = useUsersAll()
  const { products } = useProductsAll()

  const getPercentageColor = percentage => {
    const value = parseFloat(percentage)
    return value >= 5 ? 'text-green-600' : 'text-red-600'
  }

  const getDirection = percentage => {
    const value = parseFloat(percentage)
    return value >= 5 ? 'up' : 'down'
  }

  return (
    <>
      <div className='mx-auto max-w-7xl'>
        <Navbar />
      </div>

      {loading ? (
        <Spinner />
      ) : (
        <main className='mx-auto max-w-7xl py-20'>
          <div className='flex flex-col gap-4 lg:flex-row'>
            <div className='flex w-full flex-col lg:w-3/4'>
              <div className='grid grid-cols-2 gap-6 md:grid-cols-4 lg:col-span-3'>
                <Card
```

```
            icon={LuUser}
            title='Usuarios'
            count={users.length}
            percentage='8%'
            percentageColor={getPercentageColor('8%')}
            direction={getDirection('8%')}
          />
          <Card
            icon={LuShoppingBasket}
            title='Inventario'
            count={products.length}
            percentage='16%'
            percentageColor={getPercentageColor('16%')}
            direction={getDirection('16%')}
          />
          <Card
            icon={LuCircleDollarSign}
            title='Ventas'
            count='146'
            percentage='2%'
            percentageColor={getPercentageColor('2%')}
            direction={getDirection('2%')}
          />
          <Card
            icon={LuUsers}
            title='Clientes'
            count='78'
            percentage='13%'
            percentageColor={getPercentageColor('13%')}
            direction={getDirection('13%')}
          />
        </div>
```

```
                    <div className='w-full pt-6 lg:w-full'>
                        <Kardex />
                    </div>
                </div>

                <div className='flex w-full flex-col gap-y-4 lg:w-1/4'>
                    <Balance />
                    <TopProducts />
                </div>
            </div>
        </main>
    )}

    <Footer_alt />

    <Toaster richColors position='bottom-right' />
  </>
 )
}

export { DashboardLayout }
```

Desarrollo de la página de gestión de ventas del usuario “propietario”

La página de gestión de ventas del usuario “propietario” es una lista de ventas con opciones de filtrado y descarga. Muestra una tabla con información sobre cada venta, incluyendo el ticket, el cliente, el vendedor, el total, la fecha y opciones para descargar la factura en PDF y ver detalles.

Lógica de programación de la página de gestión de ventas del usuario “propietario”

```
import { Toaster } from 'sonner'
import { BiCog } from 'react-icons/bi'
import { dateDetails } from '../../utils/date.util'
```

```
import { filterData } from '../../utils/search.util'
import { useState, useEffect, useMemo } from 'react'
import { Table } from '../../components/ui/table/Table'
import { Navbar } from '../../components/ui/navbars/Navbar'
import { dateFormater } from '../../utils/dateFormater.util'
import { useSettingsAll } from '../../redux/hooks/useConfig'
import { Spinner } from '../../components/ui/loaders/spinner'
import { getAllSaleRequest } from '../../services/api/sale.api'
import { SearchBar } from '../../components/ui/inputs/SearchBar'
import { Footer_alt } from '../../components/ui/footers/Footer_alt'
import { generatePDF } from '../../components/ui/molecules/pdf/generatePDF'
import { SaleDetailsModal } from '../../components/ui/modal/SaleDetailsModal'
import { LuCalendar, LuDownload, LuEye, LuFile, LuTag, LuUser, LuUsers }
from 'react-icons/lu'
import { Button } from '../../components/ui/atoms/button'

const SalesLayout = () => {
const { settings } = useSettingsAll()
const [sales, setSales] = useState([])
const [search, setSearch] = useState('')
const [error, setError] = useState(null)
const [loading, setLoading] = useState(false)
const [selectedSale, setSelectedSale] = useState(null)
const [modalOpen, setModalOpen] = useState(false)

useEffect(() => {
        const fetchSales = async () => {
                setLoading(true)
                try {
                        const data = await getAllSaleRequest()
                        setSales(data)
                } catch (error) {
```

```
                setError('Error al cargar las ventas')
            } finally {
                setLoading(false)
            }
        }
        fetchSales()
    }, [])

    const handleSearch = value => setSearch(value.toLowerCase())

    const filteredData = useMemo(() => {
        return filterData(sales, search, ['id_bill_sale'])
    }, [sales, search])

    const handleViewDetails = sale => {
        setSelectedSale(sale)
        setModalOpen(true)
    }

    const handleCloseModal = () => {
        setSelectedSale(null)
        setModalOpen(false)
    }

    const companyInfo = {
        ruc: settings?.ruc,
        phone: settings?.phone,
        email: settings?.email,
        imagen: settings?.image,
        address: settings?.address,
        property_name: settings?.property_name,
    }
```

```
const title = 'Reporte - Ventas'
const tableHead = [['Ticket', 'Cliente', 'Vendedor', 'Total', 'Fecha']]
const tableBody = sales.map(sale => [
        sale.id_ticket,
        sale.client.names,
        sale.employee.username,
        `$${sale.total.toFixed(2)}`,
        dateDetails(sale.createdAt),
])

// Función para manejar el clic del botón
const handleDownloadReport = () => generatePDF(title, tableHead, tableBody, companyInfo, 'portrait', `Reporte_Ventas`)

export { SalesLayout }
```

5.3.5. Desarrollo del sprint 5. Análisis y Planificación de pruebas

3.1.7.25. Planificación del sprint 5

El desarrollo del sprint 5 se centra en la planificación de las pruebas para el sistema online de gestión de ventas del local comercial Isabelita. El sprint tiene una duración de 5 días, con un promedio de 6 horas de trabajo diarias.

El objetivo principal de este sprint es definir la estrategia de pruebas, identificar los casos de prueba, crear un plan de pruebas detallado, establecer los criterios de aceptación y asignar responsables para la ejecución de las pruebas. Esto permitirá evaluar la calidad y funcionalidad del sistema, asegurando que cumpla con los requisitos y expectativas del proyecto.

3.1.7.26.Tareas del sprint 5

Tabla 27. Tareas del sprint 5 – Análisis y Planificación de pruebas.

ID	Rol	Tareas	Tiempo de estimación (horas)
1	Scrum Master	Realizar análisis de requisitos y diseño de pruebas.	4
2	Scrum Master	Crear cronograma de actividades de prueba.	10
3	Scrum Master	Definir estrategia de pruebas y objetivos.	6
4	Scrum Master	Establecer alcance y tipos de pruebas a realizar.	6
5	Desarrollador	Configurar entorno de prueba (hardware y software) para ejecutar las pruebas.	4
		Total	**30**

3.1.7.27.Ejecución del sprint 5

Tabla 28. Scrum diario del sprint 5 - Análisis y Planificación de pruebas.

Día	Miembro del Equipo	Tarea Realizada	Impedimentos	Acciones a seguir
1	Scrum Master	Se realizó el análisis de requisitos y diseño de pruebas.		Identificar los casos de prueba críticos y establecer prioridades
2	Scrum Master	Se creó el cronograma de		Establecer responsables para cada actividad de

		actividades de pruebas.		prueba, fijar tiempos límite y asigna recursos necesarios.
3	Scrum Master	Se definió estrategias de pruebas y objetivos.		Establecer criterios de aceptación.
4	Scrum Master	Se estableció el alcance de las pruebas y tipos de pruebas a realizar.	Falta de definición de tipos de pruebas a realizar.	Detallar los tipos de pruebas.
5	Desarrollador	Se configuró entorno de prueba (hardware y software).		Instalar software de prueba adicional y realizar pruebas de conexión con el servidor y el sistema de ventas.

3.1.7.28.Resultados

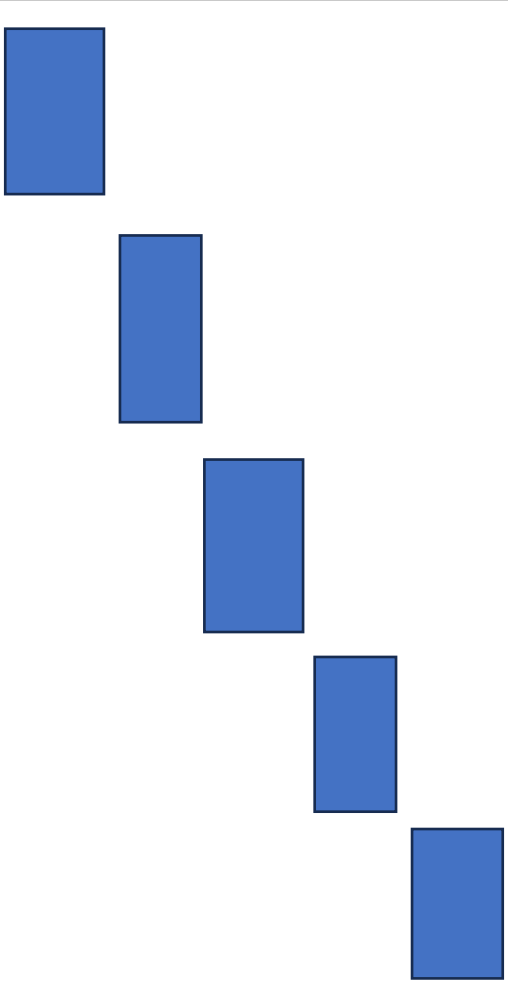
Los resultados del sprint 5 son el análisis y diseño de los casos de prueba, la definición del objetivo, alcance y tipos de pruebas que se aplicara al sistema para asegurar su calidad y correcto funcionamiento.

5.3.5.1.1. Objetivo de las pruebas

El objetivo de las pruebas es verificar que el sistema cumpla con los requisitos funcionales y no funcionales establecidos, y asegurar que el sistema sea estable, seguro y proporcione conformidad por parte de los usuarios.

5.3.5.1.2. Cronograma de actividades para las pruebas

Tabla 29. Cronograma de actividades para las pruebas.

N.º	Actividades de pruebas	Responsable	Tiempo 1 día 2 días 1 día 1 día 16 días
1	Análisis de requisitos y diseño de pruebas.	Juan Aldas	
2	Creación de cronograma de actividades de prueba.	Juan Aldas	
3	Definición de estrategia de pruebas y objetivos.	Juan Aldas	
4	Configuración del entorno de prueba.	Juan Aldas, Angelo García.	
5	Ejecución de pruebas y finalización.	Juan Aldas, Angelo García.	

5.3.5.1.3. Alcance de las pruebas

El alcance de las pruebas abarca todos los componentes del sistema, incluyendo la interfaz de usuario, la lógica de negocio, la base de datos y la integración del backend con el frontend. Las pruebas por realizar son:

Pruebas funcionales:

1. Pruebas de funcionalidad
2. Pruebas unitarias
3. Pruebas de integración

4. Pruebas de errores y excepciones

Pruebas no funcionales:

1. Pruebas de rendimiento
2. Pruebas de usabilidad

4.1.1.1.1. Tipos de pruebas

Pruebas de Funcionalidad

Tabla 30. Pruebas de funcionalidad.

Objetivo	**Descripción**	**Técnicas**	**Tiempo (horas)**
Verificar que el sistema realice las operaciones básicas correctamente.	Verificar que el sistema permita realizar operaciones básicas como ingresar a los diferentes módulos.	Ejecutar operaciones básicas del sistema.	8-10

Pruebas Unitarias

Tabla 31. Pruebas unitarias.

Objetivo	**Descripción de la prueba**	**Técnicas**	**Tiempo (horas)**
Verificar que cada componente del sistema (función o método) funcione correctamente.	Verificar que cada función o método del sistema funcione correctamente.	Ejecutar pruebas unitarias para cada función o método.	4-6

Pruebas de Integración

Tabla 32. Pruebas de integración.

Objetivo	**Descripción de la prueba**	**Técnicas**	**Tiempo (horas)**
Verificar que los módulos que tienen	Verificar que los procesos entre módulos se ejecuten	Ejecutar pruebas para verificar la integración de los	8-10

relación se integren correctamente.	según la lógica de negocio.	procesos entre módulos.	

Pruebas de Errores y Excepciones

Tabla 33. Pruebas de errores y excepciones.

Objetivo	**Descripción de la prueba**	**Técnicas**	**Tiempo (horas)**
Verificar que el sistema maneje correctamente los errores y excepciones.	Verificar que el sistema maneje correctamente errores y excepciones.	Ejecutar pruebas para verificar el manejo de errores y excepciones.	4-6

Pruebas de Rendimiento

Tabla 34. Pruebas de rendimiento.

Objetivo	**Descripción de la prueba**	**Técnicas**	**Tiempo (horas)**
Verificar que el sistema pueda manejar un gran volumen de transacciones sin afectar su rendimiento.	Verificar que el sistema pueda manejar un gran volumen de transacciones sin afectar su rendimiento.	Ejecutar pruebas de rendimiento para verificar el manejo de transacciones.	8-10

Pruebas de Usabilidad

Tabla 35. Pruebas de usabilidad.

Objetivo	**Descripción de la prueba**	**Técnicas**	**Tiempo (horas)**
Verificar que el sistema sea fácil de usar y accesible para los usuarios.	Verificar que el sistema sea fácil de usar y accesible para los usuarios.	Realizar pruebas de usabilidad con usuarios reales.	4-6

5.3.6. Desarrollo del sprint 6. Ejecución de las pruebas

3.1.7.29. Planificación del sprint 6

En la ejecución del sprint 6 se realizó la ejecución de las pruebas mencionadas durante el sprint 5, este sprint tiene una duración de 16 días con un aproximado de 6 horas diarias.

En cada tipo de prueba se realizan casos de pruebas en los diferentes aspectos del sistema para determinar errores presentes o el correcto funcionamiento de las funcionalidades.

3.1.7.30. Tareas del sprint 6

Tabla 36. Tareas del sprint 6 - Ejecución de las pruebas.

ID	Rol	Tareas	Tiempo de estimación (horas)
1	Desarrollador	Ejecutar pruebas de funcionalidad.	16
2	Desarrollador	Aplicar pruebas unitarias.	8
3	Desarrollador	Realizar pruebas de integración.	12
4	Desarrollador	Ejecutar pruebas de errores y excepciones.	10
5	Desarrollador	Ejecutar pruebas de rendimiento.	12
6	Desarrollador	Realizar pruebas de usabilidad.	12
7	Desarrollador	Documentar defectos y errores encontrados durante las pruebas del sistema.	5
8	Desarrollador	Realizar seguimiento y correcciones de los defectos y errores encontrados.	11
		Total	**86**

3.1.7.31.Ejecución del sprint 6

Tabla 37. Scrum diario del sprint 6 - Ejecución de las pruebas.

Día	Miembro del Equipo	Tarea Realizada	Impedimentos	Acciones a seguir
1	Juan Aldas	Se ejecutaron pruebas de funcionalidad.	Falta de datos de prueba.	Generar datos de prueba adicionales.
2	Angelo García	Se realizaron pruebas unitarias.		
3	Juan Aldas	Se ejecutaron pruebas de integración.	Falta de documentación de la API.	Solicitar documentación de la API al equipo de desarrollo.
4	Angelo García	Se aplicaron pruebas de errores y excepciones.	Falta de definición de casos de pruebas para este tipo de pruebas.	Definir los casos de prueba adecuados para este tipo de pruebas.
5	Angelo Garcia	Se realizaron pruebas de rendimiento.		
6	Juan Aldas	Se ejecutaron pruebas de usabilidad.	Falta de usuarios para realizar pruebas de usabilidad.	Crear usuarios ficticios adicionales para realizar pruebas de usabilidad.
7	Angelo Gracia, Juan Aldas	Se reportaron defectos y errores encontrados		

		durante las pruebas.		
8	Angelo García, Juan Aldas	Se realizó seguimiento y control de los defectos y errores reportados.	Falta de priorización de defectos.	Priorizar defectos y establecer tiempos límite para su corrección.

3.1.7.32. Resultados

Los resultados de este sprint es la ejecución de diferentes tipos de casos de pruebas en el sistema.

5.3.6.1.1. Pruebas de funcionalidad

Tabla 38. Caso de prueba 1 - Autentificación de usuario.

	Caso de prueba: Autentificación de usuario.				
No. Caso de prueba:	1	**Responsable:**		Juan Aldas	
	Aspectos que debe cumplir la prueba.				
No.	**Descripción**	**Resultado esperado**	**Resultado obtenido**	**Aprueba**	
				Si	**No**
1	Al iniciar el sistema online por primera vez la página que debe mostrar es la del login.	La página de inicio de sesión se muestra correctamente.	La página de inicio de sesión se muestra correctamente.	✓	
2	El formulario de inicio de sesión debe validar los	Mensaje de error de la validación en la interfaz cuando	Se muestra correctamente el mensaje de	✓	

	datos de entrada (cedula y contraseña)	los mensajes son incorrectos.	validación en la interfaz.		
3	Si el usuario y contraseña es correcto debe permitir el acceso a la página principal según el usuario que se autentifique.	El sistema debe mostrar la página de inicio según el tipo de usuario correctamente.	La página de inicio se muestra correctamente según el tipo de usuario.	✓	

Tabla 39. Caso de prueba 2 - Ingreso a la página principal del sistema.

	Caso de prueba: Ingreso a la página principal del sistema.				
No. Caso de prueba:	2	**Responsable:**		Juan Aldas	
	Aspectos que debe cumplir la prueba.				
No.	**Descripción**	**Resultado esperado**	**Resultado obtenido**	**Aprueba**	
				Si	**No**
1	Iniciar el sistema online por primera vez después de iniciar sesión.	La página principal del sistema se muestra correctamente.	La página principal del sistema se muestra correctamente.	✓	
2	Verificar la carga y visualización de la página principal.	La página principal del sistema se carga correctamente sin errores y se muestra con	La página principal del sistema se carga correctamente sin errores y se	✓	

		todos los elementos visuales esperados (logotipo, menú, dashboard, componentes de resúmenes, etc.).	muestra con todos los elementos visuales esperados.		

Tabla 40. Caso de prueba 3 – Ingreso al módulo de inventario.

	Caso de prueba: Ingreso al módulo de inventario.				
No. Caso de prueba:	3	**Responsable:**		Juan Aldas	
	Aspectos que debe cumplir la prueba.				
No.	**Descripción**	**Resultado esperado**	**Resultado obtenido**	**Aprueba**	
				Si	**No**
1	Acceder al módulo desde el menú en la opción inventario.	La pantalla de inventario se muestra correctamente.	La pantalla de inventario se muestra correctamente.	✓	
2	Verificar la carga y visualización de la pantalla de inventario	La pantalla de inventario se carga correctamente sin errores y se muestra con todos los elementos	La pantalla de inventario carga correctamente con todos los elementos esperados.	✓	

		visuales esperados (listado de productos y servicios, botones de acción y componente de búsqueda.).			

Tabla 41. Caso de prueba 4 - Ingreso al módulo de proveedores.

	Caso de prueba: Ingreso al módulo de proveedores.				
No. Caso de prueba:	4	**Responsable:**		Juan Aldas	
	Aspectos que debe cumplir la prueba.				
No.	**Descripción**	**Resultado esperado**	**Resultado obtenido**	**Aprueba**	
				Si	**No**
1	Acceder al módulo desde el menú en la opción proveedores.	La pantalla de proveedores se muestra correctamente.	La pantalla de proveedores se muestra correctamente.	✓	
2	Verificar la carga y visualización de la pantalla de proveedores.	La pantalla de proveedores se carga correctamente sin errores y se muestra con todos los elementos visuales	La pantalla de proveedores carga correctamente con todos los elementos esperados.	✓	

		esperados (listado de proveedores, botones de acción y componente de búsqueda.).			

Tabla 42. Caso de prueba 5 - Ingreso al módulo de gestión de ventas.

	Caso de prueba: Ingreso al módulo de gestión de ventas.				
No. Caso de prueba:	5	**Responsable:**		Juan Aldas	
	Aspectos que debe cumplir la prueba.				
No.	**Descripción**	**Resultado esperado**	**Resultado obtenido**	**Aprueba**	
				Si	**No**
1	Acceder al módulo desde el menú en la opción ventas.	La página de gestión de ventas se muestra correctamente.	La página de gestión de ventas se muestra correctamente.	✓	
2	Verificar la carga y visualización de la pantalla de ventas.	La pantalla de ventas carga correctamente sin errores y se muestra con todos los elementos visuales esperados (listado de ventas, botones	La pantalla de ventas carga correctamente con todos los elementos esperados.	✓	

		de acción y componente de búsqueda).			

Tabla 43. Caso de prueba 6 - Ingreso al módulo de usuarios.

	Caso de prueba: Ingreso al módulo de usuarios.				
No. Caso de prueba:	6	**Responsable:**		Juan Aldas	
	Aspectos que debe cumplir la prueba.				
No.	**Descripción**	**Resultado esperado**	**Resultado obtenido**	**Aprueba**	
				Si	**No**
1	Acceder al módulo de usuarios desde el menú de opciones.	La página de usuarios se muestra correctamente.	La página de usuarios se muestra correctamente.	✓	
2	Verificar la carga y visualización de la página de usuarios.	La página de usuarios se carga correctamente sin errores y se muestra con todos los elementos visuales esperados (listado de usuarios, botones de acción y componente de	La página de usuarios carga correctamente con todos los elementos esperados.	✓	

		búsqueda de usuarios).			

Tabla 44. Caso de prueba 7 - Ingreso al módulo de clientes.

	Caso de prueba: Ingreso al módulo de clientes.				
No. Caso de prueba:	7	**Responsable:**		Juan Aldas	
	Aspectos que debe cumplir la prueba.				
No.	**Descripción**	**Resultado esperado**	**Resultado obtenido**	**Aprueba**	
				Si	**No**
1	Ingresar al módulo de clientes desde el menú de opciones.	La página de clientes se muestra correctamente.	Se muestra correctamente la página de clientes.	✓	
2	Verificar la carga y visualización de la página de clientes.	La página de clientes se carga correctamente sin errores y se muestra con todos los elementos visuales esperados (listado de clientes, botones de acción y componente de búsqueda de clientes).	La página de clientes carga correctamente con todos los elementos esperados.	✓	

Pruebas Unitarias

Tabla 45. Caso de prueba 8 - Agregar productos al inventario.

	Caso de prueba: Agregar productos al inventario.				
No. Caso de prueba:	8	**Responsable:**		Angelo García	
Aspectos que debe cumplir la prueba.					
No.	**Descripción**	**Resultado esperado**	**Resultado obtenido**	**Aprueba**	
				Si	**No**
1	Ingresar al sistema como usuario “propietario”.	El propietario ingresa correctamente al sistema.	El propietario ingresa correctamente al sistema.	✓	
2	Acceder al módulo de inventario.	El módulo de inventario se muestra correctamente.	El módulo de productos se muestra correctamente.	✓	
3	Agregar nuevos productos con información válida desde una factura en formato XML.	Los productos se agregan correctamente al inventario.	Los productos se agregan correctamente al inventario.	✓	

Tabla 46. Caso de prueba 9 - Registrar nuevo proveedor.

	Caso de prueba: Registra nuevo proveedor.			
No. Caso de prueba:	9	**Responsable:**		Angelo García
Aspectos que debe cumplir la prueba.				
No.	**Descripción**			**Aprueba**

		Resultado esperado	Resultado obtenido		
				Si	No
1	Ingresar al sistema como usuario “propietario”.	El propietario ingresa correctamente al sistema.	El propietario ingresa correctamente al sistema.	✓	
2	Acceder al módulo de inventario.	El módulo de inventario se muestra correctamente.	El módulo de productos se muestra correctamente.	✓	
3	Agregar nuevos productos con información válida desde una factura en formato XML.	Los productos se agregan correctamente al inventario.	Los productos se agregan correctamente al inventario.	✓	

Tabla 47. Caso de prueba 9 - Registrar nuevo usuario

	Caso de prueba: Agregar productos al inventario.				
No. Caso de prueba:	9	**Responsable:**		Angelo García	
Aspectos que debe cumplir la prueba.					
No.	**Descripción**	**Resultado esperado**	**Resultado obtenido**	**Aprueba**	
				Si	**No**
1	Agregar nuevo usuario desde el módulo usuarios.	El botón de agregar nuevo usuario se muestra correctamente	El botón de agregar nuevo usuario se muestra correctamente.	✓	

		en la página de usuarios.			
2	Ingresar datos del usuario en el formulario de registro.	El formulario se muestra correctamente y permite el registro de los datos del nuevo usuario.	El formulario se muestra correctamente y permite el registro de los datos del usuario.	✓	

Tabla 48. Caso de prueba 10 - Registro de clientes.

	Caso de prueba: Registro de clientes.				
No. Caso de prueba:	10	**Responsable:**		Angelo García	
	Aspectos que debe cumplir la prueba.				
No.	**Descripción**	**Resultado esperado**	**Resultado obtenido**	**Aprueba**	
				Si	**No**
1	Registrar cliente desde el módulo de registro de ventas.	Se muestra la opción de facturación con datos.	Se muestra correctamente la opción de facturación con datos.	✓	
2	Agregar datos del cliente en el formulario de registro.	Se muestra el formulario y permite el ingreso de los datos del cliente.	Se muestra correctamente el formulario y permite el ingreso de los datos del cliente.	✓	

3	Facturar con datos del cliente.	Se registra al cliente de la venta.	Se registra correctamente al cliente de la venta.	✓	

Pruebas de integración

Tabla 49. Caso de prueba 11 - Realizar venta.

	Caso de prueba: Realizar venta.				
No. Caso de prueba:	11	**Responsable:**		Juan Aldas	
	Aspectos que debe cumplir la prueba.				
No.	**Descripción**	**Resultado esperado**	**Resultado obtenido**	**Aprueba**	
				Si	**No**
2	Acceder al módulo de registro de ventas.	El módulo de registro de ventas se muestra correctamente al iniciar sesión.	El módulo de registro de ventas se muestra correctamente al iniciar sesión.	✓	
	Buscar producto.	El sistema permite buscar producto o servicio por código de barras o por nombre.	El sistema permite buscar el producto de las dos formas.	✓	
	Agregar producto a la compra.	El sistema debe agregar debe agregar el	El sistema permite agregar el	✓	

		producto al detalle de la compra y permitir modificar la cantidad.	producto y modificar la cantidad.		
	Procesar la venta.	El sistema debe calcular el valor de la venta, permitir la facturación por consumidor final o factura con datos.	El valor de la venta se muestra correctamente y permite la facturación.	✓	
3	Registrar la venta.	La venta se registra correctamente.	La venta se registra correctamente.	✓	

Tabla 50. Caso de prueba 12 - Integración de Kardex.

	Caso de prueba: Integración de Kardex.				
No. Caso de prueba:	12	**Responsable:**		Juan Aldas	
	Aspectos que debe cumplir la prueba.				
No.	**Descripción**	**Resultado esperado**	**Resultado obtenido**	**Aprueba**	
				Si	**No**
1	Acceder a la página de Kardex.	El sistema debe mostrar el listado de Kardex.	El sistema muestra el listado de Kardex.	✓	

2	Acceder al listado del Kardex.	El listado de Kardex deberá permitir identificar fácilmente todas las entradas y salidas de los productos y servicios.	El listado del Kardex se muestra todas las entradas y salidas de los productos y servicios.	✓	
3	Acceder al historial de cambio de un producto en el Kardex.	El sistema permitirá ver el historial de cambio de cada producto o servicio en el Kardex.	El historial de cambio de cada producto o servicio en el Kardex se muestra correctamente.	✓	

Tabla 51. Caso de prueba 13 - Generación de reportes.

	Caso de prueba: Generación de reportes.				
No. Caso de prueba:	13	**Responsable:**		Juan Aldas	
	Aspectos que debe cumplir la prueba.				
No.	**Descripción**	**Resultado esperado**	**Resultado obtenido**	**Aprueba**	
				Si	**No**
1	Genera reporte de inventario.	El sistema debe permitir generar un reporte de inventario de los productos e inventarios	El sistema permite generar el reporte de inventario según el tipo	✓	

		según el tipo de estado, una fecha y los datos deseados.	de estado, una fecha y los datos deseados.		
2	Genera reporte de proveedores.	El sistema debe permitir generar un reporte de proveedores por fecha o de un mes y los datos deseados.	El sistema permite generar un reporte por fecha o de un mes y los datos deseados.	✓	
3	Genera reporte de ventas.	El sistema debe permitir generar un reporte de ventas por fecha, de un mes en específico y los datos deseados.	El sistema permite generar un reporte de ventas por fecha o de un mes en específico y los datos deseados.	✓	
4	Genera reporte de clientes.	El sistema debe permitir generar un reporte de clientes por fecha o de un mes y los datos deseados.	El sistema permite generar un reporte de clientes por fecha o de un mes en específico y los datos deseados.	✓	

5	Genera reporte de usuarios.	El sistema debe permitir generar un reporte de usuarios por fecha o de un mes y los datos deseados.	El sistema permite generar un reporte de usuarios por fecha o de un mes en específico y los datos deseados.	✓	
6	Generar reporte de Kardex.	El sistema debe permitir generar un reporte por cada registro de Kardex, según la fecha, de un mes en específico y los datos deseados.	El sistema permite generar un reporte por cada registro de Kardex, según la fecha, de un mes en específico y seleccionar los datos deseados.	✓	

Pruebas de errores y excepciones

Tabla 52. Caso de prueba 14 - No hay productos seleccionados para realizar la venta.

	Caso de prueba: No hay productos seleccionados para realizar una venta.		
No. Caso de prueba:	14	**Responsable:**	Angelo García.

Descripción:	Verificar que el sistema lance un mensaje de error si no hay productos para realizar una venta.				
Aspectos que debe cumplir la prueba.					
Precondiciones	**Pasos para ejecutar la prueba**	**Resultados esperados**	**Resultado obtenido**	**Aprueba**	
				Si	**No**
No debe haber productos agregados al detalle de la compra.	1. Iniciar sesión como vendedor. 2. Intentar procesar una venta sin productos agregados.	El sistema debe lanzar un mensaje de error: " **No hay productos seleccionados para comenzar la facturación.** "	El mensaje lanza el mensaje de error correctamente.	✓	

Tabla 53. Caso de prueba 15 - Archivo no seleccionado para la carga de productos o servicios al inventario.

	Caso de prueba: Archivo no seleccionado para la carga de productos o servicios al inventario.				
No. Caso de prueba:	15	**Responsable:**	Angelo García.		
Descripción:	Verificar que el sistema lance un mensaje de error si no hay un archivo seleccionado para la carga de productos o servicios al inventario.				
Aspectos que debe cumplir la prueba.					
Precondiciones	**Pasos para ejecutar la prueba**	**Resultados esperados**	**Resultado obtenido**	**Aprueba**	
				Si	**No**
No debe existir un archivo seleccionado	1. Iniciar sesión como propietario.	El sistema debe lanzar un mensaje de	Los mensajes de error al intentar cargar	✓	

para la carga de productos o servicios al inventario.	2. Acceder al módulo de inventario. 3. Intentar cargar los producto o servicios al inventario.	error: " **Por favor, selecciona una factura. "** al intentar cagar productos y **“Por favor, selecciona un archivo XLSX.”** **a**l intentar cargar servicios.	productos y servicios cuando no hay archivos seleccionados muestran correctamente.		

Tabla 54. Caso de prueba 16 - Formato de archivo incorrecto para agregar productos o servicios al inventario.

	Caso de prueba: Formato de archivo incorrecto para agregar productos o servicios al inventario.			
No. Caso de prueba:	16	**Responsable:**	Angelo García.	
Descripción:	Verificar que el sistema lance un mensaje de error si el formato de archivo de la factura para cargar productos o servicios es incorrecto.			
Aspectos que debe cumplir la prueba.				
Precondiciones	**Pasos para ejecutar la prueba**	**Resultados esperados**	**Resultado obtenido**	**Aprueba**
				Si / **No**
Debe haber un archivo seleccionado de formato diferente a	1. Iniciar sesión como propietario. 2. Acceder al	El sistema debe lanzar un mensaje de error: " **Por favor,**	Los mensajes de error al intentar cargar productos y servicios de un	✓ /

XML en el caso de los productos y diferente a XLSX (Excel) para los servicios.	módulo de inventario. 3. Seleccionar un archivo de formato diferente. 4. Intentar cargar los producto o servicios al inventario.	**selecciona un archivo XML válido",** al intentar cargar productos **y "Por favor, selecciona un archivo XLSX válido",** al intentar cargar servicios al inventario.	archivo con formato incorrectos se muestran correctamente.		

Pruebas de rendimiento

Tabla 55. Caso de prueba 17 - Carga y estrés de usuarios y peticiones con un usuario.

		Caso de prueba: Carga y estrés de usuarios y peticiones con un usuario.						
No. Caso de prueba:		**17**		**Responsable:**		**Juan Aldas**		
Descripción:		Verificar independientemente cada escenario según una carga y estrés de 5 usuarios, con una cantidad determinada de peticiones.						
Escenario	**Solicitudes procesadas**	**Solicitudes exitosas**	**Solicitudes fallidas**	**Tasa de éxito (%)**	**Tiempo total (ms)**	**Tiempo promedio. (ms)**	**Tiempo Max. (ms)**	**Tiempo Min. (ms)**
Ingreso de productos.	100	100	0	100	5015	50	135	21

Registro de usuarios.	20	20	0	100	57000	2852	3341	2081
Registro de ventas.	100	100	0	100	70000	29	37	24

Tabla 56. Caso de prueba 18 - Carga y estrés de usuarios y peticiones con 5 usuarios.

	Caso de prueba: Carga y estrés de usuarios y peticiones con 5 usuarios.							
No. Caso de prueba:	**18**		**Responsable:**		**Juan Aldas**			
Descripción:	Verificar independientemente cada escenario según una carga y estrés de 5 usuarios, con una cantidad determinada de peticiones.							
Escenario	**Solicitudes procesadas**	**Solicitudes exitosas**	**Solicitudes fallidas**	**Tasa de éxito (%)**	**Tiempo total (ms)**	**Tiempo promedio. (ms)**	**Tiempo Max. (ms)**	**Tiempo Min. (ms)**
Ingreso de productos.	200	200	0	100	9767	48	136	19
Registro de usuarios.	20	20	0	100	57000	2852	3341	2081
Registro de ventas.	100	100	0	100	70000	29	37	24

Pruebas de usabilidad

Tabla 57. Caso de prueba 19 - Procesar una venta estándar.

	Caso de prueba: Procesar una venta estándar.			
No. Caso de prueba:	19	**Responsable:**	Ángelo García. Sñr. Wilman Remigio Gómez.	
Descripción:	Evaluar la facilidad de uso al procesar una venta con 5 productos.			
Aspectos que debe cumplir la prueba.				
Métricas	**Pasos para ejecutar la prueba**	**Resultados esperados**	**Resultado obtenido**	**Aprueba**
				Si / **No**
Tiempo, errores,	1. Escanear productos. 2. Registrar la venta. 3. Ver el recibo de la venta.	Venta procesada en menos de 1 minuto, sin errores significativos.	La venta se procesa en 45 segundos.	✓ /

Tabla 58. Caso de prueba 20 - Ver historial de ventas.

	Caso de prueba: Ver historial de ventas.			
No. Caso de prueba:	20	**Responsable:**	Ángelo García. Sñr. Wilman Remigio Gómez.	
Descripción:	Evaluar la capacidad del usuario para consultar el historial de ventas anteriores a través del sistema.			
Aspectos que debe cumplir la prueba.				
Métricas				**Aprueba**

	Pasos para ejecutar la prueba	Resultados esperados	Resultado obtenido		
				Si	No
Tiempo, facilidad de navegación, errores.	1. Navegar al listado. 2. Revisar detalles de una venta específica.	El usuario debe poder acceder al historial y encontrar una venta en menos de 30 segundos.	El usuario puede acceder al listado del historial de ventas y ver sus detalles en menos de 30 segundos.	✓	

5.4. Prueba de cierre

En esta parte, se revisan los resultados obtenidos de las pruebas y se identifican los aspectos positivos y negativos, y se establecen acciones para mejorar aspectos del sistema.

Tabla 59. Prueba de cierre.

Prueba de cierre				
Responsables: Juan Aldas, Angelo García.		**Objetivo:** Determinar los resultados obtenidos de las pruebas aplicadas al sistema On-line para la gestión de ventas del local comercial Isabelita.		
Promedio de valoración				
Deficiente = 1	**Aceptable = 2**		**Excelente = 3**	
No.	**Características**	**Métrica interna**	**Valoración**	**Promedio (%)**
1	Pruebas de funcionalidad	Verificar que todas las funcionalidades	3	100%

		cumplan con su propósito.		
2	Pruebas unitarias	Comprobar que las pruebas individuales cumplen con los requisitos.	3	100%
3	Pruebas de integración	Verificar la correcta integración entre los módulos.	3	100%
4	Pruebas de errores y excepciones	Comprobar la gestión de errores al ingresar datos inválidos.	3	100%
5	Pruebas de rendimiento	Evaluar el tiempo de respuesta en operaciones del sistema.	2.7	90%
6	Pruebas de usabilidad	Grado de satisfacción del usuario en la navegación.	2.8	93.33%
Promedio total			**2.91**	**97.221%**

5.5. Lanzamiento

5.5.1. Implementación

La implementación del sistema online para la gestión de ventas de la tienda Isabelita es un proceso que no solo se centra en las interfaces de los usuarios, sino también

se considera una infraestructura que brinda un soporte y un rendimiento estable y óptimo.

Para este proceso se ha utilizado una VPS (Virtual Private Server) donde se aloja tanto el backend como el frontend, desarrollado utilizando Node JS y React JS respectivamente, y la base de datos creada en PostgreSQL.

3.1.7.33. Preparación del entorno en la VPS

Para levantar el servidor y gestionar las conexiones entre el backend (Node.js) y el frontend, se utilizó Nginx como un intermediario y Docker que permite gestionar cada servicio del sistema de manera aislada, facilitando la administración y el despliegue.

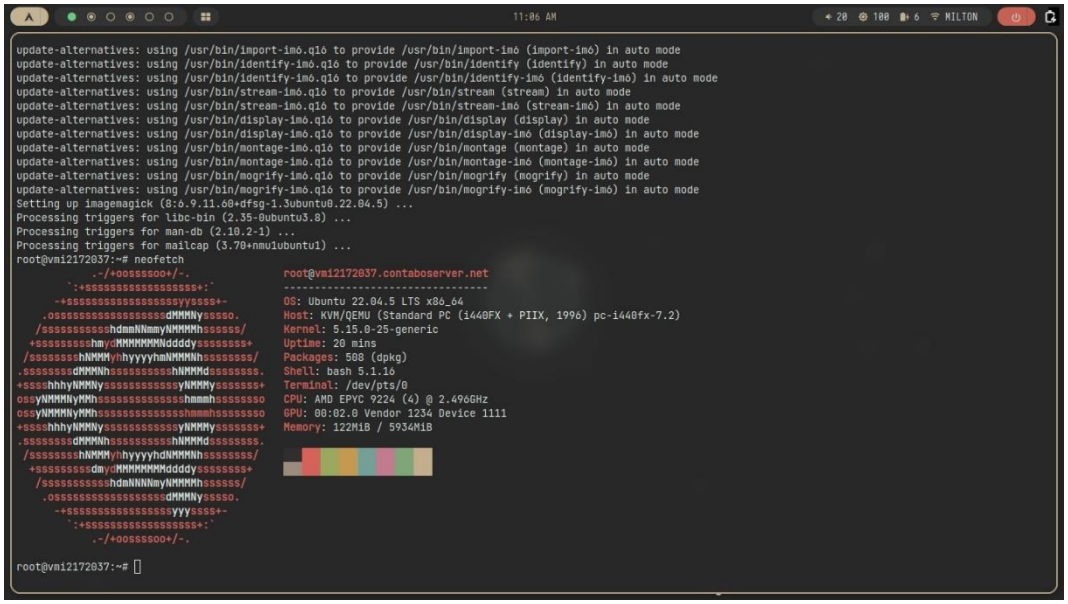


Gráfico 53. Configuración del servidor virtual privado.

3.1.7.34. Despliegue del sistema web

Para el despliegue se creó contenedores en la VPS utilizando Docker, el frontend fue empaquetado como una aplicación de React y el backend en Node.js permitiendo el balanceo de carga para repartir las peticiones.

3.1.7.35. Configuración de domino y subdominios

Se configuró el dominio y los subdominios en el proveedor de dominios, apuntándolos hacia la IP de la VPS. Esto permitió que tanto el frontend como la API en el backend fueran accesibles desde un subdominio personalizado.

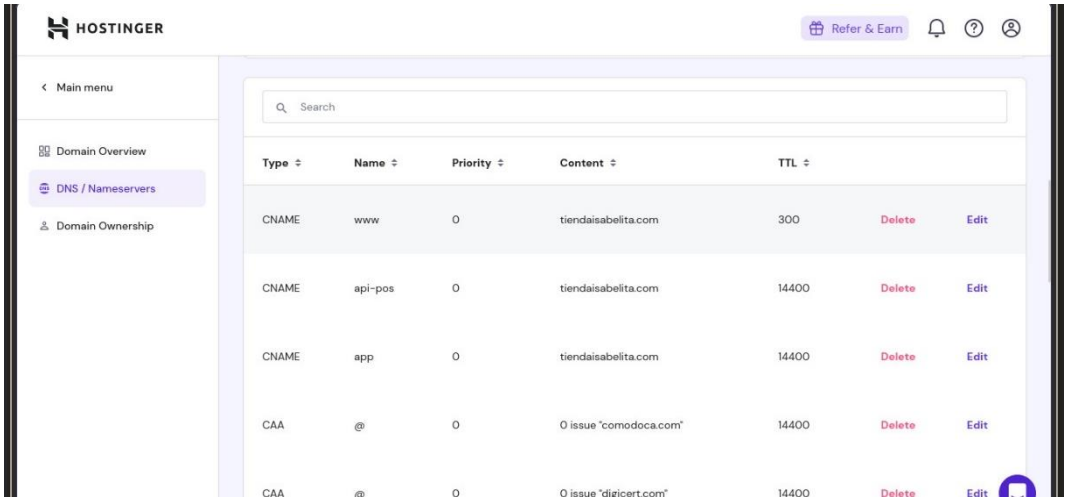


Gráfico 54. Configuración de dominio y subdominios.

3.1.7.36. Configuración de certificados SSL

Para asegurar las conexiones con el sistema, se utilizó Certbot para generar un certificado SSL a través de Let's Encrypt. Este certificado se configuró en Nginx para garantizar la seguridad y cifrado de las comunicaciones.

```
root@vmi2172037:~# cat /etc/nginx/sites-available/isabelita
upstream pos_isabelita {
    least_conn;
    server 172.18.0.3:3000;  # IP del contenedor spc-api_1
    server 172.18.0.4:3000;  # IP del contenedor spc-api_2
}

server {
    listen 80;
    listen [::]:80;
    server_name 62.84.183.227;

    location / {
        return 301 https://$host$request_uri;
    }
}

server {
#    listen 443 ssl;
 #   listen [::]:443 ssl;
    server_name 62.84.183.227;

 #   ssl_certificate /etc/letsencrypt/live/spc-api.devsoftec.com/fullchain.pem;
  #  ssl_certificate_key /etc/letsencrypt/live/spc-api.devsoftec.com/privkey.pem;

    # Configuración de SSL para protocolos fuertes
#    ssl_protocols TLSv1.2 TLSv1.3;
 #   ssl_prefer_server_ciphers off;
  #  ssl_ciphers "EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH";

    location / {
        proxy_pass http://pos_isabelita;  # Nombre del upstream definido arriba
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_cache_bypass $http_upgrade;
    }
```

Gráfico 55. Configuración de certificados SSL.

3.1.7.37. Resultado del despliegue

El sistema para la gestión de ventas quedó completamente desplegado y accesible a través del dominio configurado, con balanceo de carga entre las instancias de API, conexiones seguras mediante SSL y completamente en funcionamiento.

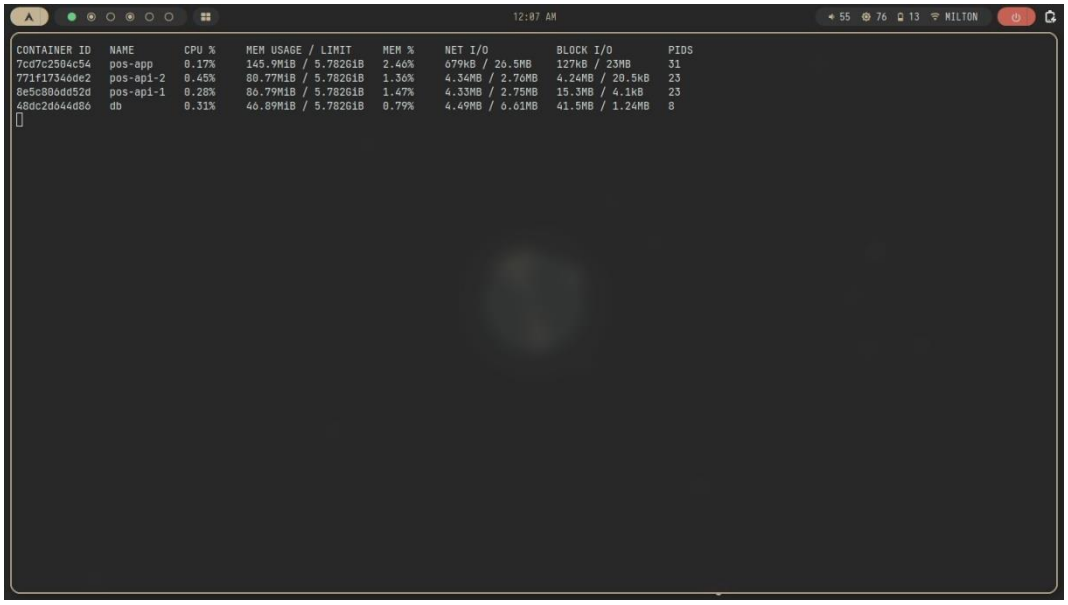


Gráfico 56. Servicios del sistema On-line desplegados.

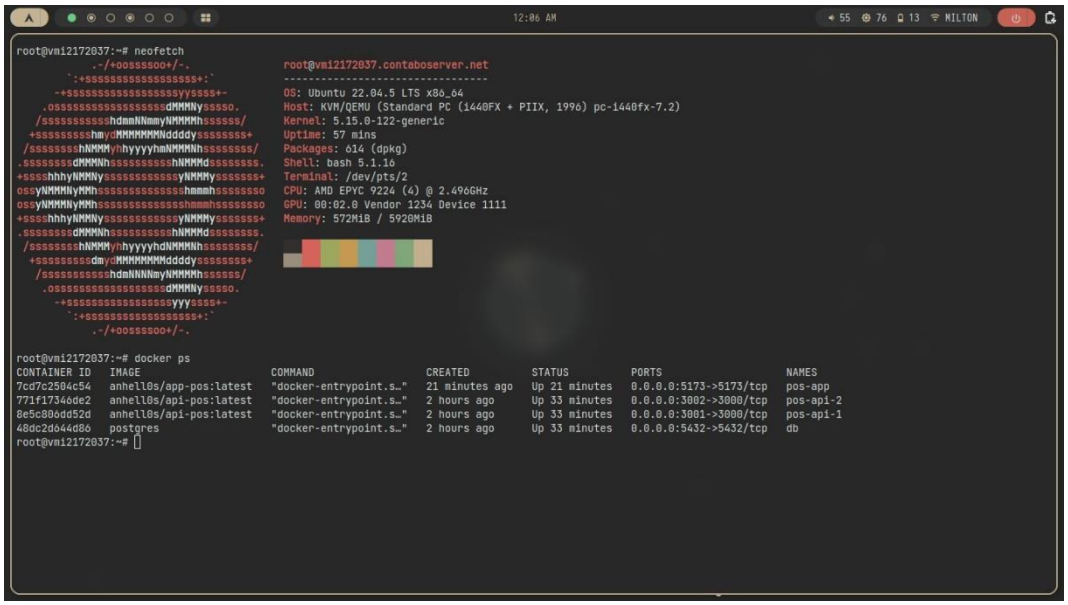


Gráfico 57. Servidor virtual privado activo y ejecutando el sistema.

3.1.7.38. Fecha de implementación.

El sistema On-line para la gestión de ventas fue implementado el 03/10/2024 en la tienda isabelita.

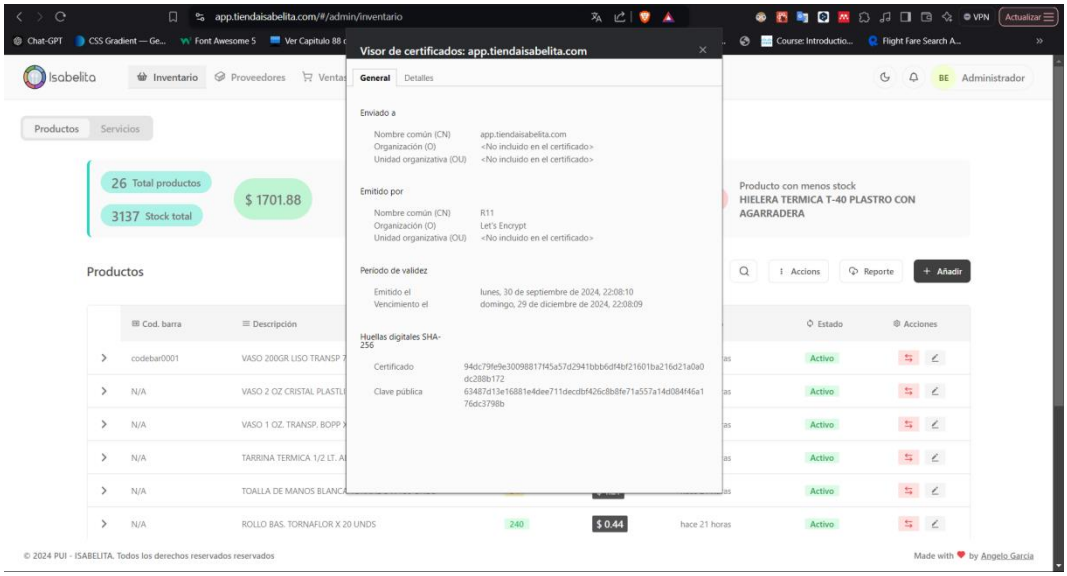


Gráfico 58. Sistema en ejecución en el navegador, mostrando el dominio y el candado de seguridad SSL visible.

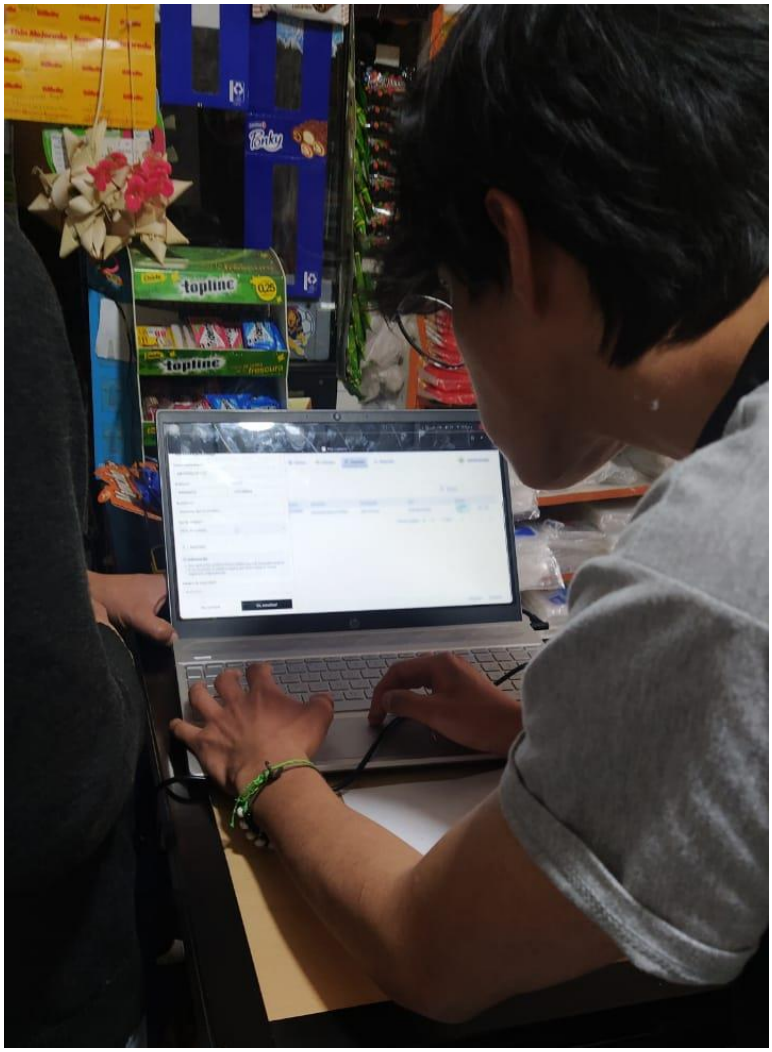


Gráfico 59. Capacitación del sistema con el propietario.

CONCLUSIONES

- El análisis de los requerimientos funcionales y no funcionales del negocio permitió identificar las necesidades del local comercial para el desarrollo del sistema informático de manera correcta, obteniendo un sistema a la medida.
- La aplicación de la metodología Scrum en el desarrollo del sistema on-line para la gestión de ventas en el local comercial ISABELITA se llevó a cabo con éxito, completando los 6 sprints planificados, cada sprint permitió al equipo de desarrollo entregar funcionalidades valiosas de forma incremental, asegurando que el sistema se adaptará a las necesidades cambiantes del negocio, la colaboración estrecha entre el equipo de desarrollo y el propietario durante cada sprint garantizó que el sistema web se ajustara a sus expectativas.
- El diseño del sistema on-line es dinámico y minimalista, manteniendo una concordancia entre la tipografía y los colores que se derivan del logo que maneja el local comercial, con una interfaz intuitiva que permite navegar fácilmente por el sistema y realizar las tareas de gestión de ventas de manera sencilla y eficiente, centrándose en la facilidad de uso y la accesibilidad para todos los usuarios.
- La aplicación de un plan de pruebas ha sido fundamental para asegurar la calidad y el correcto funcionamiento del sistema, las pruebas han detectado y corregido errores potenciales, asegurando que el sistema sea estable y funcione correctamente.
- La implementación del sistema web representa un cambio significativo para ISABELITA, pasando de un sistema manual a uno digital que automatiza y optimiza las tareas de gestión de ventas, permitiendo operar de manera más eficiente, con mayor control sobre sus operaciones y acceso a información en tiempo real.

RECOMENDACIONES

- Mantener un contacto directo y frecuente con el cliente para determinar los requerimientos de forma precisa y confiables, asegurando así satisfacer las necesidades de cada usuario.
- Utilizar metodología ágil como Scrum, que ha demostrado ser efectiva en el desarrollo de software al permitir flexibilidad y adaptabilidad, esto permitirá una gestión eficiente, una respuesta rápida a los cambios y la entrega constante de funcionalidades valiosas para satisfacer las necesidades del cliente o propietario de un sistema.
- Se sugiere que para un mejor diseño de sistemas online es importante mantener la concordancia entre la tipografía, los colores a usar y la distribución de los elementos.
- Para asegurar la calidad del sistema a largo plazo, se recomienda realizar pruebas regulares, en diferentes etapas del ciclo de vida del software, esto ayudará a detectar errores antes de que afecten a los usuarios y garantizar que el sistema se mantenga estable y funcional.
- Es importante realizar copias de seguridad periódicas y completas del sistema para garantizar la recuperación de datos en caso de fallos, una de las formas de asegurar esto es adquiriendo un hosting que incluya el mantenimiento y los backups como parte de sus características.

BIBLIOGRAFÍA

Armijos Gadvay, D. J., & Fernández Ninabanda, N. E. (2022). *Sistema de Evaluación y Aseguramiento de la Calidad (SEAC) para las carreras de la Facultad de Ciencias Administrativas, Gestión Empresarial e Informática de la Universidad Estatal de Bolívar, año 2022*.

Barreto Barreto, K. A., & Vargas Tumbaco, M. D. (2021). ANÁLISIS DEL SISTEMA DE GESTIÓN DE VENTAS DE LA EMPRESA WALPESI S.A. *UNIVERSIDAD DE GUAYAQUIL* .

Bermúdez León, M. J. (2020). *INTRODUCCIÓN A LAS BASES DE DATOS.*

Canive, T. (2020, mayo 27). *Metodología XP o Programación Extrema: ¿Qué es y cómo aplicarla?* Gestor de proyectos online. https://www.sinnaps.com/blog-gestion-proyectos/metodologia-xp

Catalán Calabuig, C., & Esparza Peidro, J. (2022). *Aplicación web para la gestión de los comerciales en una empresa*.

Cubero, H. (2020). *INTRODUCCIÓN A LA INGENIERÍA DE SOFTWARE*.

Fernanda Begnini Domínguez, L. I., Clara Lecaro Lavayen, A. I., & Daniel Shauri Romero III, J. (2022). Ventajas de la automatización de la gestión por procesos. *Polo del Conocimiento*, *7*(7).

Fernández Aranda, Ó. (2018). *Aplicación de metodología Scrum para el desarrollo de una API web de gestión médica con MEAN Stack*.

Fernández, N., Reyes, C., Ramirez, J., & Barbieri Sarmiento, N. (2019). Desarrollo ágil y reutilización de Software Libre. *SIE, Simposio de Informática en el Estado*.

Flat. (2019, septiembre 6). *Métodos de control de inventarios más usados - ESERP Business School*. ESERP Digital Business & Law School. https://es.eserp.com/articulos/metodo-control-inventarios/

Furman Dzyadzya, R., & Aura Esteve, C. (2022). *Análisis para la implantación de los programas de Gestión de Relaciones con los Clientes (CRM) en pequeñas y medianas empresas (PYMES)*.

García Islas, L. H., Franco Sánchez, K. D., Samperio Monroy, T. I., Pérez Tavera, I. H., & Ocampo López, A. (2021). C-ToolP: Herramienta de apoyo para la enseñanza de desarrollo Web-SPA-MVC. *Pädi Boletín Científico de Ciencias Básicas e Ingenierías del ICBI*, *9*(Especial), 79–84. https://doi.org/10.29057/icbi.v9iespecial.7470

Heredia Sayay, S. R. (2022). *Sistema web para la gestión de ventas del taller de reparación y mantenimiento automotriz Heredia utilizando el framework Mean Stack*.

Hernandez Sola, G. (2022, noviembre 19). *Un Product Owner no es un Project Manager*. Scrum.org. https://www.scrum.org/resources/blog/un-product-owner-no-es-un-project-manager

Maldeadora. (2018, febrero 21). *Qué es Frontend y Backend: características, diferencias y ejemplos*. Platzi. https://platzi.com/blog/que-es-frontend-y-backend/

Martín Gómez, S. (2020). Aplicación de las Metodologías Ágiles al proceso de enseñanza-aprendizaje universitario. *Revista d'Innovació Docent Universitària*, 62–73. https://doi.org/10.1344/ridu2020.12.7

Martins, J. (2024, febrero 15). *Scrum: conceptos clave y cómo se aplica en la gestión de proyectos [2024] • Asana*. Asana. https://asana.com/es/resources/what-is-scrum

MDN. (2024, marzo 5). *JavaScript | MDN*. MDN Web Docs. https://developer.mozilla.org/en-US/docs/Web/JavaScript

Mecías García, S. E. (2022). SISTEMA PARA LA GESTIÓN DE LAS VENTAS EN LA EMPRESA SUPER 10 UBICADA EN LA CIUDAD DE SANTO DOMINGO DE LOS TSÁCHILAS. *UNIVERSIDAD REGIONAL AUTÓNOMA DE LOS ANDES*.

Moisset, S. (2023, julio 5). *Guía de seguridad del software de código abierto—Mejores prácticas para asegurar tus proyectos*. Medium. https://medium.com/@sonya.moisset/manual-de-seguridad-del-software-de-c%C3%B3digo-abierto-mejores-pr%C3%A1cticas-para-asegurar-tus-proyectos-12bfb5a1fffb

Molina Ríos, J. (2020). *Modelo de evaluación de metodologías de desarrollo de software web*.

Morales Carrillo, J., Cedeño Valarezo, L., Cajape Bravo, J. S., & Ormaza Calderón, G. J. (2021). *Metodologías de desarrollo de software y su ámbito de aplicación: Una revisión sistemática*.

Ortega Blacio, S., Lopez Monge, Z., & Eras Agila, R. de J. (2022). Métodos de control y valoración de inventarios utilizados en empresas de electrodomésticos del cantón Machala. *593 Digital Publisher CEIT*, *7*(5–1), 315–327. https://doi.org/10.33386/593dp.2022.5-1.1363

Rouse, M. (2019, noviembre 14). *Sistema de gestión de bases de datos o DBMS*. ComputerWeekly.es. https://www.computerweekly.com/es/definicion/Sistema-de-gestion-de-bases-de-datos-o-DBMS#:~:text=Un%20sistema%20de%20administraci%C3%B3n%20de,recuperar%2C%20actualizar%20y%20administrar%20datos.

SafetyCulture. (2024, junio 25). *Sistema de inventario: Definición, tipos y métodos*. SafetyCulture.

Sánchez, P. Y. C. (2020). Implementación del patrón arquitectónico MVC en aplicaciones web para la arquitectura del software del sistema de capellanía de la UM. *Anuario de Investigación UM*, *1*(1). http://anuarioinvestigacion.um.edu.mx/index.php/anuarioium/article/view/118

Schwaber, K., & Sutherland, J. (2020). *La Guía de Scrum. La Guía Definitiva de Scrum: Las Reglas del Juego*.

SRI. (2024). *Facturación Electrónica - intersri - Servicio de Rentas Internas*. SRI. Servicio de rentas internas. https://www.sri.gob.ec/facturacion-electronica

Tetteh, S. G. (2024). Empirical Study of Agile Software Development Methodologies: A Comparative Analysis. *Asian Journal of Research in Computer Science*, *17*(5). https://doi.org/10.9734/ajrcos/2024/v17i5436

Tian, F., Liang, P., & Babar, M. A. (2022). Relationships between software architecture and source code in practice: An exploratory survey and interview. *Information and Software Technology, 141*. https://doi.org/10.1016/j.infsof.2021.106705

Unir, V. (2023, noviembre 27). *Framework: qué es, para qué sirve y algunos ejemplos.* UNIR FP. https://assemblerinstitute.com/blog/framework-programacion/#:~:text=En%20programaci%C3%B3n%2C%20un%20framework%20es,de%20un%20proyecto%2C%20por%20ejemplo.

Valverde, V., Portalanza, N., & Mora, P. (2019). ANÁLISIS DESCRIPTIVO DE BASE DE DATOS RELACIONAL Y NO RELACIONAL. *Revista Atlante: Cuadernos de Educación y Desarrollo*. https://www.eumed.net/rev/atlante/2019/06/base-datos-relacional.html

Vega, A. (2024, octubre 8). *Metodologías de desarrollo de software: ¿Qué son y cómo funcionan?* Gurusis. https://gurusis.com/metodologias-de-desarrollo-de-software/

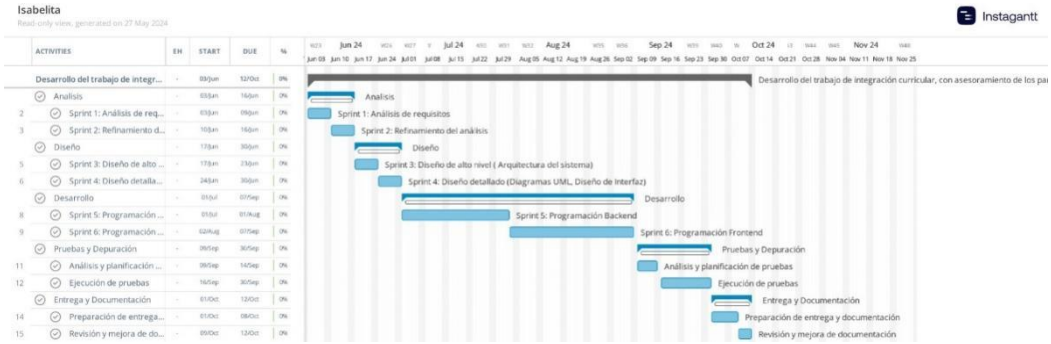
Zamora Irrazabal, E. J. (2021). *DESARROLLO DE UNA APLICACIÓN WEB PARA EL CONTROL DE VENTAS EN LA EMPRESA LF PUBLICIDAD*.

ANEXOS

ANEXO 1

Cronograma (Gantt)

Cronograma de actividades del proyecto tecnológico



ANEXO 2

Presupuesto ejecutado

PRESUPUESTO

Para la implementación del sistema para la gestión de ventas en el local comercial Isabelita se estima los siguientes costos.

Tabla de presupuesto.

Recursos	**Cantidad**	**Precio unitario**	**Total**
Computador	2	700	1400
Suministros de oficina	100	1	100
Hosting	12	5,83	70
Dominio	1	10,5	10,5
Impresora	1	300	300
Internet	4	25	100
Fotografías	100	1	100
Transporte	30	0,50	15
Otros	1	100	100
Total			**$ 2198,5**

El costo total de la ejecución del proyecto es de **2198,5** dólares.

Estimación por COCOMO

El valor del presupuesto de desarrollo se calcula mediante el modelo básico orgánico de COCOMO.

Modelo Orgánico

Esfuerzo de desarrollo (Hombre – Mes)

MM = 2.4 * (KDSI)^1.05

MM = 2.4 * (3.42)^1.05

MM = 8.72

Tiempo de desarrollo (Mes)

TDEV = 2.5 * (MM)^0.38

TDEV = 2.5 * (8.72) ^ 0.38

TDEV = 5.69

Cantidad de Hombres

CH = MM / TDEV

CH = 8.72 / 5.69

CH = 1.53 ≈ 2

Estimación del tiempo con el número de desarrolladores

TDEVA = MM / CH

TDEVA = 8.72 / 2

TDEVA = 4.36

Estimación de salario de desarrollador

Sueldo = TDEVA * CMO * CH

Sueldo = 4.36 * 460 * 2

Sueldo = 4012.80

Costo total del proyecto (CTP)

CTP = Sueldo

CTP = 4012.80 dólares.

Presupuesto Total

2198,50 $ + 4012.80 $ = 6211.30 dólares

El presupuesto total del proyecto tecnológico es de 6211.30 dólares.

ANEXO 3

Carta de aceptación de la organización donde se realizó el trabajo de integración curricular.

Carta de aceptación del local comercial ISABELITA

Guaranda 10 de mayo del 2024

Señores

JUAN CARLOS ALDAS ALDAZ

ANGELO DAVID GARCÍA ARTEAGA

Estudiantes 8vo Ciclo

Carrera de Sofware

Presente. –

De mi consideración:

Por el presente la autorización, al pedido realizado por ustedes, respecto a su interés, realizar el proyecto denominado **"DESARROLLO DE UN SISTEMA WEB PARA LA GESTIÓN DE VENTAS EN LA TIENDA ISABELITA DE LA CIUDAD DE GUARANDA"**, autorizo se desarrolle el mismo en la tienda de la cual represento y así poder apoyar en su proceso de titulación.

Atentamente,

Sr. Wilman Remigio Gómez

Propietario tienda ISABELITA

RUC: 0201985009001

ANEXO 4

Instrumentos de recopilación de datos

Entrevista

La entrevista se aplicó al propietario del local comercial ISABELITA con el objetivo de identificar las necesidades y requerimientos del sistema online.

Ficha de Entrevista
Nombre del entrevistado: Wilman Remigio Gómez **Cargo: Propietario local comercial Isabelita** **Fecha de la entrevista: 3 de junio del 2024** **Entrevistadores: Juan Aldas Aldaz, Angelo García**
Objetivo de la entrevista: Recopilar información sobre los procesos de ventas actuales, los productos vendidos y las necesidades específicas del negocio del local comercial Isabelita.
Preguntas de la entrevista: **1. ¿Cómo se lleva a cabo el proceso de ventas en el local comercial Isabelita, desde que un cliente entra hasta que realiza una compra?** Primero que todo viene el cliente para satisfacer una necesidad, a través de un producto o servicio, entonces si es que tenemos el producto o si tenemos el **servicio**, si está conforme paga y luego se despacha. Por el momento el registro de ventas se realiza en un cierre de caja después de la jornada laboral. **2. ¿Como se realiza el control de inventario actualmente?** Nosotros el control de inventario hacemos a través de un Kardex en Excel, viendo si es que el stock está completo, en caso de no estar completo o este menos del mínimo, hacemos los pedidos a los proveedores. También depende de la temporada, por ejemplo, en la temporada de fiestas, entonces hacemos un pedido mayor al stock que tenemos o en temporadas bajas como cuando no hay clases, disminuimos la cantidad de pedidos.

3. **¿Qué tipos de productos se venden en el local comercial Isabelita? ¿Existe alguna clasificación o categorización específica de productos que debamos tener en cuenta?**

   Nosotros tenemos principalmente lo que es la venta y la distribución de los utensilios desechables y dentro de estos tenemos la clasificación de lo que son el material térmico, el material transparente, utensilios de aluminio y de madera como los palillos. Y también tenemos otros servicios grandes que son la venta de los canales en vivo y los Depósitos y Retiros del Banco de Pichincha, el Banco de Guayaquil, así como también de las recargas. Esto está dentro del grupo de los servicios.

4. **¿Como realiza usted el manejo de proveedores?**

   Bueno, con los proveedores realizamos los pedidos cada vez que falta el stock y principalmente realizamos pagos a través de cheques, también de transferencias y bueno, depende de la demanda, o sea, por ejemplo, algunos proveedores tomamos los pedidos cuando son temporadas altas, como en algunas festividades, entonces de eso también depende el pedido que hagamos a los proveedores. El manejo de los proveedores, en si el registro lo realizamos en Excel, en un listado donde cuenta con los datos más relevantes del proveedor y los productos o servicios que ofrece.

5. **¿Cada que tiempo realiza los pedidos de mercadería para su local?**

   Los pedidos dependen mucho de la demanda de la temporada, entonces hay algunos productos que pedimos de manera constante, por ejemplo, los vasos, que puede ser cada tres días. Otros productos que tienen menos demanda, por ejemplo, las tarrinas transparentes para las galletas y para los frutos que se pide una vez cada mes, o también otros productos de alta demanda en el caso de las tarrinas térmicas que se pide cada 10 días en promedio. En el caso de los servicios, pedimos a los proveedores cada semana prácticamente el servicio de los canales de envío.

6. **¿Como establece el precio a cada producto para la venta?**

   Primero que todo el costo de la compra del producto, en la factura que nos emiten ya viene incluido el IVA, en este caso el 15%, a eso sumamos los costos operativos y luego de acuerdo con el mercado evaluamos si es que está dentro del rango del 30%, del 25%. También tenemos otra clasificación para negocio que se aplica el 8%, el 10 %, también depende del mercado, o sea cómo están vendiendo los demás compañeros de los otros establecimientos, si es que está muy barato entonces deja poca ganancia, lo que hacemos es buscar otro proveedor que nos pueda dejar más barato o a su vez descontinuamos ese producto o lo reemplazamos con otra marca.

7. **¿Dónde registra las ventas realizadas de sus productos?**

   Nosotros realizamos en un Kardex de salida de los productos, en Excel, de una forma bastante simple, es un listado de los productos que contiene los productos, cantidades, precisos, proveedor e información relevante para la venta. En otras ocasiones donde no hay tiempo libre y es una venta que es importante, se lo hace de forma rápida en una hoja de cuaderno.

8. **¿Qué tipo de impuesto declara usted?**

   Nosotros declaramos lo que es el impuesto del RIMPE, del negocio popular que se paga cada año. Además, también realizamos declaraciones de la otra categoría grande de los servicios, el IVA, que es del 15% y que se hace semestralmente.

9. **¿Como maneja la actualización de los precios de los productos?**

   Eso manejamos de acuerdo con el inventario, revisamos el último precio al que se ha comprado y luego en el registro de Excel simplemente lo actualizamos.

10. **¿Existen necesidades o requisitos específicos que desee que un sistema web de gestión de ventas pueda proporcionar? Por ejemplo, control de inventario, generación de informes, gestión de promociones, etc.**

Bueno, específicamente a mí me gustaría que se generen que se generen reportes de los últimos precios de manera automática, de las facturas que se generan en SRI y también me gustaría que se pueda realizar la facturación electrónica para poder al cliente despachar de manera más rápida. Algo adicional es que me gustaría que se realice es el ingreso de productos en el inventario a través de una factura en formato XML de forma automática.
Conclusiones: En la entrevista con Wilman Remigio Gómez, propietario del local comercial Isabelita, se destaca que el proceso de ventas es sencillo, con registro diario en Excel durante el cierre de caja. El control de inventario se ajusta según la demanda estacional, y los productos principales son utensilios desechables y servicios bancarios. Los proveedores se gestionan con pagos por cheques o transferencias, y los pedidos varían en frecuencia según el producto. Los precios se establecen considerando costos, impuestos y competencia. Las ventas y el inventario se registran en Excel. El local comercial declara impuestos del RIMPE y el IVA. El propietario del local comercial Isabelita expresa la necesidad de un sistema de gestión de ventas que facilite reportes automáticos, facturación electrónica y actualización de inventario mediante facturas XML.

Ficha de observación

Nombre del local comercial: Isabelita
Propietario: Wilman Remigio Gómez

Fecha de observación: 10 de junio del 2024
Observadores: Aldas Juan, García Angelo

Objetivo de la observación:
Analizar los procesos operativos y administrativos del comercial Isabelita que se dedica a la venta de productos desechables, con el fin de implementar un sistema de gestion de ventas adecuado.

---

1. Infraestructura del local comercial

- **Espacios:**
    - Área de atención al cliente: espacio principal donde se exhiben los productos desechables.
    - Área de almacenamiento: espacio destinado para inventario y control de stock.
- **Equipos:**
    - Laptop para registros y gestión en Excel.
    - Estantes o vitrinas para exhibir productos.

---

2. Procesos identificados

5.5.2. a. Ventas

- **Flujo de ventas:**
    1. Cliente ingresa y solicita productos.
    2. Se verifica el stock disponible.
    3. Producto se entrega al cliente tras el pago.
- **Métodos de pago:**
    - Efectivo.

- Pagos con tarjeta (Transferencias Banca Pichincha).

- **Registro de ventas:**
  - En Excel o en cuadernos físicos.
  - No existe un registro en tiempo real, se realiza al cierre del día.

5.5.3. b. Control de inventario

- **Gestión del inventario:**
  - Se lleva a cabo utilizando un Kardex en Excel.
  - Productos se categorizan según:
    - Material térmico.
    - Material transparente.
    - Utensilios de aluminio.
  - Ajustes de inventario según la demanda estacional.

5.5.4. c. Gestión de proveedores

- **Relación con proveedores:**
  - Pedidos se realizan en función de la demanda y el stock disponible.
  - Temporadas altas requieren pedidos más frecuentes.
- **Métodos de pago:**
  - Transferencias bancarias o pagos en efectivo.
- **Registro:**
  - Información de proveedores y pedidos se mantiene en un listado de Excel.

5.5.5. d. Fijación de precios

- **Factores considerados:**
  - Costo del producto (incluido IVA proveedor + IVA propietario).
  - Margen de ganancia (rango del 15% al 30%).
  - Competencia y demanda del mercado.
- **Actualización de precios:**
  - Manual, basada en el precio de adquisición más reciente.

5.5.6. e. Declaración de impuestos

- **Impuestos declarados:**
    - Declaraciones de impuesto al RIMPE.

---

3. Herramientas y métodos actuales

- **Tecnología utilizada:**
    - Excel para inventarios, precios, y ventas.
    - Uso ocasional de cuadernos físicos para registros rápidos.
- **Limitaciones:**
    - Proceso manual y propenso a errores.
    - Falta de registros en tiempo real para ventas e inventario.

---

4. Necesidades detectadas para un sistema de gestión de ventas

1. **Automatización:**
    - Registro en tiempo real de ventas e inventario.
    - Generación automática de reportes de ventas, inventario, y ganancias.
2. **Control de inventario:**
    - Gestión integrada de entradas y salidas.
    - Alertas automáticas para stock bajo.
3. **Gestión de proveedores:**
    - Registro centralizado de proveedores y pedidos.
4. **Interfaz amigable:**
    - Acceso rápido a productos y categorías.
    - Configuración sencilla de precios y promociones.

---

5. Conclusión de la observación

El local comercial opera con procesos manuales que generan ineficiencias, especialmente en temporadas altas. La implementación de un sistema de gestión de ventas optimizará la gestión del inventario, agilizar el proceso de ventas, y

mejorar la experiencia del cliente, además de asegurar un control preciso sobre los aspectos administrativos del negocio.

ANEXO 5

Certificado Registro Único de Contribuyentes del propietario del local comercial Isabelita

Certificado Registro Único de Contribuyentes del propietario del local comercial Isabelita.

El RUC del propietario del local comercial isabelita certifica que pertenece al RIMPE, en la categoría Negocio Popular.



SRI

Certificado
Registro Único de Contribuyentes

Apellidos y nombres		Número RUC
ALDAS GOMEZ WILMAN REMIGIO		0201985009001
Estado	**Régimen**	**Artesano**
ACTIVO	RIMPE - NEGOCIO POPULAR	No registra
Fecha de registro	**Fecha de actualización**	
20/01/2020	24/01/2024	
Inicio de actividades	**Reinicio de actividades**	**Cese de actividades**
20/01/2020	02/02/2022	30/01/2021
Jurisdicción		**Obligado a llevar contabilidad**
ZONA 5 / BOLIVAR / GUARANDA		NO
Tipo	**Agente de retención**	**Contribuyente especial**
PERSONAS NATURALES	NO	NO

Domicilio tributario

Ubicación geográfica

Provincia: BOLIVAR **Cantón:** GUARANDA **Parroquia:** GABRIEL IGNACIO VEINTIMILLA

Dirección

Calle: AZUAY **Número:** S/N **Intersección:** ENTRE 7 DE MAYO Y 9 DE ABRIL **Referencia:** A UNA CUADRA DEL BANCO PICHINCHA

Medios de contacto

Email: wilmanremigioaldas@gmail.com **Celular:** 0981173648

Actividades económicas

- G47110101 - VENTA AL POR MENOR DE GRAN VARIEDAD DE PRODUCTOS EN TIENDAS, ENTRE LOS QUE PREDOMINAN, LOS PRODUCTOS ALIMENTICIOS, LAS BEBIDAS O EL TABACO, COMO PRODUCTOS DE PRIMERA NECESIDAD Y VARIOS OTROS TIPOS DE PRODUCTOS.
- G47590304 - VENTA AL POR MENOR DE UTENSILIOS DE PLÁSTICOS EN ESTABLECIMIENTOS ESPECIALIZADOS.
- G47732302 - VENTA AL POR MENOR DE ALIMENTO PARA MASCOTAS EN ESTABLECIMIENTOS ESPECIALIZADOS.

Establecimientos

Abiertos	Cerrados
1	1

Obligaciones tributarias

- 1011 - DECLARACIÓN DE IMPUESTO A LA RENTA PERSONAS NATURALES

Las obligaciones tributarias reflejadas en este documento están sujetas a cambios. Revise periódicamente sus obligaciones tributarias en www.sri.gob.ec.

1/2

www.sri.gob.ec

Apellidos y nombres	Número RUC
ALDAS GOMEZ WILMAN REMIGIO	0201985009001

Números del RUC anteriores

No registra

Código de verificación:	RCR1722356590409439
Fecha y hora de emisión:	30 de julio de 2024 11:23
Dirección IP:	157.100.57.172

Validez del certificado: El presente certificado es válido de conformidad a lo establecido en la Resolución No. NAC-DGERCGC15-00000217, publicada en el Tercer Suplemento del Registro Oficial 462 del 19 de marzo de 2015, por lo que no requiere sello ni firma por parte de la Administración Tributaria, mismo que lo puede verificar en la página transaccional SRI en línea y/o en la aplicación SRI Móvil.

ANEXO 6

Manual de usuario del Propietario

Isabelita

Manual de usuario - Administrador

Sistema de Gestión de Ventas

Juan Aldas
Angelo García

07 de octubre del 2024

1. Introducción 3
1.1. Descripción general del sistema 3
1.2. Objetivo 3
1.3. Audiencia objetivo 3
1.4. Requisitos del sistema 3
2. Acceso al sistema 4
2.1. Inicio de Sesión 4
2.2. Recuperación de contraseña 4
3. Interfaz de usuario 6
3.1. Descripción del panel principal 6
3.2. Barra de navegación 7
4. Módulos del Sistema 8
4.1. Inventario: Gestión de productos y servicios 8
4.1.1. Productos 8
4.1.2. Servicios 13
4.2. Proveedores: Administración de datos de proveedores 17
4.3. Ventas: Control de transacciones y facturación 18
4.4. Clientes: Información y gestión de clientes 20
4.5. Usuarios: Gestión de usuarios del sistema 21
4.6. Kardex: Control detallado de movimientos de inventario 24
4.7. Configuraciones de la cuenta 26
4.8. Configuración del Sistema 27
4.8.1. Establecimiento 27
4.8.2. Económico 28
4.8.3. Respaldo 29

Introducción

Descripción general del sistema

El Sistema de Gestión de Ventas es una plataforma web diseñada para optimizar y simplificar el control de ventas, clientes, productos e inventarios, especialmente enfocada en empresas dedicadas a la venta de productos desechables. Este sistema permite registrar, gestionar y consultar ventas en tiempo real, generando informes detallados que facilitan un análisis eficiente del desempeño comercial. Con esta herramienta, las empresas pueden mejorar su toma de decisiones, asegurando una mayor eficiencia en la gestión de su inventario y aumentando la satisfacción del cliente.

Objetivo

El principal objetivo de este sistema es automatizar el proceso de ventas, incrementando la precisión en los registros, minimizando errores humanos y ofreciendo herramientas que permiten monitorear y optimizar la toma de decisiones empresariales. Esto contribuye a una gestión más eficiente de las operaciones diarias, asegurando un mayor control y rendimiento en las ventas.

Audiencia objetivo

Este manual está dirigido a:

- Administradores responsables de supervisar el rendimiento de las ventas y gestionar el inventario.

Requisitos del sistema

Para utilizar este sistema de manera óptima, se requieren los siguientes recursos:

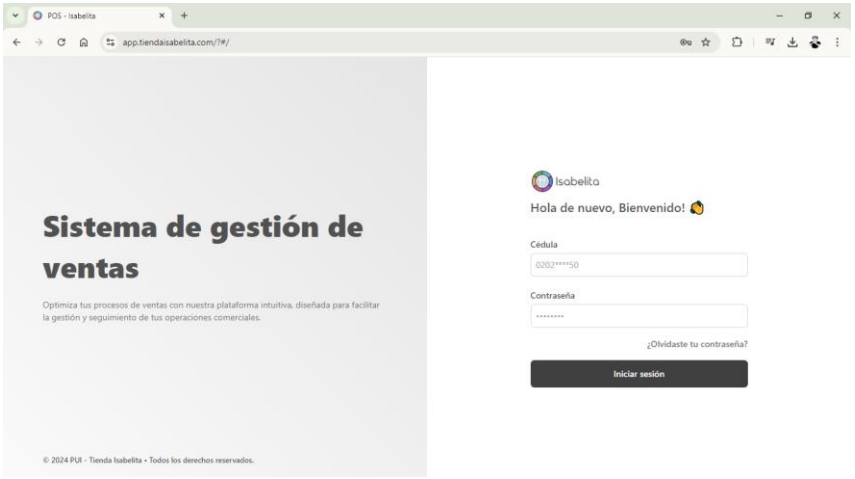
- **Navegador web:** Compatible con Google Chrome, Mozilla Firefox, Safari o Microsoft Edge (es recomendable tener la última versión instalada para garantizar la mejor experiencia).
- **Conexión a internet:** Velocidad mínima recomendada de 5 Mbps para una navegación fluida y sin interrupciones.

- **Resolución de pantalla:** Mínimo 1366x768, aunque se recomienda 1920x1080 para una visualización óptima de los gráficos y paneles.
- **Hardware:** Procesador de al menos 2 GHz y 4 GB de memoria RAM (se recomienda 8 GB para un rendimiento más eficiente en multitareas).
- **Sistema operativo:** Windows 10, macOS 10.14 o superior, o distribuciones de Linux compatibles.
- **Software adicional:** JavaScript habilitado en el navegador y cookies activas para una correcta operación del sistema.
- **Dispositivo compatible:** Se puede acceder desde computadoras de escritorio, portátiles, tabletas y smartphones, siempre que cumplan con los requisitos anteriores.

Acceso al sistema

Inicio de Sesión

Para acceder al sistema, ingrese su correo cédula y contraseña en la página principal haga clic en “Iniciar sesión”. Si las credenciales son correctas, será redirigido al panel principal.

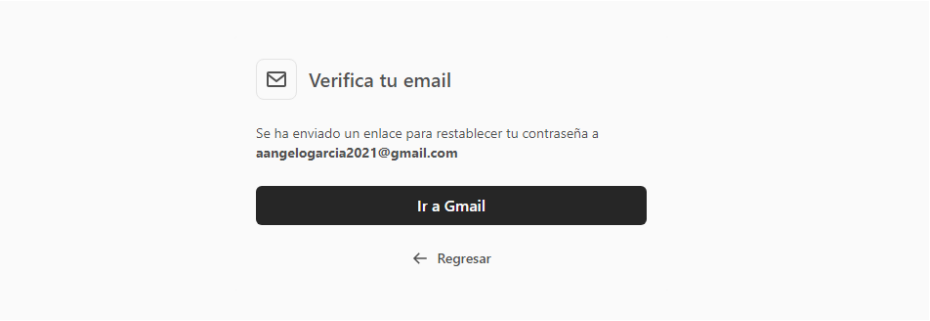


Recuperación de contraseña

Si ha olvidado su contraseña, seleccione “¿Olvidaste tu contraseña?”



Y siga las instrucciones para recibir un correo electrónico con un enlace de recuperación.





Haga clic en el enlace para crear una nueva contraseña. Asegúrese de que la nueva contraseña cumpla con los siguientes requisitos:

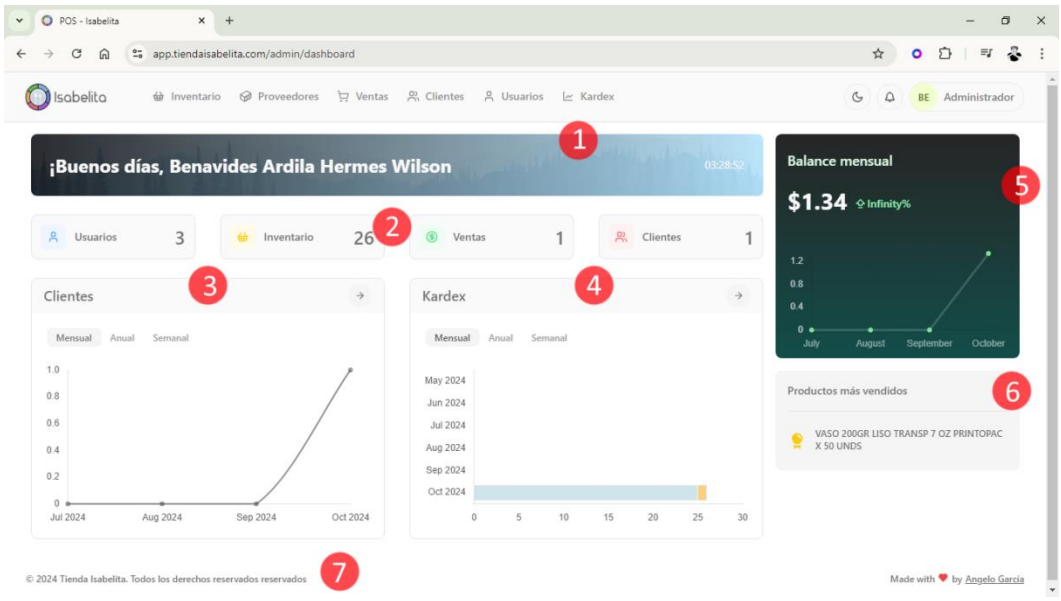
- Longitud mínima de 8 caracteres.
- Debe contener al menos una letra mayúscula.
- Debe incluir al menos una letra minúscula.
- Debe tener al menos un número.
- Debe incluir al menos un carácter especial (como @, #, $, &, etc.).
- Recuerde que una contraseña segura ayuda a proteger su cuenta.



Interfaz de usuario

Descripción del panel principal

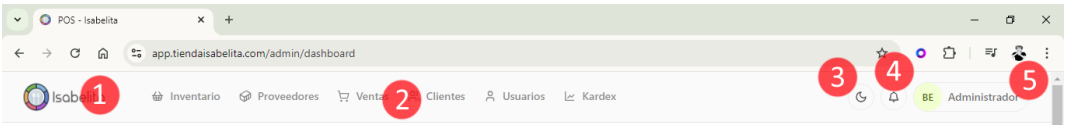
Al iniciar sesión, será dirigido al dashboard o panel de control. La redirección a las diferentes secciones del sistema se realiza en base al rol del usuario.



1. **Widget de saludo:** Un banner que muestra el nombre del usuario de la sesión y la hora actual del sistema.
2. **Widgets de información:** Visualización rápida de datos clave como la cantidad de usuarios, inventario, ventas y clientes.
3. **Métricas de clientes:** Proporciona una visión general de las transacciones realizadas por clientes en las últimas semanas, meses o años, facilitando el análisis de comportamiento y tendencias.
4. **Métricas de kardex:** Presenta un resumen detallado de los movimientos de inventario efectuados en las últimas semanas, meses o años, ofreciendo una visión clara del flujo de productos.
5. **Métricas de balance:** Resumen del total de ventas realizadas durante los últimos 4 meses, ayudando a monitorear el rendimiento financiero a corto plazo. El porcentaje de ganancia o pérdida se obtiene en base al mes anterior.
6. **Top 3 productos/servicios:** Visualización de los 3 productos o servicios más vendidos, brindando información valiosa sobre los artículos de mayor demanda.
7. **Pie de página:** Es una sección ubicada en la parte inferior de la pantalla que muestra el nombre del establecimiento. Este widget proporciona información institucional visible en todo momento, reforzando la identidad del negocio y ofreciendo un punto de referencia constante.

Barra de navegación

El menú principal, situado a la parte superior de la pantalla, incluye las siguientes opciones:



1. **Logo del establecimiento:** Situado en la parte izquierda de la barra superior, el logo representa la identidad visual del establecimiento y permanece visible en todas las secciones de la aplicación.

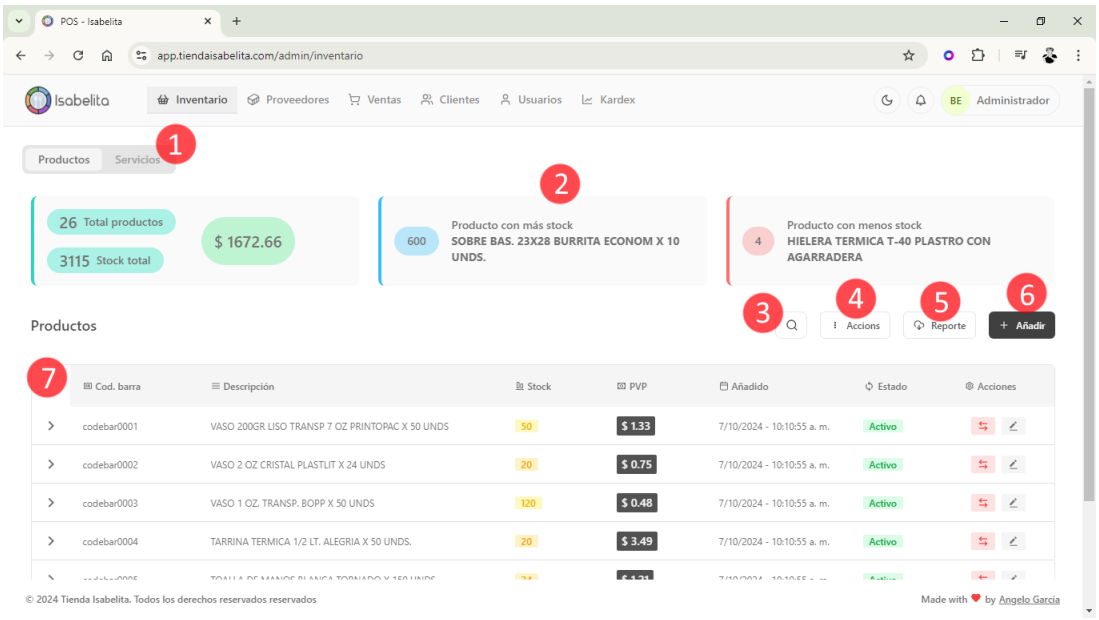
2. **Navegación de módulos:** Ubicada en el centro de la barra superior, esta sección contiene accesos directos a los diferentes módulos del sistema.
3. **Cambio de tema:** Permite alternar entre los modos de visualización (claro/oscuro).
4. **Notificaciones:** Muestra alertas o mensajes importantes.
5. **Menú desplegable:** Incluye opciones adicionales como configuración de cuenta, sistema y cierre de sesión.

Módulos del Sistema

Inventario: Gestión de productos y servicios

Este módulo permite gestionar los productos y servicios que se ofrecen en la empresa, así como el control del stock. Está dividido en dos secciones principales:

5.5.7. Productos



1. **Barra de navegación:** Permite cambiar entre secciones del inventario.
2. **Widgets de Información:** Visualización rápida de datos clave como la total de productos y stock, producto con más y menos stock.
3. **Buscar:** Permite buscar productos por campos de código barras y descripción.
4. **Acciones:** Muestra un menú desplegable con opciones:

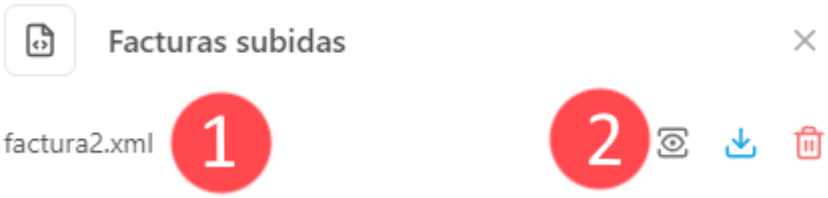
- **Importar códigos de barras**

  Permite cargar en el sistema una lista de códigos de barras de productos. Asegúrate de que el reporte generado contenga los encabezados "Código de producto" y "Código de barras" para facilitar la asignación de códigos. Los códigos de barras duplicados no serán tomados en cuenta, lo que optimiza el proceso de registro y gestión de inventario, permitiendo escanear productos de manera rápida y eficiente.



- **Ver facturas subidas**

  Ofrece acceso a las facturas de la adquisición de los productos que han sido cargadas en el sistema. Los usuarios pueden revisar, consultar y gestionar fácilmente los documentos de facturación asociados a las ventas realizadas.



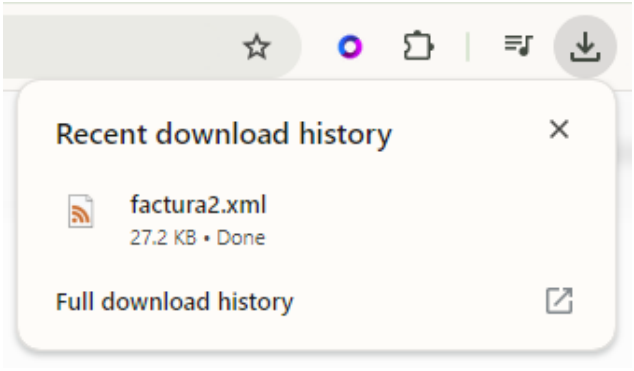
1. **Ver nombre** con el que se subió la factura al sistema.
2. **Acciones**
    - **Ver detalles de factura**

      El botón muestra un modal con el contenido de la factura en formato XML.



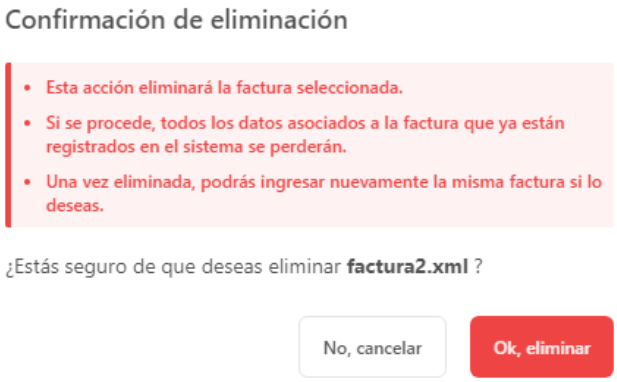
    - **Descargar factura**

      El botón descarga la factura.



    - **Eliminar factura**

      El botón muestra un modal con información y botones de acciones.



5. **Reportes:** Permite realizar reportes, con filtros como seleccionando las columnas, fechas y estado.
6. **Añadir:** Permite registrar nuevos productos en el sistema, mediante una factura en formato XML proporcionada por el SRI.



7. **Tabla de productos:** Este módulo presenta una tabla que contiene información esencial sobre los productos disponibles en el inventario.



1. **Encabezado**: Esta sección permite organizar la información de la tabla de productos. Al pasar el cursor sobre los títulos de las columnas, los usuarios pueden ordenar los datos de acuerdo con sus necesidades, facilitando la búsqueda y el análisis de la información de los productos.
2. **Acciones:** En esta sección, los usuarios tienen acceso a tres botones que permiten realizar acciones sobre los productos:

   - **Dar de baja:** Permite dar de baja un producto del inventario, marcándolo como inactivo.



   - **Reactivar:** Opción para restablecer un producto que ha sido dado de baja, devolviéndolo a un estado activo.

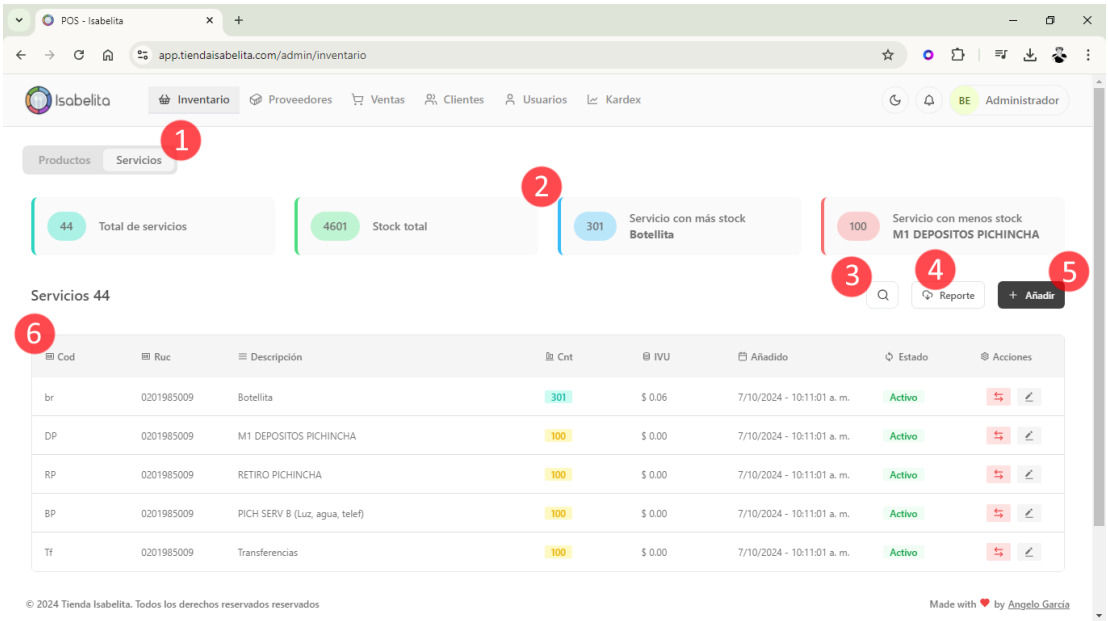


   - **Editar:** Permite modificar la información del producto seleccionado, asegurando que los datos estén actualizados y sean precisos.



3. **Detalles:** Esta sección proporciona información adicional sobre cada producto, permitiendo a los usuarios obtener una visión completa y detallada de cada artículo en el inventario.

5.5.8. Servicios

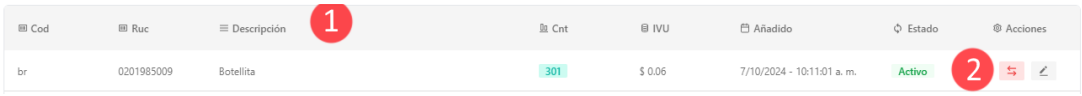


1. **Barra de navegación:** Permite cambiar entre secciones del inventario.

2. **Widgets de Información:** Visualización rápida de datos clave como la total de servicios y stock, servicio con más y menos stock.
3. **Buscar:** Permite buscar servicios por campos de código, ruc y descripción.
4. **Reportes:** Permite realizar reportes, con filtros como seleccionando las columnas, fechas y estado.
5. **Añadir:** Permite registrar nuevos servicios en el sistema, mediante un archivo en formato XLSX.

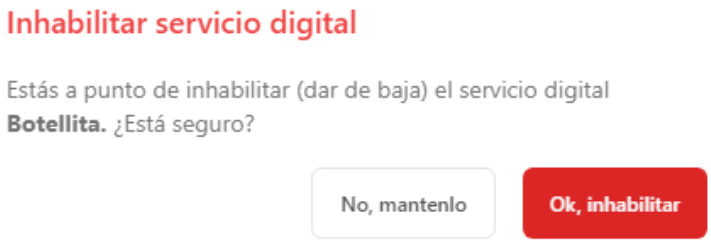


6. **Tabla de productos:** Este módulo presenta una tabla que contiene información esencial sobre los servicios disponibles en el inventario.



1. **Encabezado**: Esta sección permite organizar la información de la tabla de servicios. Al pasar el cursor sobre los títulos de las columnas, los usuarios pueden ordenar los datos de acuerdo con sus necesidades, facilitando la búsqueda y el análisis de la información de los servicios.
2. **Acciones:** En esta sección, los usuarios tienen acceso a tres botones que permiten realizar acciones sobre los productos:

- **Dar de baja:** Permite dar de baja un servicio del inventario, marcándolo como inactivo.



- **Reactivar:** Opción para restablecer un servicio que ha sido dado de baja, devolviéndolo a un estado activo.

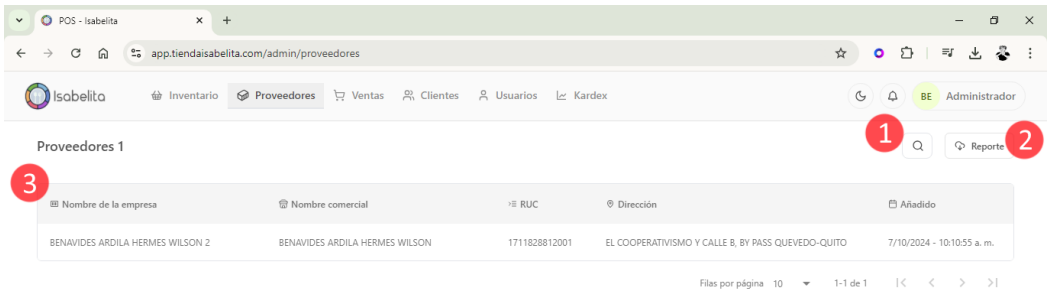


- **Editar:** Permite modificar la información del servicio seleccionado, asegurando que los datos estén actualizados y sean precisos.



Proveedores: Administración de datos de proveedores

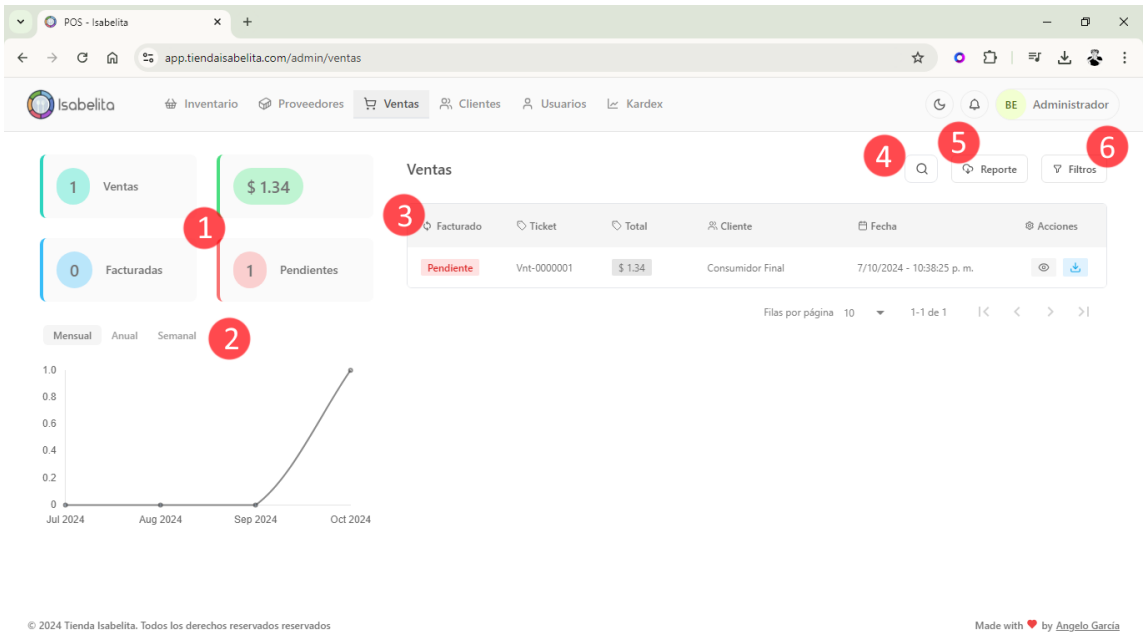
Este módulo permite la gestión de la información relacionada con los proveedores, dichos proveedores se registran al momento que se registran productos mediante la factura XML.



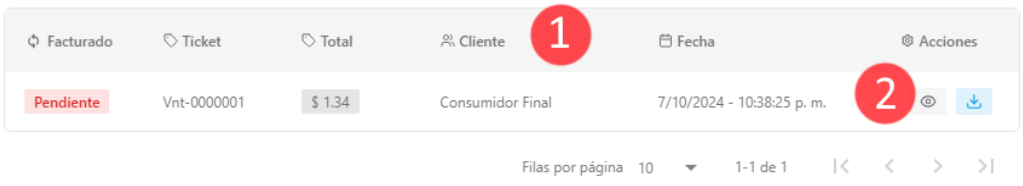
1. **Buscar:** Permite buscar servicios por campos de código, ruc y descripción.
2. **Reportes:** Permite realizar reportes, con filtros como seleccionando las columnas, fechas y estado.
3. **Tabla de proveedores:** Este módulo presenta una tabla que contiene información esencial sobre los proveedore disponibles.

Ventas: Control de transacciones y facturación

Este módulo está enfocado en la gestión de las ventas realizadas, permitiendo a los usuarios realizar un seguimiento de las transacciones, ver y generar facturas.



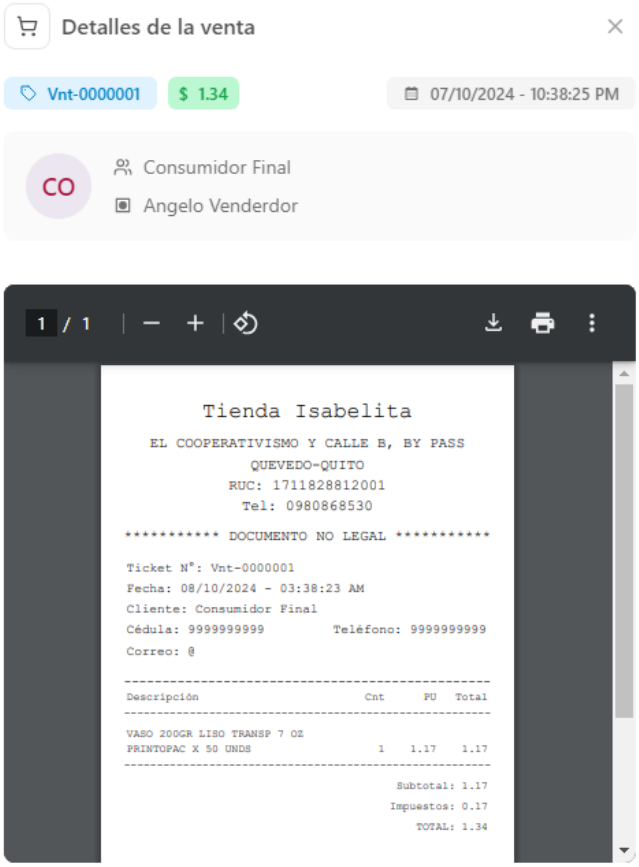
1. **Widgets de Información:** Visualización rápida de datos clave como la total de clientes y total monetario de ventas y facturas pendientes o facturadas.
2. **Métrica:** Proporciona una visión general de las ventas por en las últimas semanas, meses o años, facilitando el análisis de comportamiento y tendencias.
3. **Tabla de productos:** Este módulo presenta una tabla que contiene información esencial sobre los servicios disponibles en el inventario.



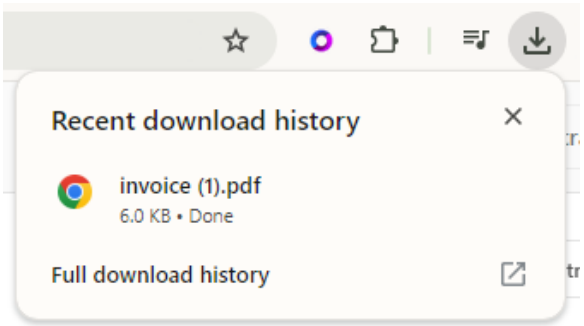
- **Encabezado**: Esta sección permite organizar la información de la tabla de ventas. Al pasar el cursor sobre los títulos de las columnas, los usuarios pueden ordenar los datos de acuerdo con sus necesidades, facilitando la búsqueda y el análisis de la información de los servicios.

- **Acciones:** En esta sección, los usuarios tienen acceso a dos botones que permiten realizar acciones sobre las ventas:

  -  **Ver detalles:** Permite ve los detalles sobre la venta.



  -  **Descargar:** Descargar el recibo de la venta en formato PDF.

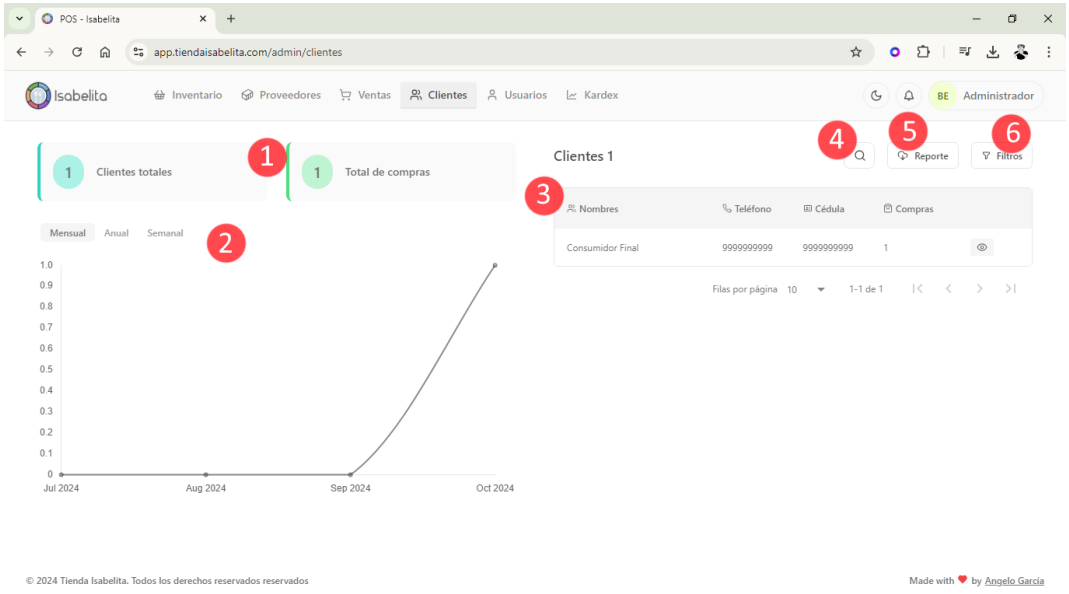


4. **Buscar:** Permite buscar venta por campos de ticket y cliente.

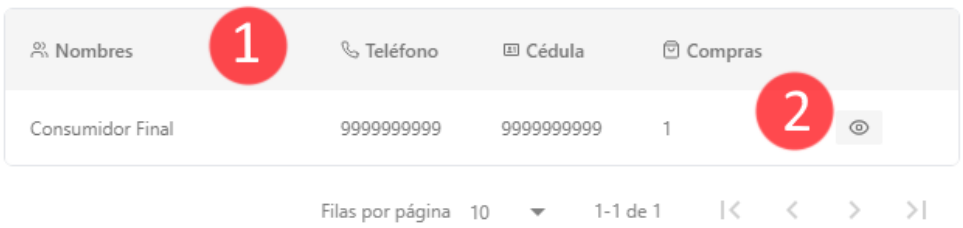
5. **Reportes:** Permite realizar reportes, con filtros como seleccionando las columnas, fechas.
6. **Filtros avanzados:** Permite hacer un filtrado avanzado por fechas, tipo cliente y estado de facturas.

Clientes: Información y gestión de clientes

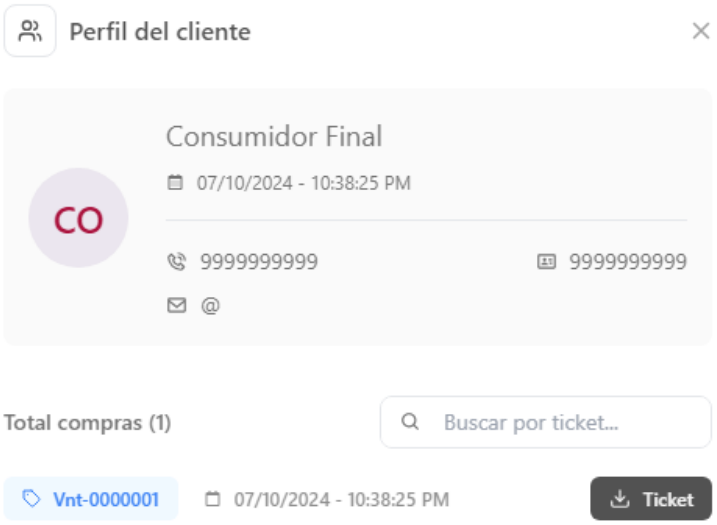
En el módulo de clientes se gestionan todos los datos relacionados con los clientes de la empresa.



1. **Widgets de Información:** Visualización rápida de datos clave como la total de clientes y compras.
2. **Métrica:** Proporciona una visión general de los clientes por en las últimas semanas, meses o años, facilitando el análisis de comportamiento y tendencias.
3. **Tabla de clientes:** Este módulo presenta una tabla que contiene información esencial sobre los clientes.



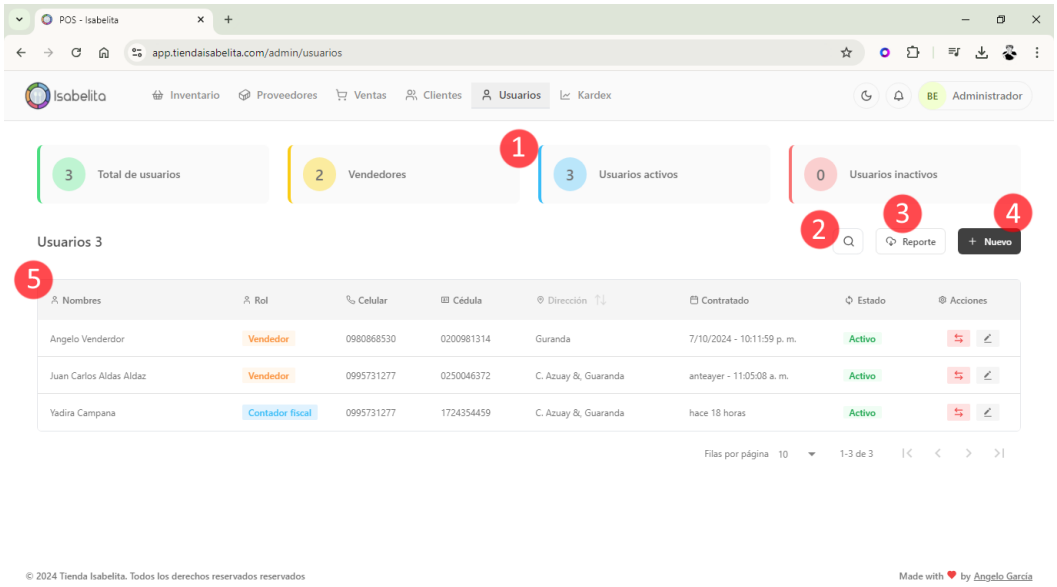
- **Encabezado**: Esta sección permite organizar la información de la tabla de ventas. Al pasar el cursor sobre los títulos de las columnas, los usuarios pueden ordenar los datos de acuerdo con sus necesidades, facilitando la búsqueda y el análisis de la información de los clientes.
- **Acciones:** En esta sección, los usuarios tienen acceso a un botón que permite realizar esta acción sobre los clientes:
  - **Ver detalles:** Permite ve los detalles sobre los clientes.



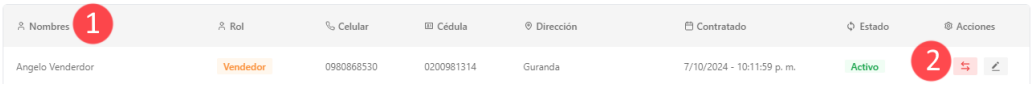
4. **Buscar:** Permite buscar una venta por campos de nombre, teléfono y cliente.
5. **Reportes:** Permite realizar reportes, con filtros como seleccionando las columnas, fechas.
6. **Filtrar:** Permite hacer un filtrado avanzado por fechas y tipo cliente.

Usuarios: Gestión de usuarios del sistema

Este módulo permite la administración de los usuarios que tienen acceso al sistema de gestión de usuarios.

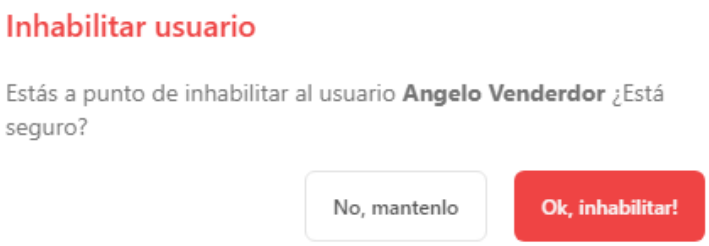


1. **Widgets de Información:** Visualización rápida de datos clave como la total de usuarios, vendedores, usuarios activos e inactivos.
2. **Buscar:** Permite buscar venta por campos de nombres, celular, cédula y dirección.
3. **Reportes:** Permite realizar reportes, con filtros como seleccionando las columnas, fechas.
4. **Tabla de usuarios:** Este módulo presenta una tabla que contiene información esencial sobre los usuarios.

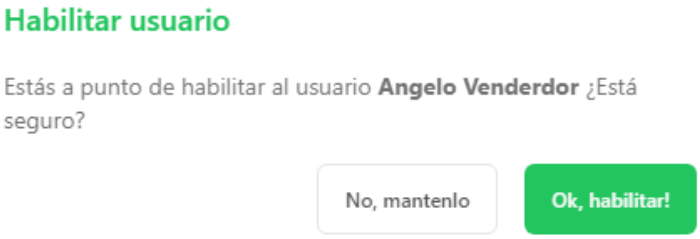


1. **Encabezado**: Esta sección permite organizar la información de la tabla de ventas. Al pasar el cursor sobre los títulos de las columnas, los usuarios pueden ordenar los datos de acuerdo con sus necesidades, facilitando la búsqueda y el análisis de la información de los usuarios.
2. **Acciones:** En esta sección, los usuarios tienen acceso a tres botones que permiten realizar acciones sobre los productos:

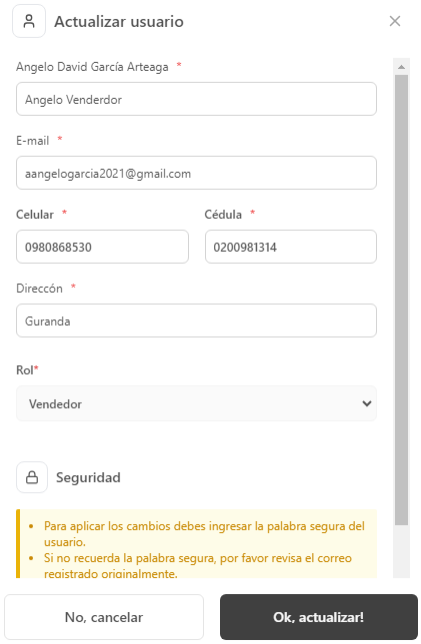
- **Dar de baja:** Permite dar de baja un usuario del sistema, marcándolo como inactivo.



- **Reactivar:** Opción para restablecer un usuario que ha sido dado de baja, devolviéndolo a un estado activo.

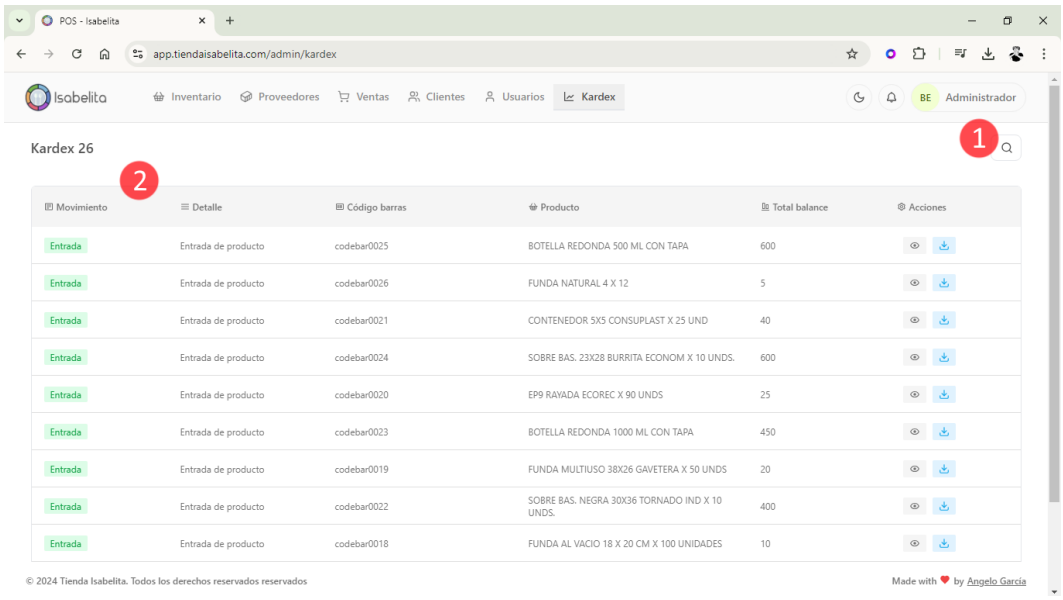


- **Editar:** Permite modificar la información del usuario seleccionado, asegurando que los datos estén actualizados y sean precisos.



4.6. Kardex: Control detallado de movimientos de inventario

Este módulo ofrece un control exhaustivo de los movimientos de inventario, permitiendo hacer un seguimiento detallado de las entradas y salidas de productos.

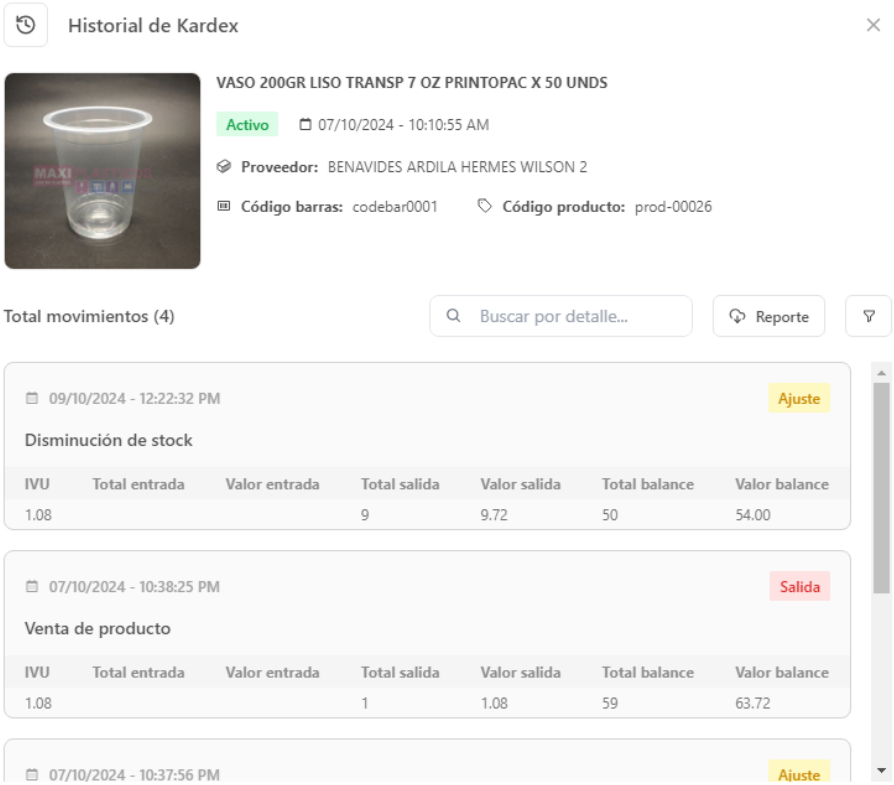


1. **Buscar:** Permite buscar por campos de detalle, producto y código barras.
2. **Tabla de kardex:** Este módulo presenta una tabla que contiene información esencial sobre el Kardex.



- **Encabezado**: Esta sección permite organizar la información de la tabla de ventas. Al pasar el cursor sobre los títulos de las columnas, los usuarios pueden ordenar los datos de acuerdo con sus necesidades, facilitando la búsqueda y el análisis de la información de los usuarios.

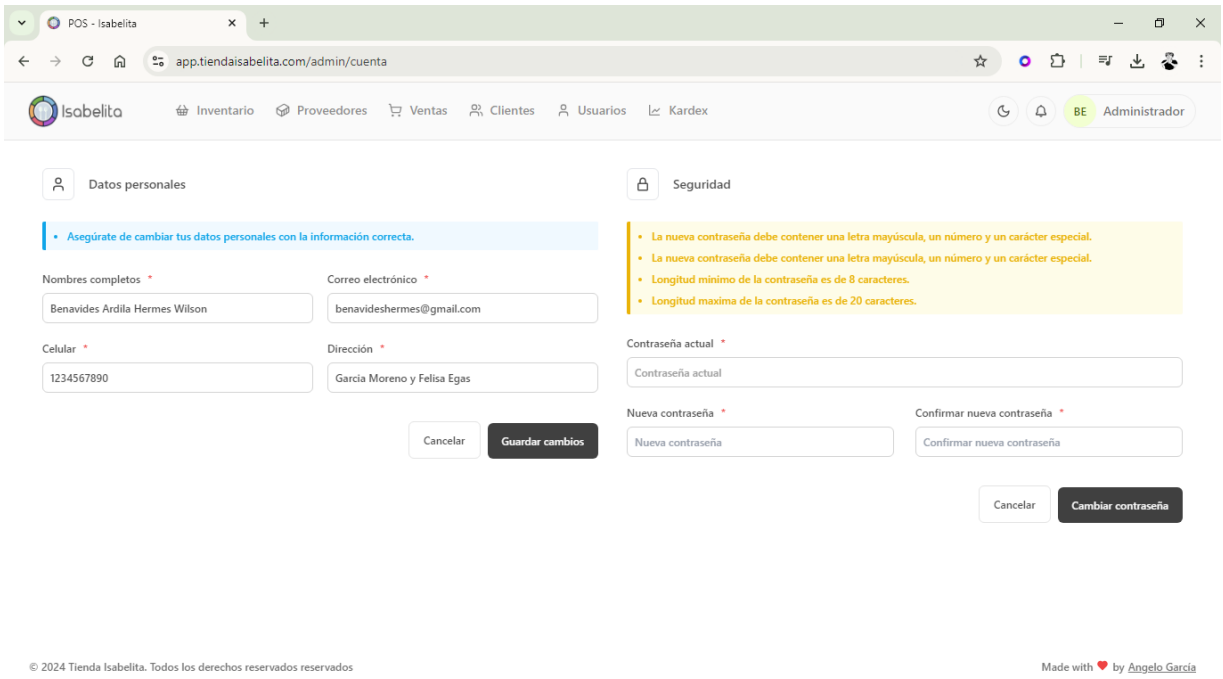
- **Acciones:** En esta sección, los usuarios tienen acceso a dos botones que permiten realizar acciones sobre el kardex:
  - **Ver detalles:** Permite ve los detalles sobre los movimientos del producto seleccionado.



- **Reportes:** Permite realizar reportes, con filtros como seleccionando las columnas, fechas.

4.7. Configuraciones de la cuenta

Esta sección permite a los usuarios gestionar y personalizar la configuración de su cuenta en el sistema. A través de este módulo, se pueden realizar diversas acciones, como:

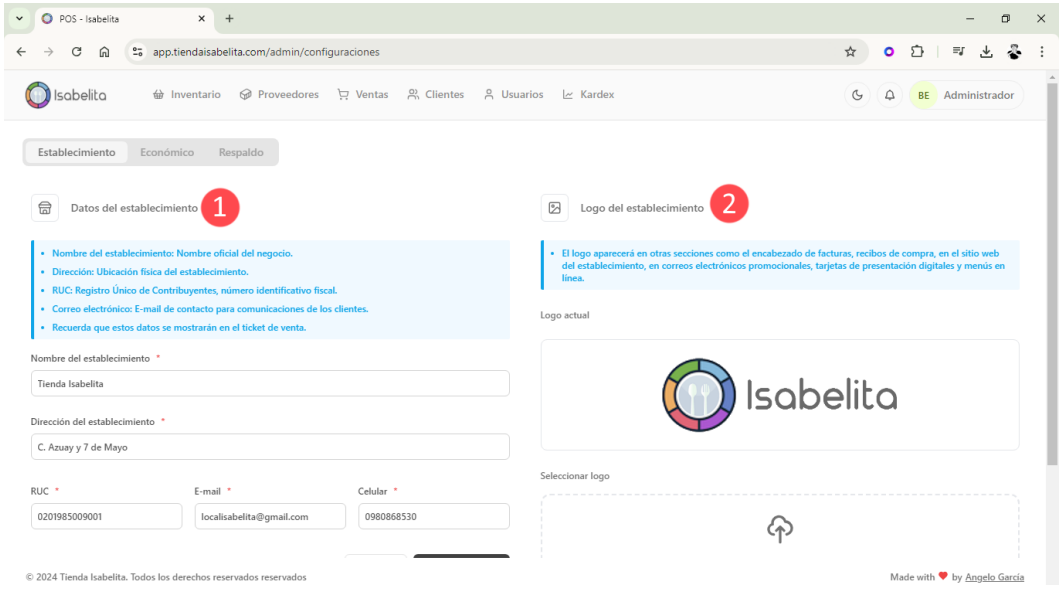


- **Actualizar información personal:** Modificar datos como nombre, correo electrónico y número de contacto
- **Cambiar contraseña:** Permite a los usuarios establecer una nueva contraseña para garantizar la seguridad de su cuenta.

4.8. Configuración del Sistema

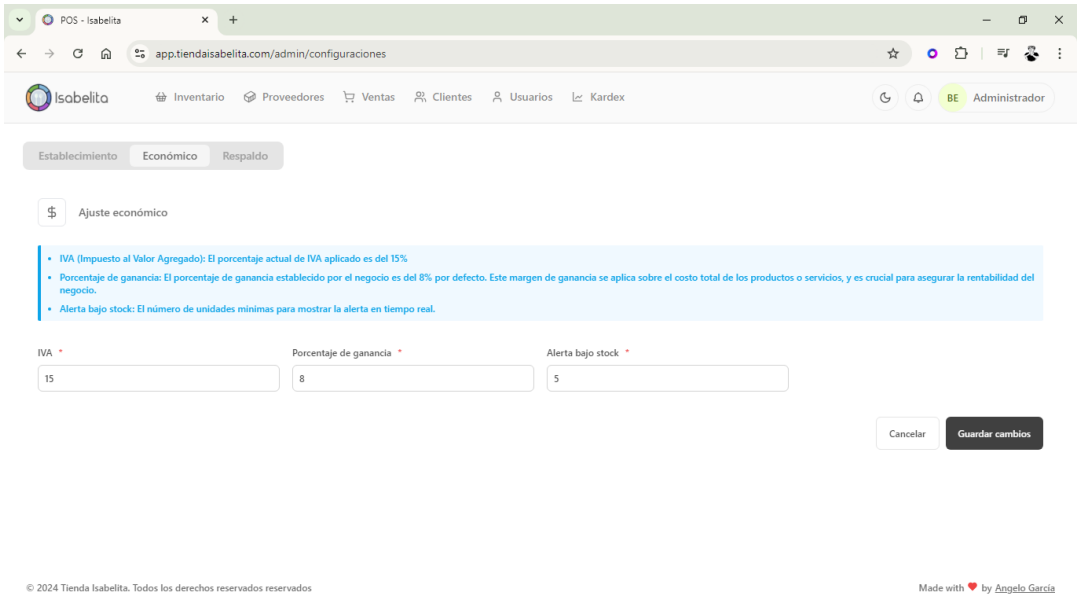
Esta sección permite a los administradores gestionar y personalizar diversas configuraciones del sistema para optimizar su funcionamiento. Está dividida en tres secciones principales:

5.5.9. 4.8.1. Establecimiento



1. Aquí se pueden ingresar y editar los datos del establecimiento, incluyendo nombre, dirección, y detalles de contacto, etc.
2. También se puede cargar el **logo del establecimiento**, que se utilizará en las facturas y reportes, asegurando que la marca esté adecuadamente representada en la documentación oficial.

5.5.10. 4.8.2. Económico



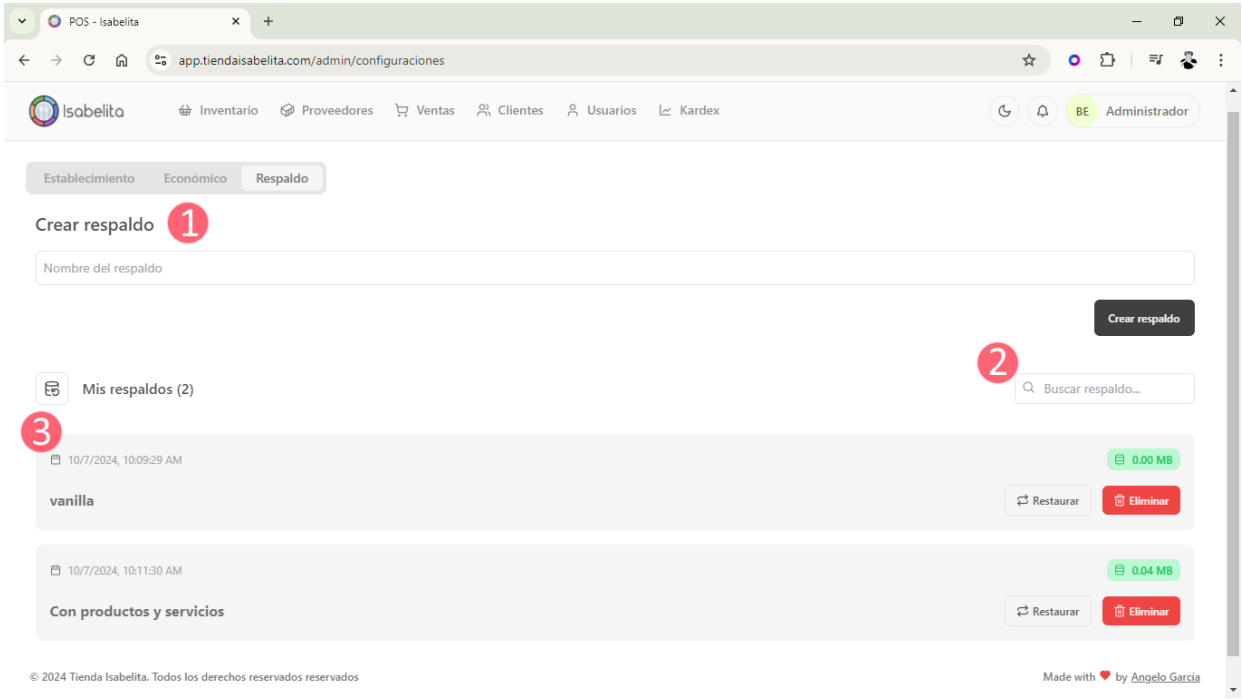
IVA: Establecer el porcentaje de impuesto al valor agregado que se aplicará a las transacciones.

Porcentaje de ganancia: Definir el margen de ganancia que se aplicará a los productos, facilitando el cálculo automático de precios de venta.

Alerta de bajo stock: Configurar notificaciones para alertar a los usuarios cuando los niveles de inventario de un producto caen por debajo de un umbral establecido, asegurando una gestión proactiva del stock.

5.5.11. 4.8.3. Respaldo

En esta sección, los usuarios pueden gestionar copias de seguridad del sistema. Estas copias de seguridad son cruciales para garantizar la seguridad de los datos y permitir la recuperación rápida en caso de pérdida de información o fallos del sistema.



3.1.7.39.4.8.3.1. Crear un respaldo

1. Dirígete a la pestaña de **Respaldo** dentro de la sección de configuración del sistema.
2. En la sección **Crear respaldo**, ingresa el nombre deseado para el respaldo en el campo de texto **Nombre del respaldo**.
3. Haz clic en el botón negro etiquetado como **Crear respaldo** para generar el respaldo.

3.1.7.40.4.8.3.2. Buscar un respaldo

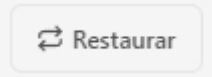
4. Para buscar entre tus respaldos, utiliza el campo de búsqueda **Buscar respaldo**

5. Escribe el nombre o palabras clave relacionadas con el respaldo para filtrar los resultados.

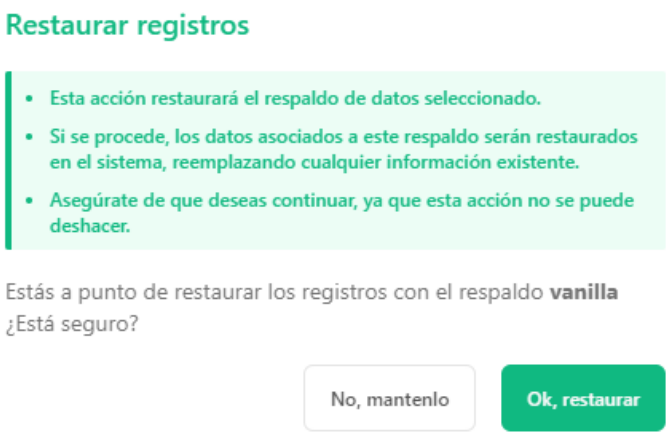
3.1.7.41.4.8.3.3. Gestionar Respaldos

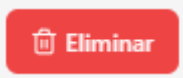
El sistema muestra una lista de tus respaldos en la sección Mis respaldos.

Para cada respaldo, puedes realizar las siguientes acciones:

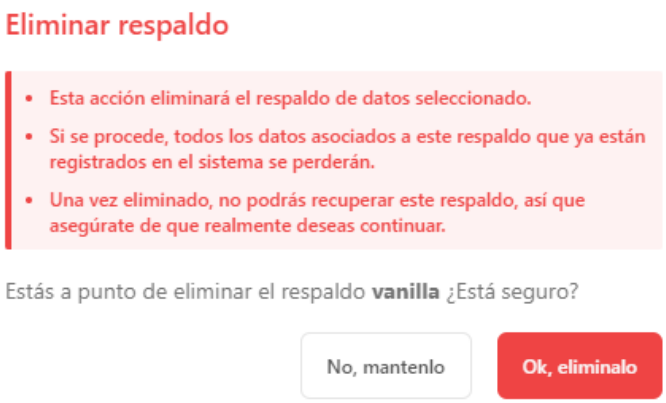
1. 

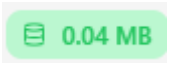
Restaurar: Haciendo clic en este botón, restauras el sistema al punto del respaldo seleccionado.



2. 

Eliminar: Haciendo clic en este botón, eliminas el respaldo seleccionado.





Al lado del botón Restaurar, también puedes ver el tamaño del respaldo en MB.

ANEXO 7

Manual de usuario del Vendedor

Isabelita

Manual de usuario - Vendedor

Sistema de Gestión de Ventas

Juan Aldas
Angelo García

07 de octubre del 2024

1. Introducción 3
1.1. Descripción general del sistema 3
1.2. Objetivo 3
1.3. Audiencia objetivo 3
1.4. Requisitos del sistema 3
2. Acceso al sistema 5
2.1. Inicio de Sesión 5
2.2. Recuperación de contraseña 5
3. Interfaz de usuario 7
3.1. Barra de navegación 7
4. Módulos del Sistema 8
4.1. Panel principal 8
4.1.2. Búsqueda de Productos 8
4.1.3. Agregar a productos 9
4.1.4. Productos Seleccionados 9
4.1.5. Detalles de la compra 10
4.1.6. Proceso de pago paso a paso 13
4.2. Ventas: Control de transacciones y facturación 15
4.3. Clientes: Información y gestión de clientes 17
4.3. Configuraciones de la cuenta 19

Introducción

Descripción general del sistema

El Sistema de Gestión de Ventas es una plataforma web diseñada para optimizar y simplificar el control de ventas, clientes, productos e inventarios, especialmente enfocada en empresas dedicadas a la venta de productos desechables. Este sistema permite registrar, gestionar y consultar ventas en tiempo real, generando informes detallados que facilitan un análisis eficiente del desempeño comercial. Con esta herramienta, las empresas pueden mejorar su toma de decisiones, asegurando una mayor eficiencia en la gestión de su inventario y aumentando la satisfacción del cliente.

Objetivo

El principal objetivo de este sistema es automatizar el proceso de ventas, incrementando la precisión en los registros, minimizando errores humanos y ofreciendo herramientas que permiten monitorear y optimizar la toma de decisiones empresariales. Esto contribuye a una gestión más eficiente de las operaciones diarias, asegurando un mayor control y rendimiento en las ventas.

Audiencia objetivo

Este manual está dirigido a:

- Vendedores que deben registrar y gestionar ventas diarias.

Requisitos del sistema

Para utilizar este sistema de manera óptima, se requieren los siguientes recursos:

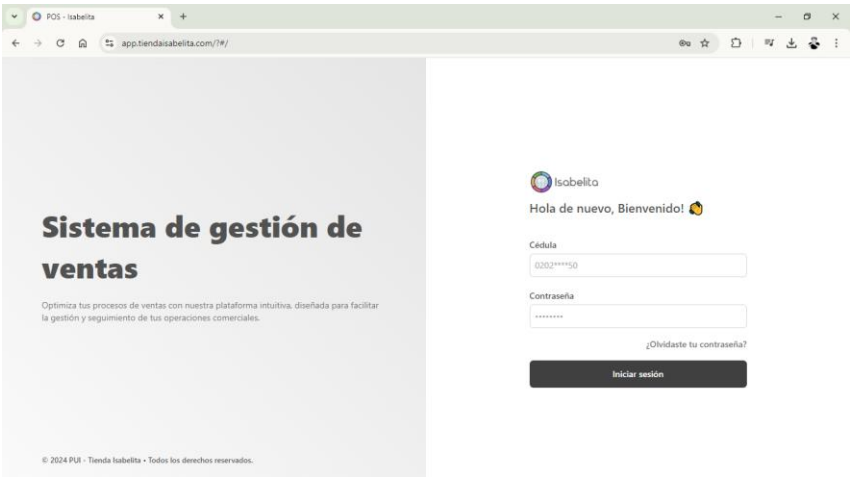
- **Navegador web:** Compatible con Google Chrome, Mozilla Firefox, Safari o Microsoft Edge (es recomendable tener la última versión instalada para garantizar la mejor experiencia).
- **Conexión a internet:** Velocidad mínima recomendada de 5 Mbps para una navegación fluida y sin interrupciones.

- **Resolución de pantalla:** Mínimo 1366x768, aunque se recomienda 1920x1080 para una visualización óptima de los gráficos y paneles.
- **Hardware:** Procesador de al menos 2 GHz y 4 GB de memoria RAM (se recomienda 8 GB para un rendimiento más eficiente en multitareas).
- **Sistema operativo:** Windows 10, macOS 10.14 o superior, o distribuciones de Linux compatibles.
- **Software adicional:** JavaScript habilitado en el navegador y cookies activas para una correcta operación del sistema.
- **Dispositivo compatible:** Se puede acceder desde computadoras de escritorio, portátiles, tabletas y smartphones, siempre que cumplan con los requisitos anteriores.

Acceso al sistema

Inicio de Sesión

Para acceder al sistema, ingrese su correo cédula y contraseña en la página principal haga clic en “Iniciar sesión”. Si las credenciales son correctas, será redirigido al panel principal.

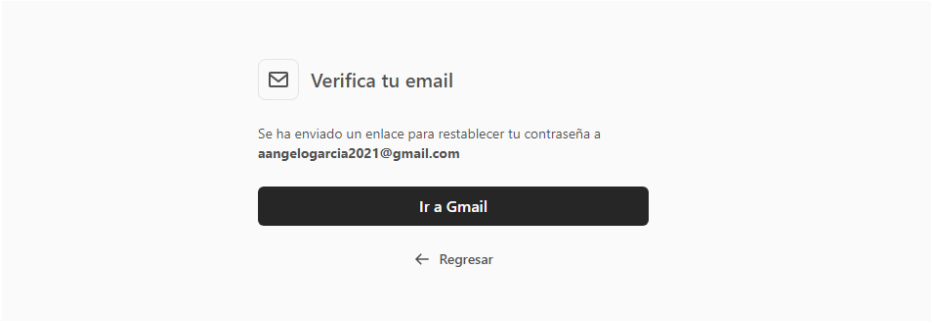


Recuperación de contraseña

Si ha olvidado su contraseña, seleccione “¿Olvidaste tu contraseña?”



Y siga las instrucciones para recibir un correo electrónico con un enlace de recuperación.





Haga clic en el enlace para crear una nueva contraseña. Asegúrese de que la nueva contraseña cumpla con los siguientes requisitos:

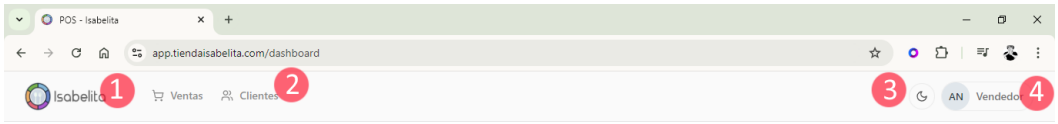
- Longitud mínima de 8 caracteres.
- Debe contener al menos una letra mayúscula.
- Debe incluir al menos una letra minúscula.
- Debe tener al menos un número.
- Debe incluir al menos un carácter especial (como @, #, $, &, etc.).
- Recuerde que una contraseña segura ayuda a proteger su cuenta.



Interfaz de usuario

Barra de navegación

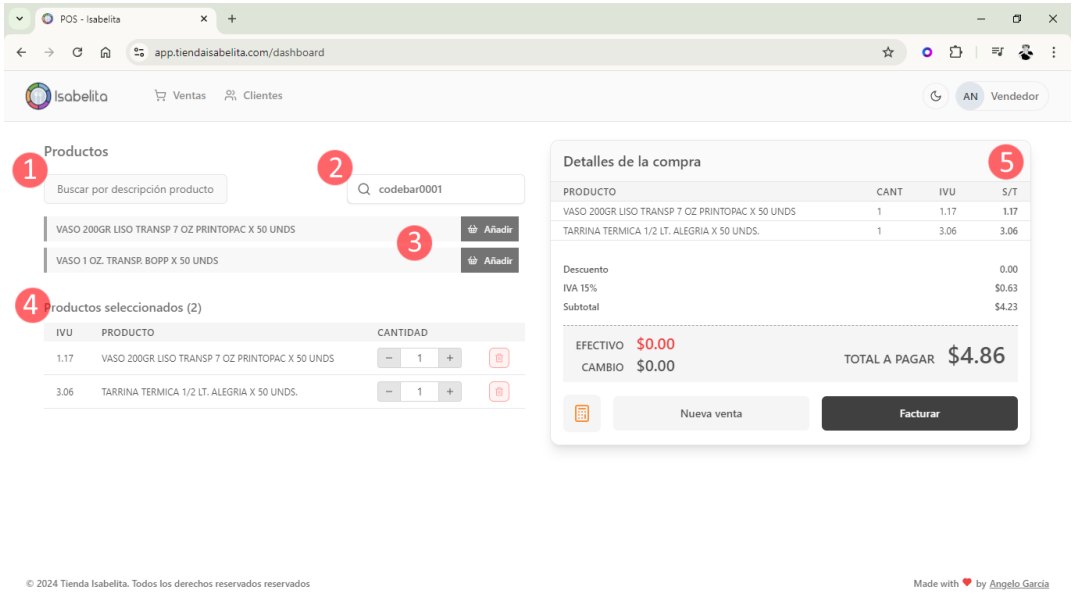
El menú principal, situado a la parte superior de la pantalla, incluye las siguientes opciones:



6. **Logo del establecimiento:** Situado en la parte izquierda de la barra superior, el logo representa la identidad visual del establecimiento y permanece visible en todas las secciones de la aplicación.
7. **Navegación de módulos:** Ubicada en el centro de la barra superior, esta sección contiene accesos directos a los diferentes módulos del sistema.
8. **Cambio de tema:** Permite alternar entre los modos de visualización (claro/oscuro).
9. **Menú desplegable:** Incluye opciones adicionales como configuración de cuenta y cierre de sesión.

Módulos del Sistema

Panel principal



5.5.12. 4.1.2. Búsqueda de Productos

- **Buscar por descripción del producto:** Introduce el nombre o descripción de un producto/servicio para buscarlo en la base de datos del inventario.
- **Buscar por código de barras:** Si cuentas con el código de barras del producto, introdúcelo en este campo para buscarlo rápidamente.



5.5.13. 4.1.3. Agregar a productos

Tras buscar el producto, aparecerá una lista de resultados si existe más de una coincidencia justo debajo del campo de búsqueda.

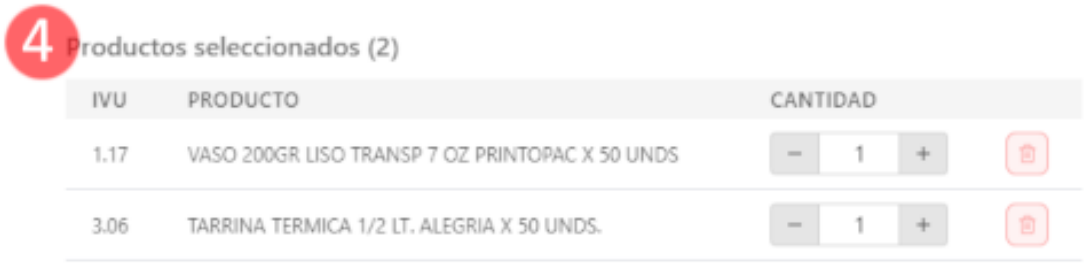




Haz clic en el botón con el ícono de carrito para agregar el producto seleccionado a la lista de productos seleccionados

5.5.14. 4.1.4. Productos Seleccionados

En esta sección se muestra el detalle de los productos que has seleccionado para la venta.



Una vez que los productos han sido seleccionados, el sistema permite ajustar la cantidad de cada producto o eliminarlo.

Botones disponibles para cada producto seleccionado:

: Reduce la cantidad del producto seleccionado en una unidad. Si la cantidad llega a 1, no puedes reducirla más.



: Campo para ingresar la cantidad manualmente. Haz clic en el campo de cantidad y escribe el número deseado para ajustar cuántas unidades del producto quieres añadir. No puedes ingresar unidades que sobrepasen el stock del producto

Aumenta la cantidad del producto seleccionado en una unidad. No puedes ingresar unidades que sobrepasen el stock del producto

Elimina el producto seleccionado de la lista. Haz clic en el icono de la papelera para eliminar el producto por completo.

5.5.15. 4.1.5. Detalles de la compra

En la sección Detalles de la compra, puedes revisar los productos seleccionados, sus cantidades, los impuestos aplicables, descuentos y el total a pagar. Además, desde esta sección puedes gestionar el pago y finalizar la transacción mediante tres botones clave.

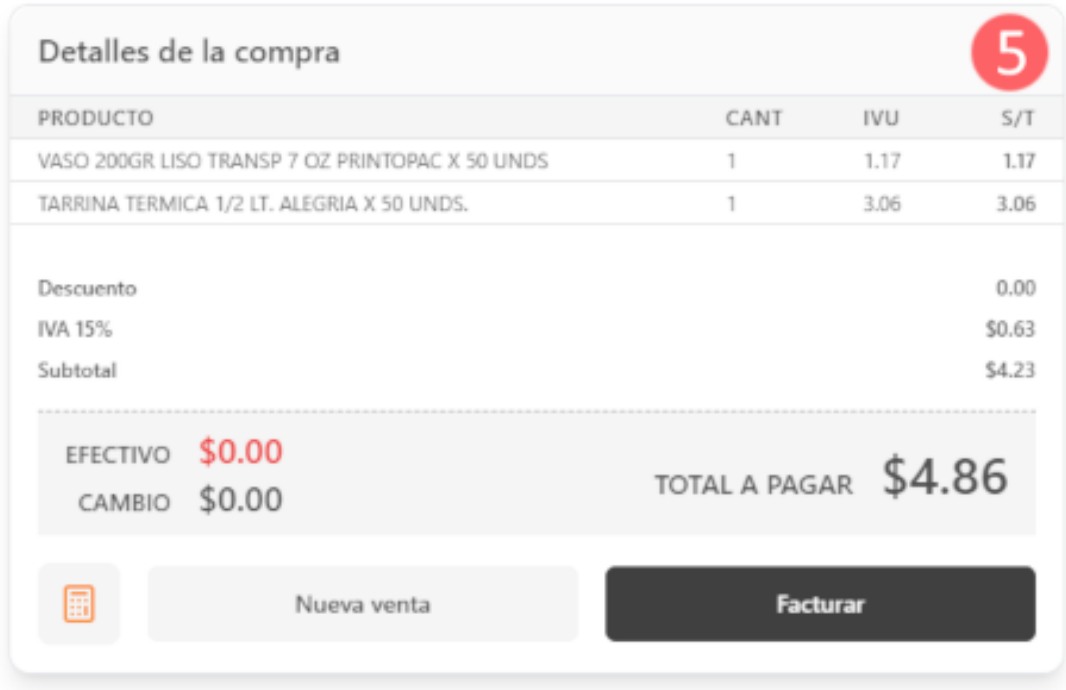


Tabla de productos

- Producto: Nombre del producto agregado al carrito.
- Cant (Cantidad): Número de unidades de cada producto.
- IVU (Impuesto sobre Ventas y Uso): Porcentaje del impuesto aplicado a cada producto.

- S/T (Subtotal): El valor total del producto sin impuestos.

Campos y cálculos importantes

En esta sección se resumen los montos finales de la compra, incluyendo descuentos, impuestos, el total a pagar y el cambio que debe darse al cliente si el pago es en efectivo. A continuación, te explicamos el significado de cada uno de los campos y cómo gestionarlos.

1. **Resumen de precios finales**



Descuento

- **Descripción:** Muestra el monto de descuento aplicado a la compra.
- **Uso:** Si se han aplicado descuentos a algún producto o al total de la compra, aparecerán reflejados aquí. En este ejemplo, no se ha aplicado ningún descuento.
- **Ejemplo:** $0.00 (No se ha aplicado ningún descuento).

IVA 15%

- **Descripción:** se aplica sobre el subtotal de la compra.
- **Uso:** El sistema calcula automáticamente este valor basándose en el subtotal antes de impuestos. El porcentaje puede variar dependiendo de la normativa fiscal.
- **Ejemplo:** $0.87 (Este es el 15% del subtotal).

Subtotal

- **Descripción:** Es la suma del costo de todos los productos seleccionados antes de aplicar el IVA.
- **Uso:** Refleja el costo total de los productos antes de cualquier impuesto o descuento.
- **Ejemplo:** $5.82 (Este es el total de los productos seleccionados).

2. **Tarjeta de precios finales**



Efectivo

- **Descripción:** Muestra la cantidad de dinero en efectivo que el cliente ha entregado para pagar la compra.
- **Uso:** Este campo se debe completar después de que el cliente entregue el efectivo. Puede ser llenado manualmente en el sistema mediante el botón de calculadora o al ingresar el monto recibido.
- **Ejemplo:** $10.00 (El cliente ha pagado $10 en efectivo).

Cambio

- **Descripción:** El sistema calcula automáticamente la cantidad de cambio que se debe devolver al cliente, basado en la diferencia entre el monto total a pagar y el efectivo recibido.
- **Uso:** Este valor se ajusta automáticamente cuando se ingresa el efectivo.
- **Ejemplo:** $3.31 (El cliente debe recibir $3.31 de cambio).

Total a pagar

- **Descripción:** Este es el total final de la compra, incluyendo el subtotal de los productos, cualquier descuento aplicado, y el IVA.

- **Uso:** Refleja el monto que el cliente debe pagar por la compra después de aplicar impuestos y descuentos.
- **Ejemplo:** $6.69 (Este es el total que debe pagar el cliente).

5.5.16. 4.1.6. Proceso de pago paso a paso

Verifica los productos y sus cantidades en la tabla de "Detalles de la compra".



Ingreso de efectivo: Usa el botón de calculadora para ingresar la cantidad que el cliente entrega.

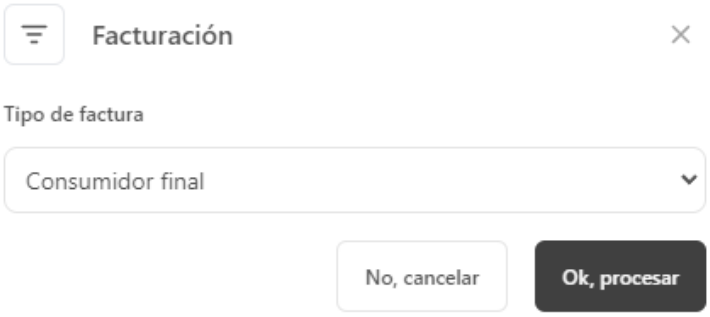


Confirmar la venta

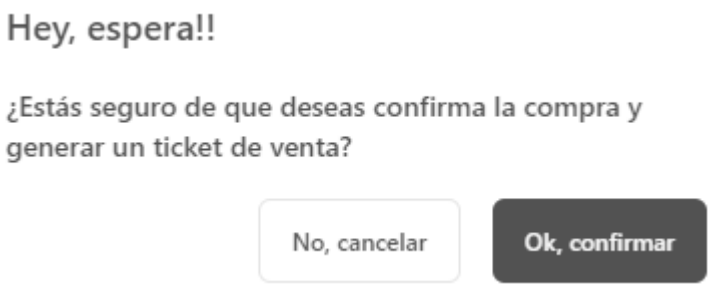


Haz clic en el botón "Facturar".

Selecciona el tipo de venta (Consumidor final o con datos).



Confirma la operación para finalizar la venta.



Impresión del ticket: Al confirmar la venta, el sistema imprimirá automáticamente el ticket.



Tienda Isabelita

C. Azuay y 7 de Mayo
RUC: 0201985009001
Tel: 0980868530

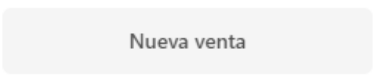
********** DOCUMENTO NO LEGAL **********

Ticket N°: Vnt-0000003
Fecha: 12/10/2024 - 05:48:44 AM
Cliente: Consumidor Final
Cédula: 9999999999 Teléfono: 9999999999
Correo: @

Descripción	Cnt	PU	Total
VASO 200GR LISO TRANSP 7 OZ PRINTOPAC X 50 UNDS	2	1.17	2.33
TARRINA TERMICA 1/2 LT. ALEGRIA X 50 UNDS.	1	3.06	3.06
VASO 1 OZ. TRANSP. BOPP X 50 UNDS	1	0.42	0.42

Subtotal: 5.82
Impuestos: 0.87
TOTAL: 6.69
Efectivo: 10.00
Cambio: 3.31

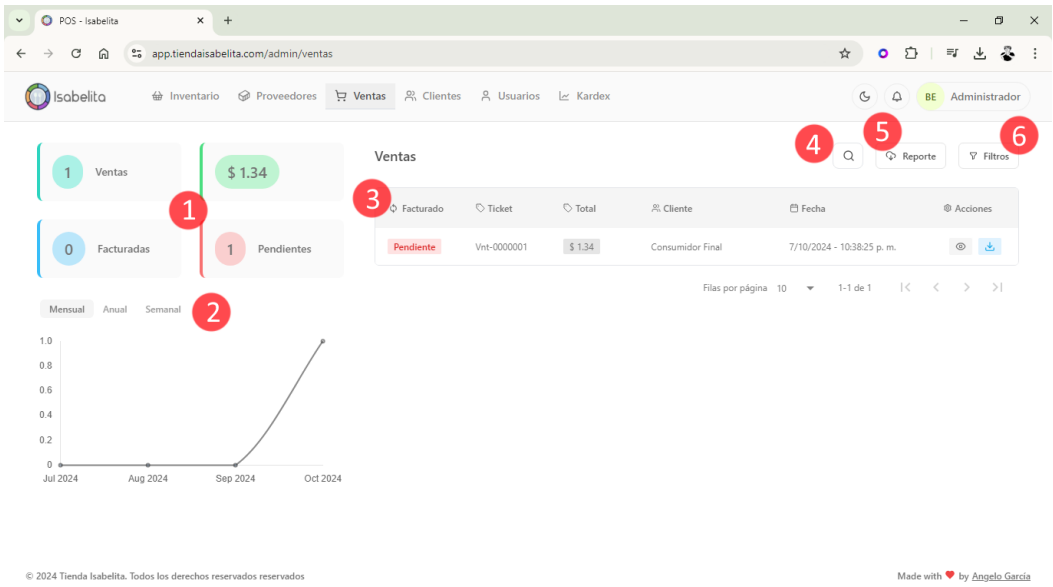
Iniciar una nueva venta



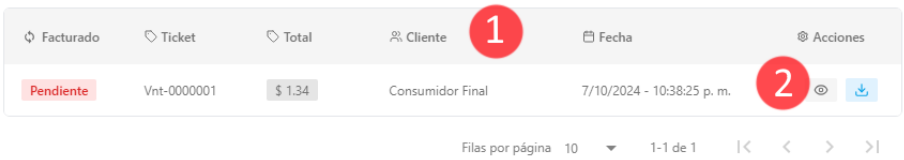
Usa el botón "Nueva venta" para comenzar un nuevo proceso de compra. Está acción borrara todos los productos/servicios y precios.

Ventas: Control de transacciones y facturación

Este módulo está enfocado en la gestión de las ventas realizadas, permitiendo a los usuarios realizar un seguimiento de las transacciones, ver y generar facturas.



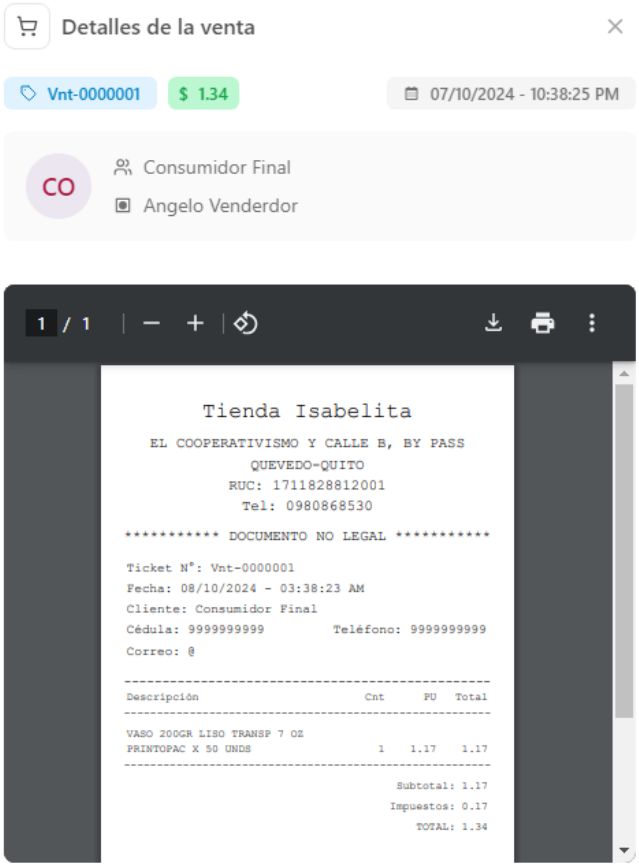
7. **Widgets de Información:** Visualización rápida de datos clave como la total de clientes y total monetario de ventas y facturas pendientes o facturadas.
8. **Métrica:** Proporciona una visión general de las ventas por en las últimas semanas, meses o años, facilitando el análisis de comportamiento y tendencias.
9. **Tabla de productos:** Este módulo presenta una tabla que contiene información esencial sobre los servicios disponibles en el inventario.



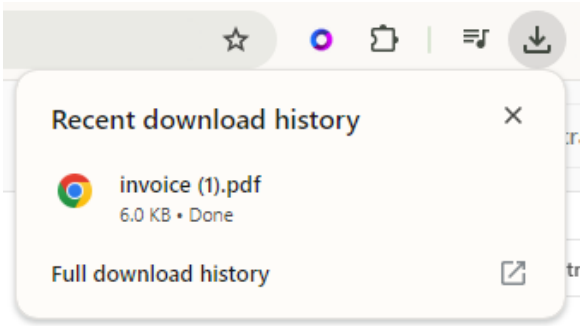
- **Encabezado**: Esta sección permite organizar la información de la tabla de ventas. Al pasar el cursor sobre los títulos de las columnas, los usuarios pueden ordenar los datos de acuerdo con sus

necesidades, facilitando la búsqueda y el análisis de la información de los servicios.

- **Acciones:** En esta sección, los usuarios tienen acceso a dos botones que permiten realizar acciones sobre las ventas:
  - **Ver detalles:** Permite ve los detalles sobre la venta.



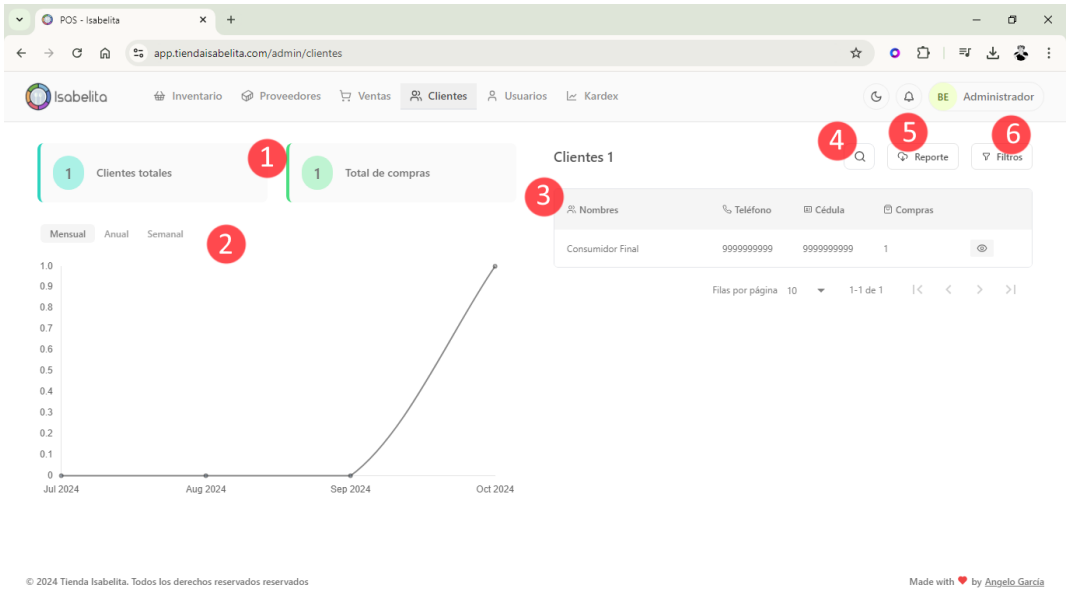
  - **Descargar:** Descargar el recibo de la venta en formato PDF.



10. **Buscar:** Permite buscar venta por campos de ticket y cliente.
11. **Reportes:** Permite realizar reportes, con filtros como seleccionando las columnas, fechas.
12. **Filtros avanzados:** Permite hacer un filtrado avanzado por fechas, tipo cliente y estado de facturas.

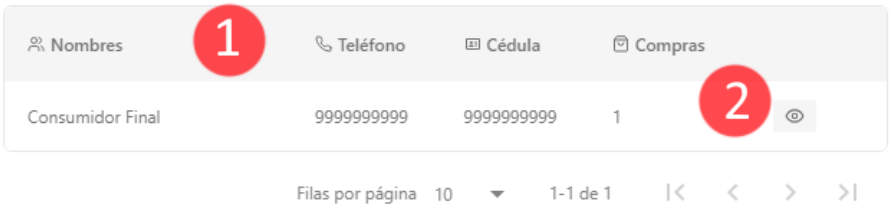
Clientes: Información y gestión de clientes

En el módulo de clientes se gestionan todos los datos relacionados con los clientes de la empresa.

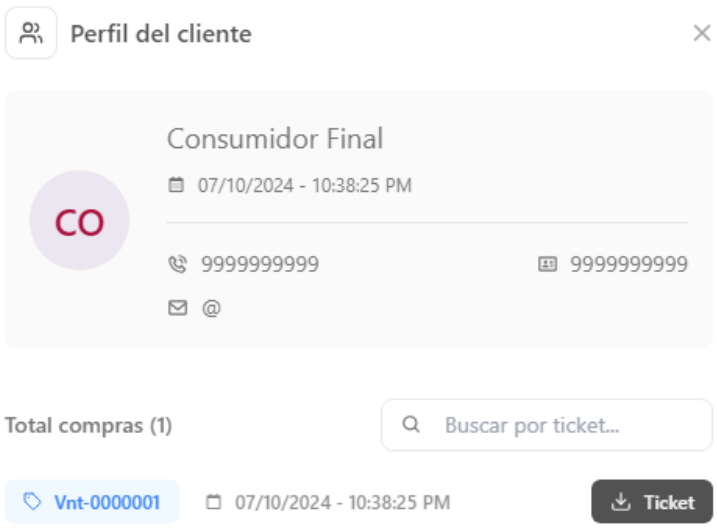


1. **Widgets de Información:** Visualización rápida de datos clave como la total de clientes y compras.

2. **Métrica:** Proporciona una visión general de los clientes por en las últimas semanas, meses o años, facilitando el análisis de comportamiento y tendencias.
3. **Tabla de clientes:** Este módulo presenta una tabla que contiene información esencial sobre los clientes.



1. **Encabezado**: Esta sección permite organizar la información de la tabla de ventas. Al pasar el cursor sobre los títulos de las columnas, los usuarios pueden ordenar los datos de acuerdo con sus necesidades, facilitando la búsqueda y el análisis de la información de los clientes.
2. **Acciones:** En esta sección, los usuarios tienen acceso a un botón que permite realizar esta acción sobre los clientes:
    - **Ver detalles:** Permite ve los detalles sobre los clientes.

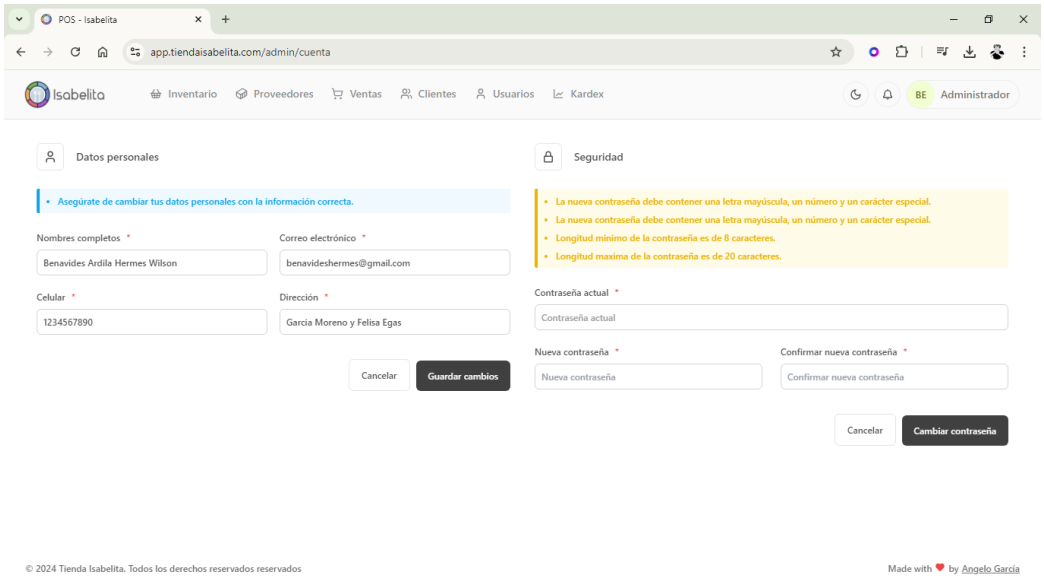


4. **Buscar:** Permite buscar una venta por campos de nombre, teléfono y cliente.

5. **Reportes:** Permite realizar reportes, con filtros como seleccionando las columnas, fechas.
6. **Filtrar:** Permite hacer un filtrado avanzado por fechas y tipo cliente.

4.3. Configuraciones de la cuenta

Esta sección permite a los usuarios gestionar y personalizar la configuración de su cuenta en el sistema. A través de este módulo, se pueden realizar diversas acciones, como:



- **Actualizar información personal:** Modificar datos como nombre, correo electrónico y número de contacto
- **Cambiar contraseña:** Permite a los usuarios establecer una nueva contraseña para garantizar la seguridad de su cuenta.

ANEXO 8

Manual de usuario del Contador

Isabelita

Manual de usuario - Contador

Sistema de Gestión de Ventas

Juan Aldas
Angelo García

07 de octubre del 2024

1. Introducción 3
1.1. Descripción general del sistema 3
1.2. Objetivo 3
1.3. Audiencia objetivo 3
1.4. Requisitos del sistema 3
2. Acceso al sistema 4
2.1. Inicio de Sesión 4
2.2. Recuperación de contraseña 4
3. Interfaz de usuario 6
3.1. Barra de navegación 6
4. Módulos del Sistema 7
4.1. Ventas: Control de transacciones y facturación 7
4.2. Clientes: Información y gestión de clientes 9
4.3. Usuarios: Gestión de usuarios del sistema 11
4.7. Configuraciones de la cuenta 14

Introducción

Descripción general del sistema

El Sistema de Gestión de Ventas es una plataforma web diseñada para optimizar y simplificar el control de ventas, clientes, productos e inventarios, especialmente enfocada en empresas dedicadas a la venta de productos desechables. Este sistema permite registrar, gestionar y consultar ventas en tiempo real, generando informes detallados que facilitan un análisis eficiente del desempeño comercial. Con esta herramienta, las empresas pueden mejorar su toma de decisiones, asegurando una mayor eficiencia en la gestión de su inventario y aumentando la satisfacción del cliente.

Objetivo

El principal objetivo de este sistema es automatizar el proceso de ventas, incrementando la precisión en los registros, minimizando errores humanos y ofreciendo herramientas que permiten monitorear y optimizar la toma de decisiones empresariales. Esto contribuye a una gestión más eficiente de las operaciones diarias, asegurando un mayor control y rendimiento en las ventas.

Audiencia objetivo

Este manual está dirigido a:

- Contadores fiscales que necesitan consultar reportes y analizar el rendimiento general de las ventas y el comportamiento de los clientes.

Requisitos del sistema

Para utilizar este sistema de manera óptima, se requieren los siguientes recursos:

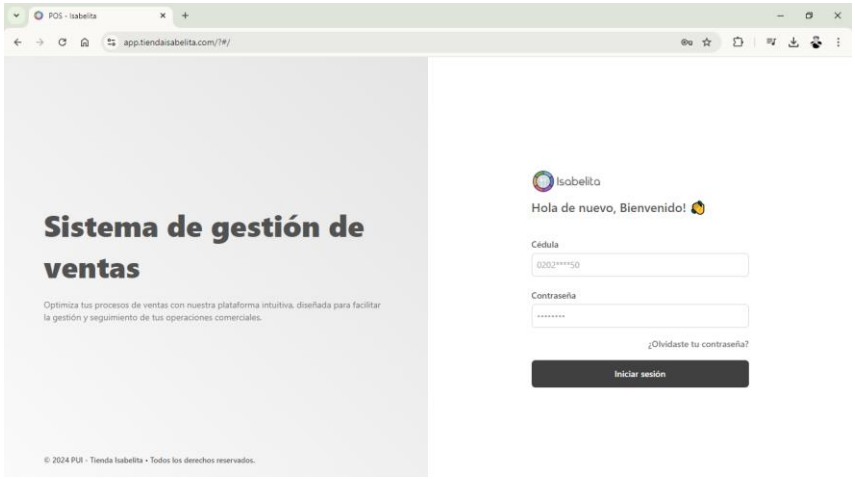
- **Navegador web:** Compatible con Google Chrome, Mozilla Firefox, Safari o Microsoft Edge (es recomendable tener la última versión instalada para garantizar la mejor experiencia).
- **Conexión a internet:** Velocidad mínima recomendada de 5 Mbps para una navegación fluida y sin interrupciones.

- **Resolución de pantalla:** Mínimo 1366x768, aunque se recomienda 1920x1080 para una visualización óptima de los gráficos y paneles.
- **Hardware:** Procesador de al menos 2 GHz y 4 GB de memoria RAM (se recomienda 8 GB para un rendimiento más eficiente en multitareas).
- **Sistema operativo:** Windows 10, macOS 10.14 o superior, o distribuciones de Linux compatibles.
- **Software adicional:** JavaScript habilitado en el navegador y cookies activas para una correcta operación del sistema.
- **Dispositivo compatible:** Se puede acceder desde computadoras de escritorio, portátiles, tabletas y smartphones, siempre que cumplan con los requisitos anteriores.

Acceso al sistema

Inicio de Sesión

Para acceder al sistema, ingrese su correo cédula y contraseña en la página principal haga clic en “Iniciar sesión”. Si las credenciales son correctas, será redirigido al panel principal.

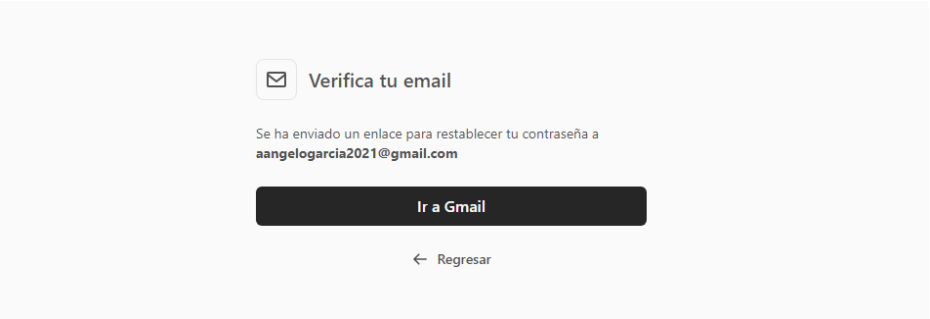


Recuperación de contraseña

Si ha olvidado su contraseña, seleccione “¿Olvidaste tu contraseña?”



Y siga las instrucciones para recibir un correo electrónico con un enlace de recuperación.





Haga clic en el enlace para crear una nueva contraseña. Asegúrese de que la nueva contraseña cumpla con los siguientes requisitos:

- Longitud mínima de 8 caracteres.
- Debe contener al menos una letra mayúscula.
- Debe incluir al menos una letra minúscula.

- Debe tener al menos un número.
- Debe incluir al menos un carácter especial (como @, #, $, &, etc.).
- Recuerde que una contraseña segura ayuda a proteger su cuenta.



Interfaz de usuario

Barra de navegación

El menú principal, situado a la parte superior de la pantalla, incluye las siguientes opciones:

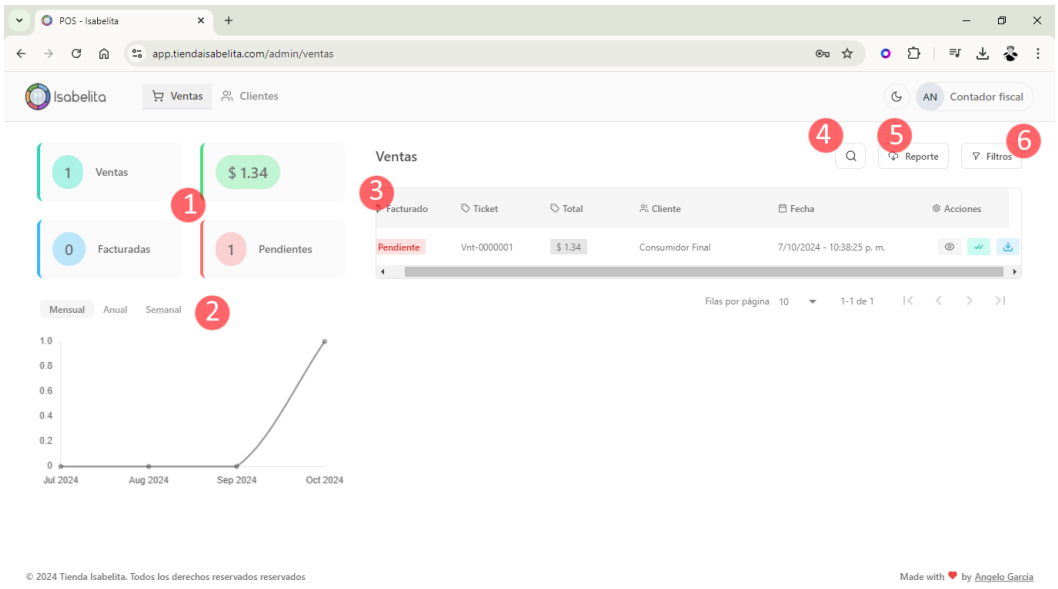


1. **Logo del establecimiento:** Situado en la parte izquierda de la barra superior, el logo representa la identidad visual del establecimiento y permanece visible en todas las secciones de la aplicación.
2. **Navegación de módulos:** Ubicada en el centro de la barra superior, esta sección contiene accesos directos a los diferentes módulos del sistema.
3. **Cambio de tema:** Permite alternar entre los modos de visualización (claro/oscuro).
4. **Menú desplegable:** Incluye opciones adicionales como configuración de cuenta, sistema y cierre de sesión.

Módulos del Sistema

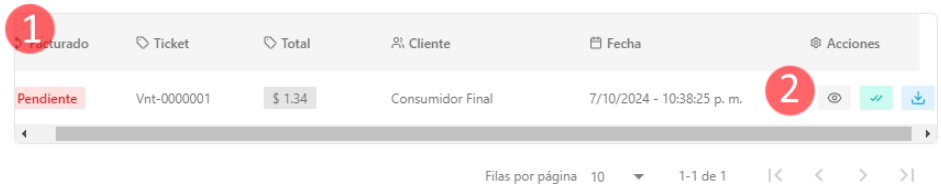
Ventas: Control de transacciones y facturación

Este módulo está enfocado en la gestión de las ventas realizadas, permitiendo a los usuarios realizar un seguimiento de las transacciones, ver y generar facturas.

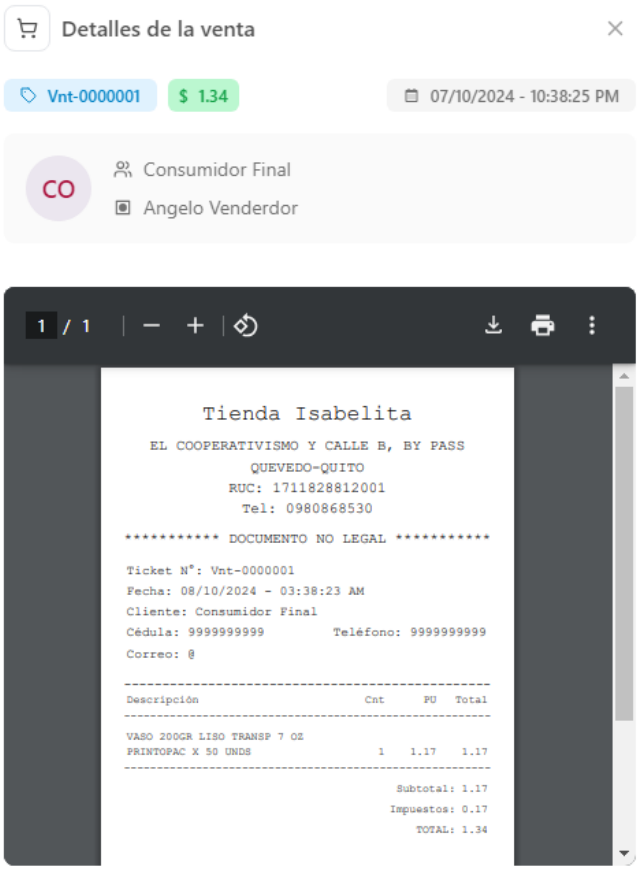


1. **Widgets de Información:** Visualización rápida de datos clave como la total de clientes y total monetario de ventas y facturas pendientes o facturadas.
2. **Métrica:** Proporciona una visión general de las ventas por en las últimas semanas, meses o años, facilitando el análisis de comportamiento y tendencias.

3. **Tabla de productos:** Este módulo presenta una tabla que contiene información esencial sobre los servicios disponibles en el inventario.



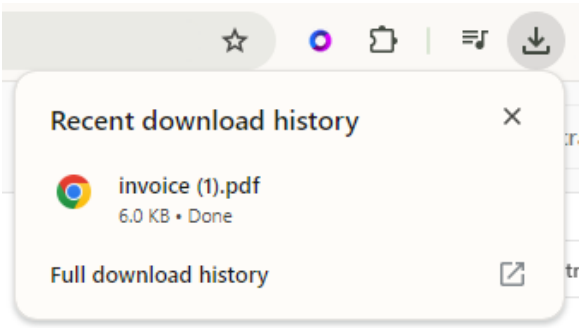
- **Encabezado**: Esta sección permite organizar la información de la tabla de ventas. Al pasar el cursor sobre los títulos de las columnas, los usuarios pueden ordenar los datos de acuerdo con sus necesidades, facilitando la búsqueda y el análisis de la información de los servicios.
- **Acciones:** En esta sección, los usuarios tienen acceso a dos botones que permiten realizar acciones sobre las ventas:
  - **Ver detalles:** Permite ve los detalles sobre la venta.



- **Cambia estado de facturación:** Esta acción cambia el estado de facturación.



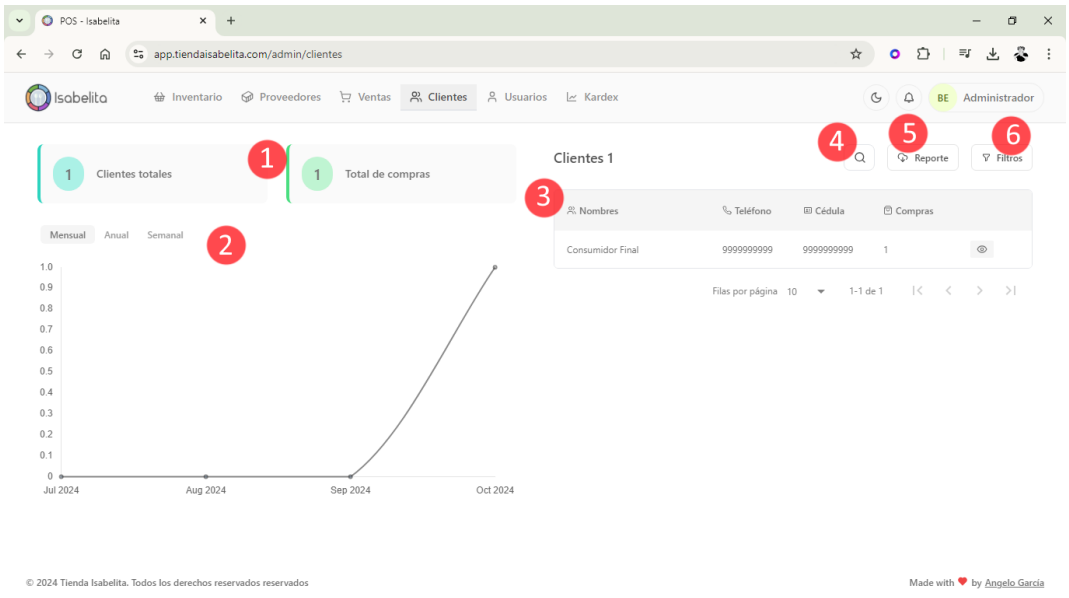
- **Descargar:** Descargar el recibo de la venta en formato PDF.



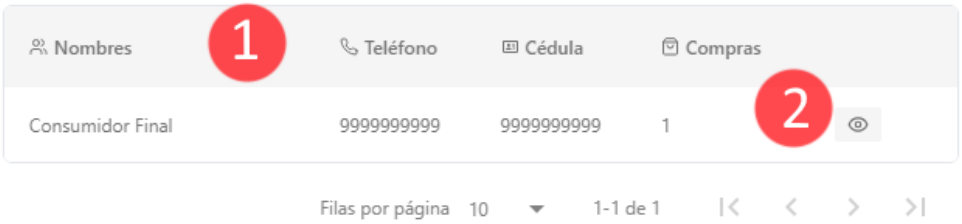
4. **Buscar:** Permite buscar venta por campos de ticket y cliente.
5. **Reportes:** Permite realizar reportes, con filtros como seleccionando las columnas, fechas.
6. **Filtros avanzados:** Permite hacer un filtrado avanzado por fechas, tipo cliente y estado de facturas.

Clientes: Información y gestión de clientes

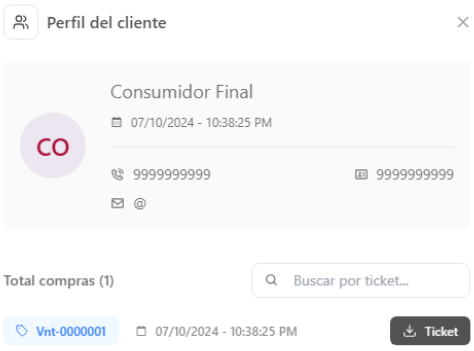
En el módulo de clientes se gestionan todos los datos relacionados con los clientes de la empresa.



1. **Widgets de Información:** Visualización rápida de datos clave como la total de clientes y compras.
2. **Métrica:** Proporciona una visión general de los clientes por en las últimas semanas, meses o años, facilitando el análisis de comportamiento y tendencias.
3. **Tabla de clientes:** Este módulo presenta una tabla que contiene información esencial sobre los clientes.



    1. **Encabezado**: Esta sección permite organizar la información de la tabla de ventas. Al pasar el cursor sobre los títulos de las columnas, los usuarios pueden ordenar los datos de acuerdo con sus necesidades, facilitando la búsqueda y el análisis de la información de los clientes.
    2. **Acciones:** En esta sección, los usuarios tienen acceso a un botón que permite realizar esta acción sobre los clientes:
        - **Ver detalles:** Permite ve los detalles sobre los clientes.

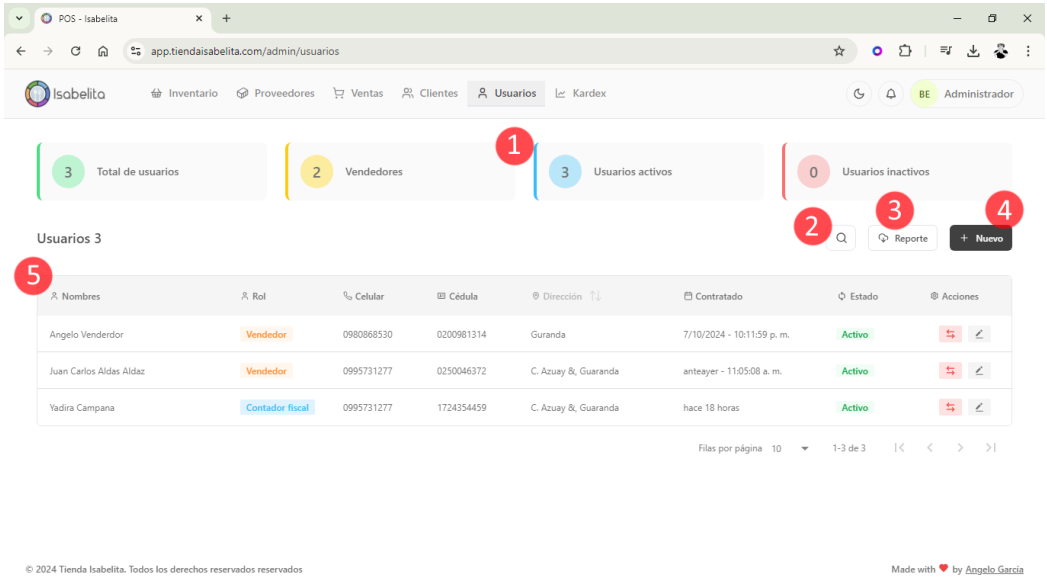


4. **Buscar:** Permite buscar una venta por campos de nombre, teléfono y cliente.
5. **Reportes:** Permite realizar reportes, con filtros como seleccionando las columnas, fechas.

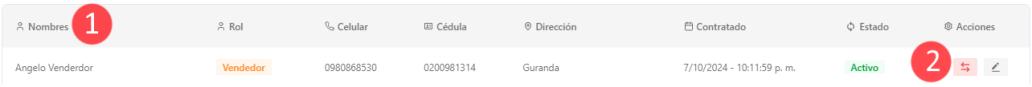
6. **Filtrar:** Permite hacer un filtrado avanzado por fechas y tipo cliente.

Usuarios: Gestión de usuarios del sistema

Este módulo permite la administración de los usuarios que tienen acceso al sistema de gestión de usuarios.



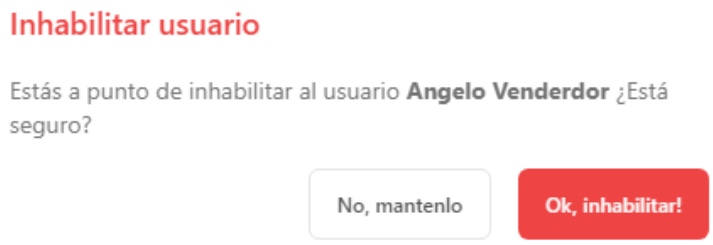
1. **Widgets de Información:** Visualización rápida de datos clave como la total de usuarios, vendedores, usuarios activos e inactivos.
2. **Buscar:** Permite buscar venta por campos de nombres, celular, cédula y dirección.
3. **Reportes:** Permite realizar reportes, con filtros como seleccionando las columnas, fechas.
4. **Tabla de usuarios:** Este módulo presenta una tabla que contiene información esencial sobre los usuarios.



1. **Encabezado**: Esta sección permite organizar la información de la tabla de ventas. Al pasar el cursor sobre los títulos de las columnas, los usuarios pueden ordenar los datos de acuerdo con sus necesidades, facilitando la búsqueda y el análisis de la información de los usuarios.

2. **Acciones:** En esta sección, los usuarios tienen acceso a tres botones que permiten realizar acciones sobre los productos:

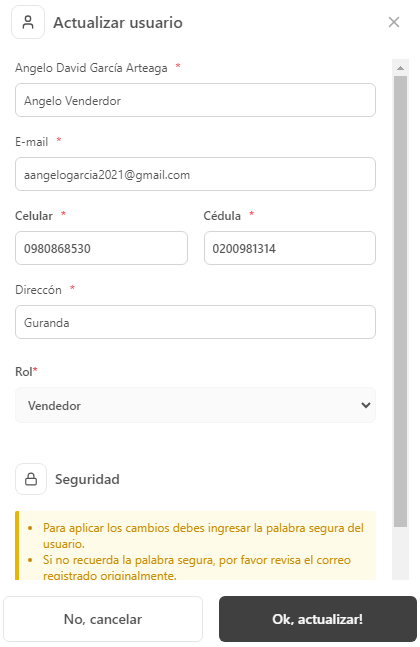
   -  **Dar de baja:** Permite dar de baja un usuario del sistema, marcándolo como inactivo.



   -  **Reactivar:** Opción para restablecer un usuario que ha sido dado de baja, devolviéndolo a un estado activo.

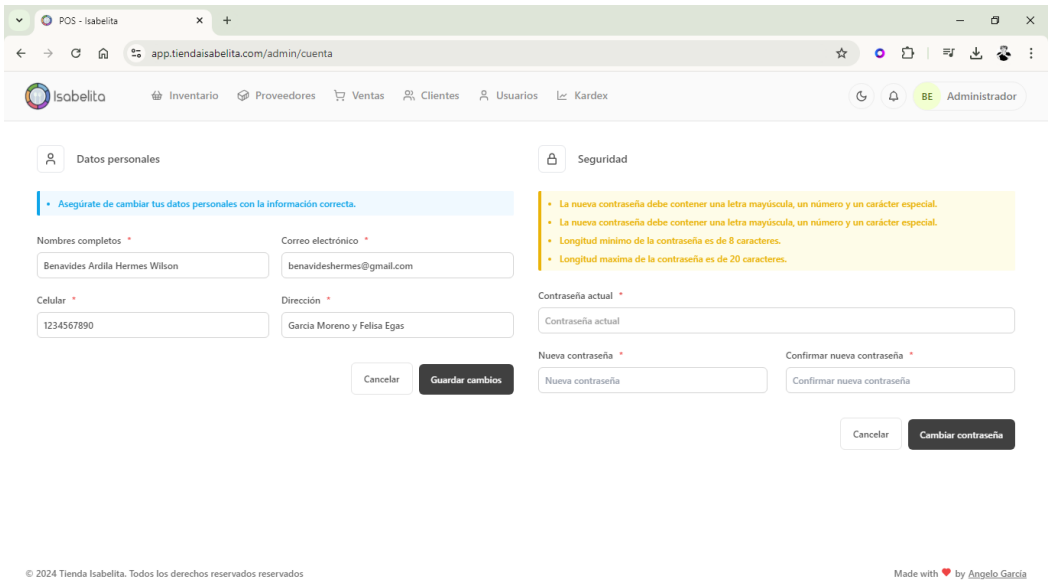


   -  **Editar:** Permite modificar la información del usuario seleccionado, asegurando que los datos estén actualizados y sean precisos.


Actualizar usuario
Angelo David García Arteaga *
Angelo Venderdor
E-mail *
aangelogarcia2021@gmail.com
Celular *
0980868530
Cédula *
0200981314
Direccón *
Guranda
Rol*
Vendedor
Seguridad
Para aplicar los cambios debes ingresar la palabra segura del usuario.
Si no recuerda la palabra segura, por favor revisa el correo registrado originalmente.
No, cancelar
Ok, actualizar!

4.7. Configuraciones de la cuenta

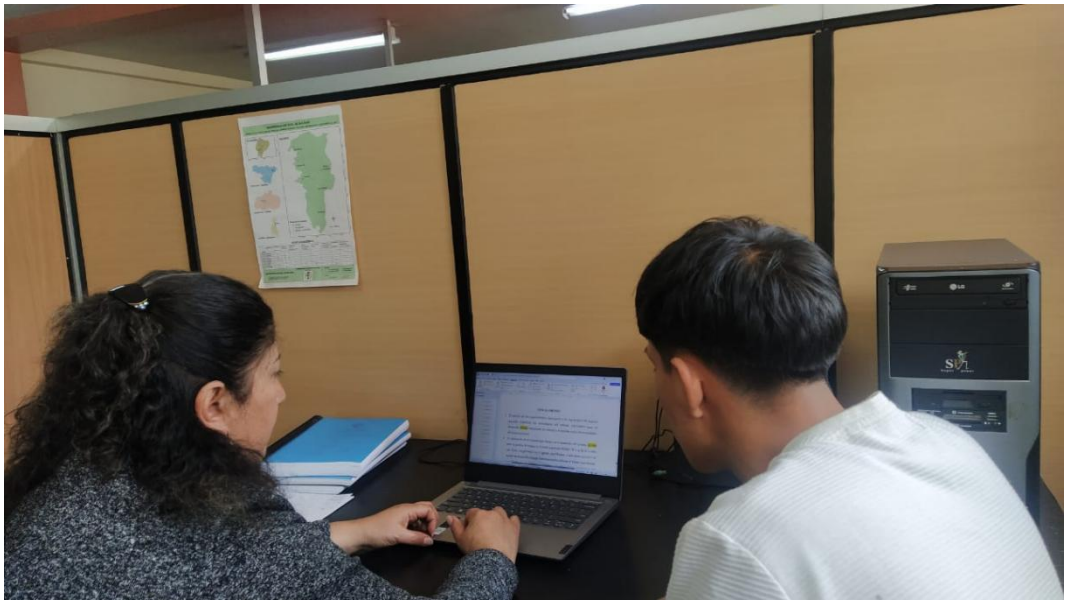
Esta sección permite a los usuarios gestionar y personalizar la configuración de su cuenta en el sistema. A través de este módulo, se pueden realizar diversas acciones, como:



- **Actualizar información personal:** Modificar datos como nombre, correo electrónico y número de contacto
- **Cambiar contraseña:** Permite a los usuarios establecer una nueva contraseña para garantizar la seguridad de su cuenta.

ANEXO 9

Fotografías





ANEXO 10

Certificado Antiplagio

ING. MÓNICA BONILLA MANOBANDA EN CALIDAD DE DIRECTORA DEL TRABAJO DE INTEGRACIÓN CURRICULAR,

CERTIFICA

Que el trabajo de integración curricular denominado "**Sistema On line para la gestión de ventas del local comercial Isabelita en la ciudad de Guaranda**", presentado por Juan Carlos Aldas Aldaz y Angelo David García Arteaga estudiantes de la **carrera de Software** pasó el análisis de coincidencia no accidental en la herramienta URKUND, reflejando un **porcentaje de similitud del 9%**, como se puede evidenciar en el documento adjunto.

Guaranda, 06 de diciembre del 2024

Atentamente,



Ing. Mónica Bonilla
Directora

ANEXO 11

Certificado Turnitin

Juan Carlos Aldas Aldaz

SISTEMA ON-LINE PARA LA GESTIÓN DE VENTAS DEL LOCAL - V2.docx

TITULACION UEB
My Files
Universidad Estatal de Bolivar

Detalles del documento

Identificador de la entrega
trn:oid:::3117:412958161

Fecha de entrega
5 dic 2024, 12:21 p.m. GMT-5

Fecha de descarga
7 dic 2024, 7:38 p.m. GMT-5

Nombre de archivo
SISTEMA ON-LINE PARA LA GESTIÓN DE VENTAS DEL LOCAL - V2.docx

Tamaño de archivo
16.5 MB

280 Páginas

35,681 Palabras

201,210 Caracteres

Escaneado con CamScanner

9% Similitud general

El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para ca...

Filtrado desde el informe

- Bibliografía
- Texto citado
- Texto mencionado
- Coincidencias menores (menos de 8 palabras)
- Trabajos entregados
- Base de datos de Crossref
- Base de datos de contenido publicado de Crossref

Exclusiones

- N.º de coincidencias excluidas

Fuentes principales

9% Fuentes de Internet
1% Publicaciones
0% Trabajos entregados (trabajos del estudiante)

Marcas de integridad

N.º de alerta de integridad para revisión

Texto oculto
45 caracteres sospechosos en N.º de páginas
El texto es alterado para mezclarse con el fondo blanco del documento.

Los algoritmos de nuestro sistema analizan un documento en profundidad para buscar inconsistencias que permitirían distinguirlo de una entrega normal. Si advertimos algo extraño, lo marcamos como una alerta para que pueda revisarlo.

Una marca de alerta no es necesariamente un indicador de problemas. Sin embargo, recomendamos que preste atención y la revise.

Escaneado con CamScanner