



PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

Unidad Académica de Formación Técnica y Tecnológica – PUCE TEC

**DESARROLLO DE APLICACIÓN WEB ADMINISTRATIVA PARA EL
REGISTRO DE VENTAS, INGRESOS E
INVENTARIO**

**Proyecto de titulación previo a la obtención del título de: Tecnóloga Superior En
Desarrollo De Software**

Autores:

Daniela Meneses Camacho

Joan Andreu Mera Ramirez

Tutor: Andrés Mauricio Chilingua Tutachá

Quito, Ecuador

2024

Índice

Índice de tablas	6
Índice de figuras.....	8
Introducción	13
Antecedentes	15
Problema técnico.....	15
Objetivos	17
Objetivo General:	17
Objetivos Específicos:.....	17
Marco Conceptual.....	17
ReactJS:.....	17
JavaScript:	18
TypeScript:	18
Vite:.....	18
Scrum	19
Capítulo I Marco Teórico.....	21
Procesos Actuales.....	21
Metodología Scrum	26
Actores Principales.....	26
Product Owner (Propietario del Producto)	26
Scrum Master (Facilitador)	26

Equipo de Desarrollo	27
Etapas del Proceso.....	27
Planificación del Sprint.....	27
El Sprint	27
Reuniones Diarias (Daily Scrum)	27
Revisión del Sprint (Sprint Review)	27
Retrospectiva del Sprint (Sprint Retrospective)	28
Capítulo II Construcción.....	33
Análisis y Diseño	33
Mapa de Navegación.....	36
Mapa de navegación del sistema de elaboración propia.....	36
Viabilidad Técnica	37
Viabilidad Económica.....	39
Casos de uso.....	40
Caso de Uso: clientes	40
Caso de Uso: Gestión de Proveedores	42
Caso de Uso: Gestión de Productos y Servicios	44
Caso de Uso: Gestión de Pedidos	46
Caso de Uso: Gestión de Ventas	48
Capítulo III Desarrollo.....	50
Requerimientos del sistema.....	50

Requerimientos funcionales:	50
Modulo Clientes.....	50
Modulo Proveedores	50
Modulo Ventas.....	50
Modulo Pedidos	50
Modulo Reportes.....	50
Modulo Caja.....	51
Modulo Usuarios.....	51
Requerimientos no funcionales:	51
Cronograma.....	51
Construcción.....	55
Diagrama de clases.....	55
Diagrama de Secuencia	56
Ingreso al sistema:.....	56
Ingreso de Nuevo Producto:.....	57
Ingreso de Nueva Venta:.....	58
Modelo BDD	63
Diccionario de datos.....	65
Arquitectura.....	73
Capítulo IV Pruebas y Resultados	74
Pruebas y Resultados.....	74

Conclusiones	82
Recomendaciones	83
Referencias bibliográficas.....	84

Índice de tablas

Tabla 1	22
Tabla 2	24
Tabla 3	28
Tabla 4	28
Tabla 5	29
Tabla 6	29
Tabla 7	30
Tabla 8	31
Tabla 9	31
Tabla 10	37
Tabla 11	39
Tabla 12	40
Tabla 13	41
Tabla 14	41
Tabla 15	42
Tabla 16	43
Tabla 17	43
Tabla 18	44
Tabla 19	45
Tabla 20	45
Tabla 21	46
Tabla 22	47
Tabla 23	47
Tabla 24	48

Tabla 25	49
Tabla 26	52
Tabla 27	53
Tabla 28	65
Tabla 29	65
Tabla 30	66
Tabla 31	67
Tabla 32	67
Tabla 33	67
Tabla 34	68
Tabla 35	69
Tabla 36	69
Tabla 37	70
Tabla 38	71
Tabla 39	71
Tabla 40	72
Tabla 41	72
Tabla 42	74
Tabla 43	75
Tabla 44	76
Tabla 45	77
Tabla 46	78
Tabla 47	79
Tabla 48	80
Tabla 49	81

Índice de figuras

Ilustración 1	36
Ilustración 2	36
Ilustración 3	40
Ilustración 4	41
Ilustración 5	42
Ilustración 6	44
Ilustración 7	46
Ilustración 8	48
Ilustración 9	55
Ilustración 10	55
Ilustración 11	56
Ilustración 12	57
Ilustración 13	58
Ilustración 14	59
Ilustración 15	60
Ilustración 16	61
Ilustración 17	63

DECLARACIÓN y AUTORIZACIÓN

Yo, **Joan Andreu Mera Ramirez** con C.I. 1752599652 autor del trabajo de Titulación intitulado: **“Desarrollo de Aplicación Web Administrativa para el Registro de ventas, Ingresos e Inventario”**, previa a la obtención del título de **Tecnólogo Superior en Desarrollo de Software** en la Unidad Académica de Formación Técnica y Tecnológica PUCE TEC:

1.- Declaro tener pleno conocimiento de la obligación que tiene la Pontificia Universidad Católica del Ecuador, de conformidad con el artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de graduación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la Pontificia Universidad Católica del Ecuador a difundir a través de sitio web de la Biblioteca de la PUCE el referido trabajo de titulación, respetando las políticas de propiedad intelectual de Universidad.

Quito, 15 de febrero 2025



Joan Andreu Mera Ramirez

C.I. 1752599652

Yo, **Daniela Meneses Camacho** con C.I. AV626509 autor(a) del trabajo de Titulación intitulado: **“Desarrollo de Aplicación Web Administrativa para el Registro de ventas, Ingresos e Inventario”**, previa a la obtención del título de **Tecnóloga Superior en Desarrollo de Software** en la Unidad Académica de Formación Técnica y Tecnológica PUCE TEC:

1.- Declaro tener pleno conocimiento de la obligación que tiene la Pontificia Universidad Católica del Ecuador, de conformidad con el artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de graduación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la Pontificia Universidad Católica del Ecuador a difundir a través de sitio web de la Biblioteca de la PUCE el referido trabajo de titulación, respetando las políticas de propiedad intelectual de Universidad.

Quito, 15 de febrero 2025



Daniela Meneses Camacho

C.I. AV626509

Agradecimientos Joan Mera

A mis padres, principal apoyo en el transcurso de mi carrera y desarrollo personal, a mis compañeros y la gente que me ha construido en el camino, por brindarme el apoyo y las experiencias necesarias, a cada uno de los retos y circunstancias que se presentaron a lo largo del camino que permitieron mi crecimiento. A todos y cada uno de las impresionantes personas que conocí, gracias.

Agradecimientos Daniela Meneses

A mi familia, que ha sido el pilar inquebrantable en cada paso de este camino. Su amor, apoyo incondicional y palabras de aliento han iluminado incluso los momentos más oscuros. sin ellos esto no hubiese sido posible. Gracias por ser mi soporte cuando no podía.

A mis compañeros, gracias por empujarme a crecer, por demostrarme que rodearse de personas excepcionales eleva nuestras propias capacidades. Su espíritu de superación fue clave para encontrar el mío.

Introducción

El presente trabajo consiste en la elaboración de una aplicación web administrativa que tiene como objetivo mejorar el control de los ingresos, el inventario y las ventas de productos y servicios de SOLTICEM TECHNOLOGYS. Este sistema permitirá a la empresa disponer de información actualizada y precisa sobre sus productos, lo cual es fundamental para tomar decisiones informadas y oportunas.

Entre los beneficios del sistema está la capacidad de mantener un inventario automático. Este inventario se actualizará en tiempo real, nos va a permitir reducir la necesidad de realizar conteos manuales y reduciendo el riesgo de errores humanos. La disponibilidad de datos actualizados ayudará a la empresa a gestionar mejor sus niveles de stock y evitar rupturas de inventario. El proyecto no contara con un sistema de facturación electrónica, las ventas serán registradas a través de un formulario, el cual tendrá relación directa con el inventario. Los módulos contemplados son:

Clientes:

- Formulario de ingreso de clientes: Permite añadir nuevos clientes al sistema.
- Vista de clientes: Muestra una lista de los clientes ya ingresados en el sistema.

Proveedores:

- Formulario de ingreso de proveedor: Permite añadir nuevos proveedores al sistema.
- Vista de proveedores: Muestra una lista de los proveedores ya ingresados en el sistema.

Inventario:

- Administración de bodega: Módulo para gestionar y visualizar los productos en inventario.
- Información por producto: Marcas, Producto, Servicios, Bodega-Sucursales, Ingresos/Egresos de mercadería, Reportes.
- Filtros disponibles: Productos más vendidos, productos menos vendidos, productos sin stock. Estos serán los únicos procesos de análisis que el sistema ofrecerá para el inventario.

Productos y Servicios:

- Formulario de ingreso: Permite añadir nuevos productos o servicios al sistema.
- Información a ingresar: Marcas, Producto o Servicio, Bodega-Sucursales, Precio de venta.
- Vista de productos y servicios: Muestra una lista de los productos y servicios ya ingresados.

Venta:

- Formulario de ingreso de venta: Permite registrar nuevas ventas.
- Vista de ventas: Muestra una lista de las ventas ya registradas.

Pedidos:

- Formulario de ingreso de pedido: Permite registrar nuevos pedidos.
- Vista de pedidos: Muestra una lista de los pedidos ya registrados.

Caja:

- Saldo en efectivo: Refleja el saldo de dinero en efectivo disponible en la empresa en un momento dado.

Seguridad:

- Administración de usuarios y perfiles: Permite gestionar los usuarios y sus perfiles en el sistema.

La implementación de este sistema administrativo surge de la necesidad de contar con procesos automáticos que mejoren la eficiencia y efectividad de la administración comercial. En un entorno de negocios cada vez más digital y competitivo, disponer de herramientas que faciliten un mejor control operativo en el día a día es esencial.

Para su desarrollo, se emplearán técnicas modernas de gestión de datos, automatización de procesos, conocimientos en JavaScript, bases de datos y análisis de la información, asegurando que la empresa pueda adaptarse rápidamente a las cambiantes demandas del mercado, enriquecer la satisfacción del cliente e incrementar su rentabilidad y crecimiento

Antecedentes

La empresa gestionaba su inventario de manera manual, utilizando hojas de cálculo y registros físicos para controlar la entrada y salida de productos. Este método era propenso a errores humanos, dificultaba el acceso a información en tiempo real y generaba problemas como rupturas de stock y acumulación de mercancías.

La falta de automatización también complicaba la planificación de compras y ventas, afectando el flujo de operaciones y el tiempo de respuesta al cliente. Debido a la constante necesidad del mercado de recibir una respuesta rápida y precisa, la empresa identificó la urgencia de implementar un sistema automatizado de gestión de inventarios para optimizar sus procesos y mantenerse competitiva.

Problema técnico

El problema técnico identificado en la empresa es la falta de un control actualizado y preciso del inventario. Esta deficiencia ha generado varios inconvenientes, tales como la incapacidad de conocer en tiempo real la disponibilidad de productos, el riesgo de rupturas de

stock y la acumulación innecesaria de mercancías. La falta de un sistema de inventario automatizado también impide una correcta planificación de las compras y ventas, causando una afectación directa hacia la experiencia del cliente.

Tomando en cuenta el contexto socioeconómico, en donde cada vez más y más competidores ofrecen soluciones tecnológicas, es crucial para las empresas mantener una gestión eficiente de sus recursos. La empresa opera en un entorno donde la rapidez y la precisión en la entrega de productos son factores determinantes para el éxito. Además, la integración de soluciones tecnológicas avanzadas se ha convertido en una necesidad para mantenerse competitivo.

La adopción de esta tecnología no solo optimiza los procesos internos, sino que también mejora la capacidad de respuesta ante las cambiantes necesidades del sector y las expectativas de los compradores, contribuyendo de esta manera al crecimiento y la sostenibilidad de la empresa en un entorno digitalizado.

Objetivos

Objetivo General:

Desarrollar una aplicativo web para SOLTICEM TECHNOLOGYS que permita a los empleados gestionar de manera eficiente las operaciones relacionadas con las ventas, ingresos e inventario en tiempo real.

Objetivos Específicos:

- Crear una estructura de fuente de datos que gestione eficientemente usuarios, productos y registros de ventas, asegurando la integridad y disponibilidad de la información clave para SOLTICEM.
- Implementar interfaces intuitivas para la gestión de usuarios, productos, ventas, pedidos, caja, y proveedores facilitando la interacción y actualización en tiempo real de datos críticos para el negocio.
- Desarrollar lógica de negocio en el back-end para gestionar operaciones complejas como actualizaciones de inventario, generación de reportes y registro de ventas, asegurando una comunicación eficiente y segura a través de la fuente de datos y la interfaz gráfica.

Marco Conceptual

Para la construcción de la interfaz por la que los usuarios utilizarán el programa se utilizarán tecnologías modernas como React, JavaScript, TypeScript y Vite para la creación de interfaces dinámicas y responsivas. Estas herramientas permitirán crear una experiencia de usuario fluida y atractiva, optimizando la usabilidad y la accesibilidad de la aplicación web.

ReactJS:

ReactJS, una biblioteca de JavaScript altamente usada generalmente para desarrollar interfaces UI intuitivas. Basado en componentes que se pueden llegar a utilizar varias veces,

facilita el desarrollo modular y escalable de aplicaciones web modernas. Su eficiencia en la actualización del DOM (Document Object Model) y la gestión del estado lo convierten en una excelente alternativa cuando se trata de tener una interfaz ágil. "Los empresarios prefieren el desarrollo de ReactJS ya que ayuda a aumentar las tasas de conversión, proporciona una opción de backlink checker, reduce los costes de desarrollo y aumenta el reconocimiento de la marca. Incluso Meta y una amplia comunidad de desarrolladores individuales y corporaciones apoyan ReactJS, que es un marco de trabajo gratuito y de código abierto. (Kundariya, 2024)

JavaScript:

Lenguaje de programación base para React, permite la creación de aplicaciones dinámicas e interactivas. Su flexibilidad y compatibilidad con múltiples entornos lo convierten en la opción más utilizada para el desarrollo web moderno. " se trata de una de las principales tecnologías de la World Wide Web". (Amazon aws, 2025)

TypeScript:

TypeScript es un lenguaje basado en JavaScript que introduce tipado estático, lo que ayuda a prevenir errores y mejora la mantenibilidad del código. Su adopción en proyectos React ha ido en aumento debido a los beneficios en la detección temprana de errores y en el autocompletado en los entornos de desarrollo. " Validar las conexiones entre cada parte de un proyecto puede consumir mucho tiempo rápidamente. Usar un lenguaje con verificación de tipos como TypeScript puede manejar eso automáticamente y brindar retroalimentación instantánea durante el desarrollo." (Microsoft, 2025)

Vite:

es un entorno de construcción moderno que optimiza la experiencia de desarrollo en React al proporcionar una ejecución más rápida gracias a su servidor de desarrollo basado en

ES Modules. Comparado con otros bundlers tradicionales, Vite ofrece una recarga en caliente instantánea y una construcción optimizada para producción. " Vite es una herramienta de compilación de frontend extremadamente rápida, que impulsa la próxima generación de aplicaciones web." (Vite, 2025)

Scrum

La metodología escogida para desarrollar el proyecto es Scrum, esta metodología ágil se seleccionó específicamente por su capacidad probada para gestionar proyectos complejos de manera eficiente y adaptativa en el campo técnico y tecnológico. Scrum no solo facilita la entrega continua de valor, sino también promueve la colaboración estrecha entre los miembros del equipo, lo cual es crucial para resolver problemas de manera efectiva

La estructura de trabajo en iteraciones cortas, conocidas como sprints, permite que el equipo desarrolle, pruebe y mejore continuamente las soluciones. Esta metodología promueve la transparencia y la comunicación frecuente a través de reuniones diarias cortas (Daily Standups), donde cada miembro del equipo informa sobre el progreso y los posibles obstáculos encontrados.

Para la recolección de información y la investigación, se emplearán técnicas adaptadas al ciclo de vida de Scrum. Inicialmente, se definirá un Backlog de Producto que contendrá los requisitos y funcionalidades prioritarias. Este servirá como guía durante todo el proyecto, siendo actualizado y ajustado según las necesidades emergentes y la retroalimentación del cliente.

Durante cada sprint, se realizarán demostraciones del producto funcional al cliente para obtener comentarios tempranos y validación de las soluciones desarrolladas. Además, al final de cada sprint, se llevarán a cabo retrospectivas para identificar áreas de mejora en el proceso y ajustar la próxima planificación.

La elección de Scrum como metodología para este proyecto asegura una gestión eficiente y adaptable, garantizando la entrega de valor continuo y la satisfacción del cliente en el ámbito técnico y tecnológico.

Capítulo I Marco Teórico

Procesos Actuales

A través de reuniones con el propietario del negocio se pudieron identificar los procesos actuales que corresponden al proceso de venta del negocio:

Proceso de venta

1. El cliente se comunica mediante de las redes sociales de la empresa, página web o se acerca a la sucursal para solicitar el producto o servicio que requiere.
2. El cliente es atendido por el asesor de ventas quien identifica el servicio o producto que el cliente necesita.
3. Si es un producto se le muestra las distintas opciones que existen y cual se ajusta más a su presupuesto y objetivo, si es un servicio se le pide una descripción del caso e información personal además se le provee de indicaciones, contacto y dirección de la sucursal en caso de comunicarse en línea o si el cliente ya se encuentra en la sucursal se recibe el equipo a tratar.
4. Si el cliente confirma el producto que desea llevarse se completa la venta y se registra el producto que se vendió en un Excel, en caso del servicio se llena una hoja que contiene la descripción del problema junto con el contacto del cliente y se envía una notificación avisando que hay un equipo a ser revisado con una breve descripción del servicio solicitado.
5. Dado el caso en que el producto solicitado no se encuentre disponible se consulta si es que desea traerlo bajo pedido, si la respuesta es positiva se anotan los datos del cliente y se le indica que habrá una notificación cuando su producto este en la sucursal, si es un servicio se realiza un diagnóstico para confirmar el origen del problema junto con la descripción del caso.

6. Para el producto fuera de disponibilidad una vez confirmado se coordina con el encargado de compras para adquirirlo con el proveedor y se le informa una fecha estimada al cliente, en el servicio se envían los resultados del diagnóstico junto con el precio a pagar y se espera por una confirmación para aplicar la solución, caso contrario se devuelve el equipo y se cobra el diagnóstico.

A continuación, en esta tabla se describen los procesos principales que actualmente se llevan a cabo en SOLCITEM, información que fue proporcionada directamente por los propietarios durante entrevistas. Detalla cómo opera cada área de la empresa, identificando responsables, métodos actuales de trabajo y los problemas asociados.

Tabla 1

Procesos Actuales

Área	Responsable	No. de Personas en Atención	Descripción del Proceso	Problema Identificado
Ventas	Asesor de Ventas	1	Registra las ventas en hojas físicas, actualiza manualmente los ingresos en hojas de Excel y coordina la entrega con el cliente.	La actualización manual genera demoras, errores de registro y dificultades para acceder a datos históricos confiables.

Compras	Encargado de Compras	1	Realiza pedidos a proveedores mediante correo electrónico, mantiene un control en hojas de Excel y verifica manualmente el inventario para determinar las necesidades.	La falta de integración en el control de inventario provoca duplicidad de pedidos o desabastecimientos.
Inventario	Encargado de Compras	1	Actualiza entradas y salidas de productos en un registro físico y después transfiere la información a una hoja de cálculo.	El proceso manual dificulta la trazabilidad, incrementa errores y no permite conocer el inventario en tiempo real.

Facturación	Contador	1	Emite facturas en formato físico a partir de registros en Excel y realiza cálculos de impuestos manualmente.	El proceso manual es propenso a errores y genera demoras, afectando la emisión eficiente de facturas y los cálculos de impuestos.
--------------------	----------	---	--	---

Nota: En esta tabla se describe al personal involucrado y sus roles principales junto con el problema que acarrea.

Esta tabla presenta un panorama detallado de los recursos tecnológicos que SOLTICEM tiene en sus distintas áreas, según información obtenida directamente de los propietarios. Incluye características específicas de los dispositivos en uso, así como su ubicación dentro de la empresa.

Tabla 2

Infraestructura Actual de la empresa

Área	Cantidad	Dispositivo	Características
Área de Atención al cliente	9	PC de escritorio	Procesador Core I3
			Ram 8GB
			Disco Duro 1TB
			Arquitectura 32 bits
Oficina principal	4	Impresora Epson L3210 (con escáner)	Impresora multifuncional (impresión + escáner)

			con conexión Wi-Fi, impresión en color
	1	Router inalámbrico WiFi TP-Link	3 antenas, 450Mbps, velocidad inalámbrica ideado para aplicaciones sensibles como video HD, encriptado fácil, control de ancho de banda basado en IP
	2	UPS PC UPS CDP R-UPR508	Interactivo con regulador, 500VA, 240W, 8 tomas
	5	Laptop 240 G7 14" (color plateado ceniza oscuro)	4GB de RAM
			500GB HDD
			Intel Celeron
	1	Servicio de internet Celerity	Tipo: Inalámbrico de alta velocidad con 450Mbps, adecuado para aplicaciones de video HD y uso en múltiples dispositivos.

Nota: Infraestructura con la que cuenta el negocio.

Metodología Scrum

Scrum es una metodología ágil muy útil a la hora de gestionar un proyecto, facilita a los equipos planificar y coordinar su trabajo. Este modelo fomenta la colaboración estrecha, el aprendizaje continuo y la adaptación a los cambios de forma ágil. Si bien es asociado con frecuencia al área de tecnología, sus principios y rituales pueden extenderse a muchos otros ámbitos y entornos que requiera un enfoque iterativo y orientado a la mejora constante.

(Drumond, 2025)

La herramienta está diseñada para gestionar proyectos de manera colaborativa y eficiente. Se enfoca en alcanzar objetivos específicos a través de iteraciones dentro de un período definido de tiempo, adaptándose rápidamente a los cambios y fomentando la mejora continua del producto. (Martins, 2024)

Actores Principales

Product Owner (Propietario del Producto)

Es el encargado de plasmar los intereses del cliente o usuario en el proyecto. Su función principal es garantizar que el equipo trabaje en lo que realmente aporta valor junto con decidir cuando un avance está realmente listo para ser entregado. Organiza y prioriza las tareas del equipo en una lista que se llama Product Backlog. (Martins, 2024)

Scrum Master (Facilitador)

Este rol se centra en guiar al equipo y asegurar que se sigan los eventos de Scrum. También elimina cualquier obstáculo que pueda interferir con el progreso de los desarrolladores y fomenta un entorno de trabajo eficiente. (Martins, 2024)

Equipo de Desarrollo

Es el personal implicado en la resolución de requerimientos técnicos para transformar las ideas en productos funcionales. Está compuesto por personas con habilidades diversas que colaboran estrechamente para alcanzar los objetivos del Sprint. (Martins, 2024)

Etapas del Proceso

Planificación del Sprint

En esta fase el equipo separa las tareas del Product Backlog que se completarán durante el Sprint, un período de trabajo definido (generalmente de 1 a 4 semanas) y un objetivo común a lograr. (Martins, 2024)

El Sprint

Es el núcleo del proceso Scrum. Durante este tiempo, el equipo se dedica exclusivamente a las tareas seleccionadas, con el objetivo de entregar un resultado funcional al finalizar el Sprint. (Martins, 2024)

Reuniones Diarias (Daily Scrum)

Estas reuniones cortas (con un máximo de 15 minutos) permiten que el equipo comparta avances, planes y obstáculos, promoviendo la transparencia y la coordinación. (Martins, 2024)

Revisión del Sprint (Sprint Review)

Al concluir el Sprint, el equipo muestra los resultados obtenidos a los interesados (clientes, usuarios, etc.). Durante esta etapa, se recopila retroalimentación valiosa para futuras mejoras. (Martins, 2024)

Retrospectiva del Sprint (Sprint Retrospective)

En esta sesión, los implicados analizan y reflexionan sobre su desempeño durante el Sprint, identificando fortalezas y áreas de mejora para optimizar el proceso en los próximos ciclos. (Martins, 2024)

Definido el esquema Scrum se presentan a continuación las tablas de descripción de los Sprints para el desarrollo del proyecto

Tabla 3

Primer Sprint

Sprint	Tarea	Prioridad	Responsable	Tiempo (horas)
Sprint 1	Creación de mockups front end	Media	Joan Mera	4
Sprint 1	Creación del mapa de navegación front end	Media	Joan Mera	2
Sprint 1	Investigación AWS (costos, RDS, EC2) back end	Alta	Daniela Meneses	2
Sprint 1	Configuración del proyecto con React y TypeScript back end	Alta	Daniela Meneses	1

Nota: Diseño e investigación para el proyecto

Tabla 4

Segundo Sprint

Sprint	Tarea	Prioridad	Responsable	Tiempo (horas)
Sprint 2	Investigación y elección sobre librería para el front end (MUI)	Media	Joan Mera	2
Sprint 2	Configuración de proyecto y rama para desarrollo del front end	Media	Joan Mera	2

Sprint 2	Configuración de estructura de carpetas y dependencias back end	Alta	Daniela Meneses	2
Sprint 2	Diseño de la fuente de datos back end	Alta	Daniela Meneses	1
Sprint 2	Creación del esquema de la fuente de datos back end	Alta	Daniela Meneses	2

Nota: Creación y configuraciones.

Tabla 5

Tercer Sprint

Sprint	Tarea	Prioridad	Responsable	Tiempo (horas)
Sprint 3	Desarrollo del Login -FRONT en minúscula y completar front end	Alta	Joan Mera	5
Sprint 3	Desplegar Fuente de datos back end	Alta	Daniela Meneses	3
Sprint 3	Configuración del proyecto backend (Node.js con Express)	Alta	Daniela Meneses	3
Sprint 3	Creación de endpoints de Login back end	Alta	Daniela Meneses	3
Sprint 3	Integración con fuente de datos para Login back end	Alta	Daniela Meneses	7

Nota: Desarrollo de login y fuente de datos.

Tabla 6

Cuarto Sprint

Sprint	Tarea	Prioridad	Responsable	Tiempo (horas)
Sprint 4	Desarrollo del módulo proveedores front end	Alta	Joan Mera	4
Sprint 4	Desarrollo del formulario para crear proveedores front end	Media	Joan Mera	2
Sprint 4	Desarrollo del módulo clientes front end	Alta	Joan Mera	4
Sprint 4	Desarrollo del formulario para crear clientes front end	Media	Joan Mera	2
Sprint 4	Desarrollo CRUD de clientes back end	Media	Daniela Meneses	7
Sprint 4	Desarrollo CRUD de proveedores back end	Media	Daniela Meneses	7

Nota: Generación de interfaz y API clientes

Tabla 7

Quinto Sprint

Sprint	Tarea	Prioridad	Responsable	Tiempo (horas)
Sprint 5	Desarrollo del módulo productos front end	Alta	Joan Mera	4
Sprint 5	Desarrollo del formulario para crear productos front end	Media	Joan Mera	2
Sprint 5	Desarrollo del módulo inventario front end	Alta	Joan Mera	4
Sprint 5	Desarrollo del formulario para crear inventario front end	Media	Joan Mera	2
Sprint 5	Desarrollo CRUD de productos back end	Alta	Daniela Meneses	8

Sprint 5	Desarrollo CRUD de inventario back end	Alta	Daniela Meneses	8
Sprint 5	Desarrollo de validaciones y seguridad para CRUDs back end	Alta	Daniela Meneses	3

Nota: Desarrollo de modulos y APIs

Tabla 8

Sexto Sprint

Sprint	Tarea	Prioridad	Responsable	Tiempo (horas)
Sprint 6	Desarrollo del módulo ventas front end	Alta	Joan Mera	4
Sprint 6	Desarrollo del formulario para crear la venta front end	Media	Joan Mera	3
Sprint 6	Desarrollo del módulo pedidos front end	Alta	Joan Mera	4
Sprint 6	Desarrollo del formulario para crear pedidos front end	Media	Joan Mera	3
Sprint 6	Desarrollo CRUD de ventas back end	Alta	Daniela Meneses	8
Sprint 6	Desarrollo CRUD de pedidos back end	Alta	Daniela Meneses	8

Nota: Desarrollo de módulos y APIs

Tabla 9

Séptimo Sprint

Sprint	Tarea	Prioridad	Responsable	Tiempo (horas)
Sprint 7	Desarrollo del módulo caja front end	Alta	Joan Mera	4

Sprint 7	Desarrollo del módulo seguridad front end	Alta	Joan Mera	4
Sprint 7	Desarrollo de formularios para gestión de perfiles y usuarios front end	Alta	Joan Mera	3
Sprint 7	Revisión de conexión entre módulos con router front end	Media	Joan Mera	3
Sprint 7	Revisión de flujo y comprobación de validaciones front end	Media	Joan Mera	5
Sprint 7	Gestionar seguridades de acceso y perfiles front end	Alta	Joan Mera	5
Sprint 7	Desarrollo Endpoint para el módulo Caja back end	Media	Daniela Meneses	2
Sprint 7	Pruebas finales y ajustes back end	Alta	Daniela Meneses	7

Nota: Pruebas y ajuste de validaciones

Capítulo II Construcción

Análisis y Diseño

La empresa SOLTICEM, dedicada a la venta de productos y servicios tecnológicos, enfrentaba diversos problemas derivados de sus procesos manuales actuales. La información se gestiona principalmente en hojas de Excel y documentos físicos, lo que genera demoras en la actualización de datos, errores de registro y dificultades para acceder a la información histórica de manera confiable. Además, al no contar con un sistema automatizado, el control de inventarios se vuelve impreciso, y es fácil cometer duplicidad de pedidos o desabastecimientos. Estos procesos manuales carecen de medidas de seguridad, haciendo vulnerables los datos sensibles de la empresa y aumenta el riesgo de error humano.

Para abordar estas problemáticas, se propone el desarrollo de un aplicativo web que digitalice y optimice el proceso de registro y visualización de información. Este sistema tiene como objetivo reemplazar los registros físicos con un entorno digital más confiable y dinámico, permitiendo una gestión eficiente de los datos. La solución propuesta simplificará la consulta de información, mejorará la precisión en los registros y agilizará la toma de decisiones estratégicas.

El diseño del aplicativo web se adhiere los colores corporativos de la marca, utilizando una plantilla basada en MUI (Material-UI) con una interfaz amigable e intuitiva inspirada en el estilo visual de Google. Este enfoque facilita el manejo del sistema para los usuarios, mejorando significativamente la experiencia de uso. El sistema se estructura en módulos clave, accesibles desde un menú principal de fácil navegación. Las pantallas disponibles incluyen:

- Clientes
- Proveedores

- Productos y servicios
- Ventas
- Pedidos
- Caja
- Usuarios

Cada módulo está diseñado para proporcionar acceso rápido y eficiente a la información necesaria, lo que permite su consulta y análisis. Con esta herramienta, se busca no únicamente optimizar los procesos operativos de la empresa, sino fomentar una toma de decisiones informada y ágil.

El diseño del sistema está basado en una arquitectura (MVC), haciendo posible una separación organizada entre la lógica de las funciones, la interfaz de usuario y la gestión de servicios. Esta arquitectura no solo facilita la implementación y el mantenimiento de nuevas funcionalidades, sino también hace que el sistema sea escalable a futuro, permitiendo la incorporación de nuevos módulos sin complicaciones.

En cuanto a la fuente de datos, se realizó un análisis de la información que la empresa gestionaba en sus hojas de Excel. Se identificaron las entidades clave que debían ser almacenadas de manera estructurada en la fuente de datos entre las que se encuentran: usuarios, productos, clientes, proveedores, ventas, pedidos y caja. El modelo de datos se diseñó aplicando la primera forma normal (1FN) para evitar redundancias innecesarias y asegurando la integridad de los datos. Se establecieron relaciones entre las distintas tablas. La fuente de datos se implementó mediante PostgreSQL 16, una solución robusta y eficiente que garantiza la fiabilidad y la capacidad de manejar datos a gran escala.

En cuanto al backend, se alojará localmente hasta su implementación y se utilizará Node.js con Express.js para la creación de las APIs que gestionarán el acceso hacia la

información guardada en base. Node.js es adecuado para manejar grandes volúmenes de solicitudes simultáneas, lo cual es necesario para un sistema de inventarios que debe realizar múltiples operaciones de lectura y escritura de datos con un flujo constante. Express.js proporcionará una estructura robusta y flexible para las rutas y los controladores del sistema.

Para la seguridad, el sistema utilizará JWT para la autenticación de usuarios. Este método es eficiente para manejar sesiones de usuario de forma segura, ya que el token contiene información cifrada y es imposible de modificar sin la clave adecuada. Además, para garantizar la protección de las contraseñas de los usuarios, se implementará bcrypt, un algoritmo de hashing que convierte las contraseñas en un formato irrecuperable. Esto hace que, incluso si la fuente de datos es comprometida, las contraseñas de los usuarios permanecen protegidas.

La estructura de carpetas en el proyecto será organizada en módulos claros para facilitar la escalabilidad. Las carpetas principales serán config, models, controllers y routes. En config se almacenarán las configuraciones del sistema, como la fuente de datos y la seguridad. En models se definirán los esquemas de la fuente de datos y las relaciones entre las tablas. Controllers manejará la lógica de negocio, procesando las solicitudes y respondiendo con la información correspondiente, mientras que routes mapeará las URLs a los controladores adecuados.

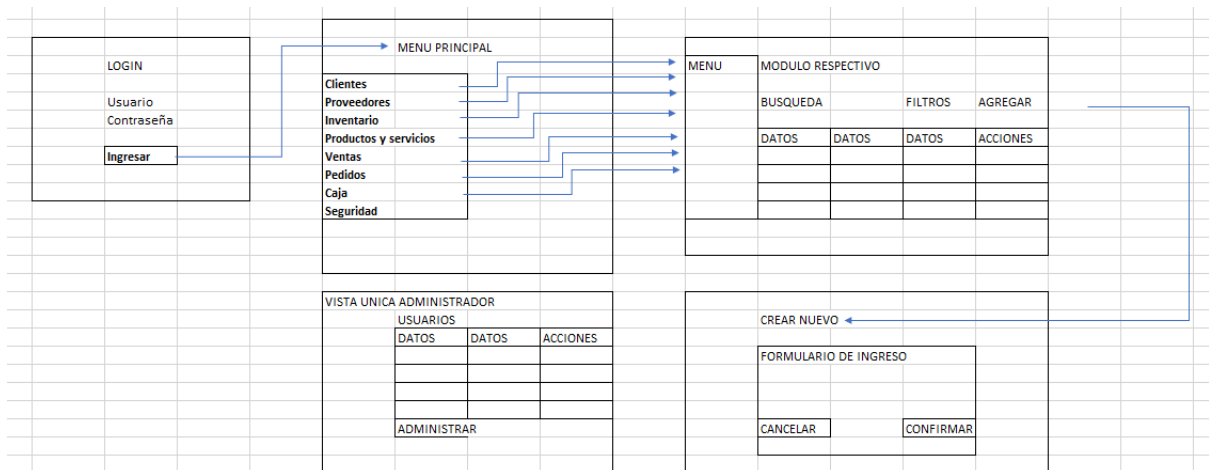
El sistema permitirá que los usuarios ingresen nuevos usuarios, clientes, ventas, productos y proveedores a través de formularios específicos. Cada acción que el personal realice dentro del sistema, como registrar una venta o agregar un producto al inventario, desencadenará una actualización totalmente automatizada de los datos, lo que asegura que la información siempre esté actualizada con un mínimo de intervención manual. Esta automatización reduce los errores humanos y optimiza el tiempo del personal, que ya no tendrá que realizar actualizaciones manuales en hojas de Excel.

Mapa de Navegación

A continuación, se presenta el mapa de navegación de la web para la presentación de los módulos e información de la empresa:

Ilustración 1

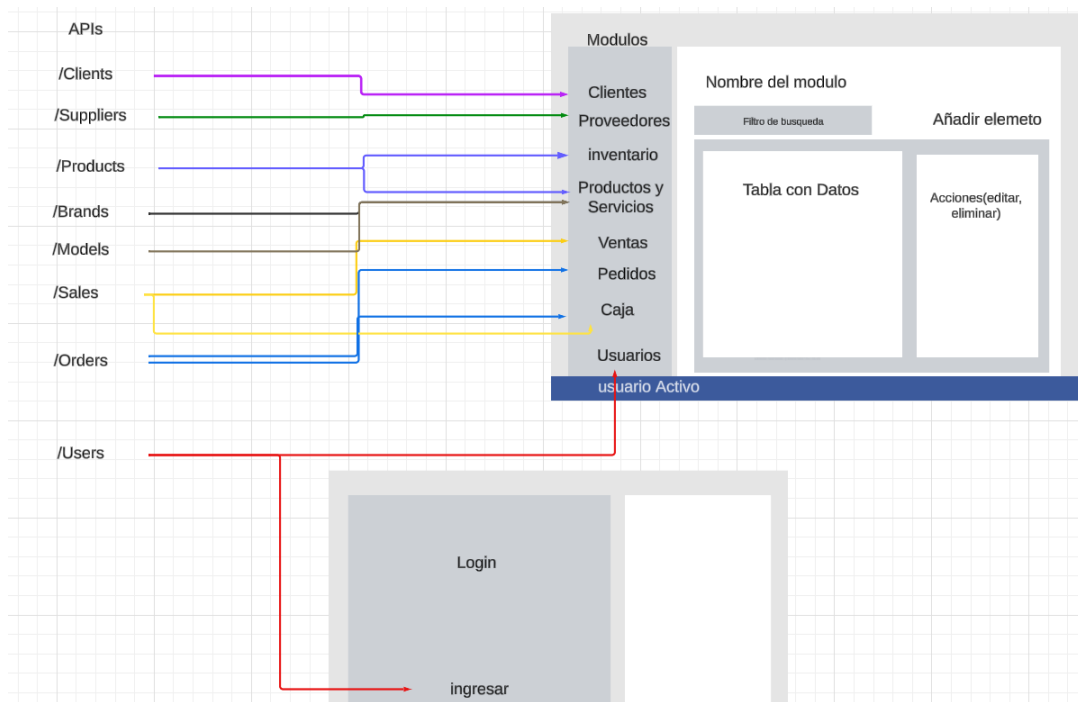
Mapa de navegación del sistema de elaboración propia



Nota: mapa de navegación e interfaz general del software.

Ilustración 2

Mapa de navegación enfocado al backend



Nota: Relación de las APIs con los módulos del proyecto.

Viabilidad Técnica

Esta tabla identifica los requisitos técnicos necesarios para alojar y operar del sistema de inventarios. Detalla aspectos como las herramientas de desarrollo requeridas, las características del servidor, y las configuraciones necesarias para la fuente de datos. También se incluye información sobre los componentes relacionados con la seguridad del sistema.

Tabla 10

Elementos necesarios para despliegue del Front

Área	Cantidad	Dispositivo	Características
Frontend	1	Computadora/Servidor	Navegador web actualizado (Chrome, Firefox, Edge).

			React.js (versión mínima 16.8 con soporte para hooks).
			Sistema operativo compatible: Windows, macOS, Linux.
Visualización de datos	1	Computadora/Servidor	Office básico
			Resolución mínima de pantalla: 1366x768.
Fuente de datos	1	Computadora/Servidor	PostgreSQL (versión 12 o superior).
			Capacidad de almacenamiento según datos proyectados.
Autenticación	1	Servidor	Uso de JWT para manejo de sesiones seguras.

Infraestructura	1	Computadora/Servidor	Capacidad mínima: 1 GB de RAM, 1 vCPU.
			Soporte para API REST y archivos estáticos.

Nota: Detalle de requisitos necesarios para una operabilidad con opciones a expansión

En base a una comparación de los recursos con los que cuenta la empresa y los elementos necesarios para el despliegue se concluye que el proyecto si es viable con unas adquisiciones como el SSL adicional, servidores en caso de no usar computadores y un hosting en nube

Viabilidad Económica

Luego de tener la constancia de que el proyecto es técnicamente viable, se realizara un análisis de viabilidad económica para determinar los costos asociados del proyecto mediante la siguiente tabla:

Tabla 11

Tabla de Viabilidad Económica

Cantidad	Dispositivo	Periodicidad	Costo
1	Servidor Local(PC o Servidor)		\$800.00
2	Desarrolladores	Mensual (4 meses)	\$600.00
		Total	\$1.400.00

Nota: Detalle de costos del proyecto a implementar

La empresa cuenta con la mayoría del equipo necesario para realizar el proyecto, el costo de los desarrolladores presentados en la tabla anterior será asumidos por los titulares del proyecto evaluando un tiempo de desarrollo de 4 meses

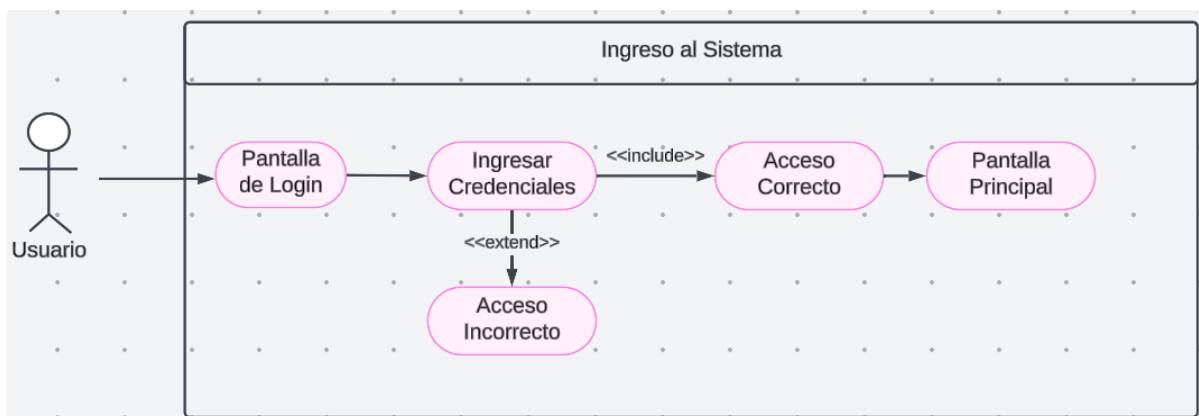
Casos de uso

Los casos de uso tienen como finalidad proporcionar una visión detallada de las acciones específicas que los usuarios pueden realizar dentro de cada módulo del sistema de inventarios.

Caso de Uso: clientes

Ilustración 3

caso de uso ingreso al sistema



Nota: Grafico UML sobre el caso de uso de ingreso del usuario al sistema

Tabla 12

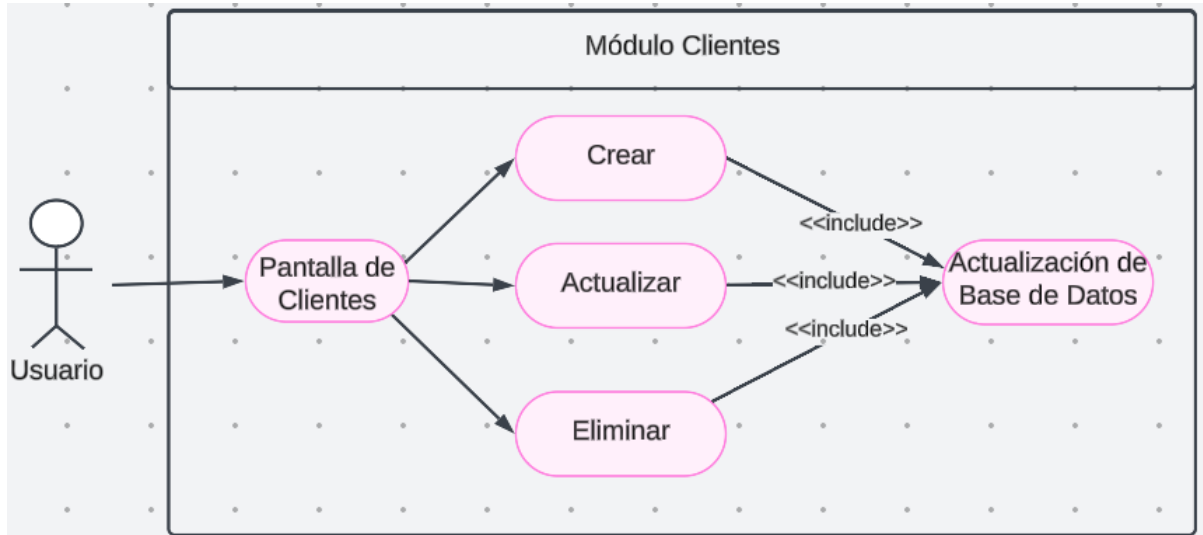
Ingreso del usuario al sistema

Descripción	Ingreso al sistema
Precondición	No haber iniciado sesión
Entrada	Usuario y contraseña
Proceso	Ingresar al sistema

Nota: caso de uso ingreso al sistema

Ilustración 4

caso de uso modulo clientes



Nota: Grafico UML sobre el módulo clientes

Tabla 13

Creación de clientes

Descripción	Creación de clientes
Precondición	Iniciar sesión y acceder al módulo clientes
Entrada	Formulario cliente: identificación, nombre, dirección, teléfono, email
Proceso	Crea al cliente en la fuente de datos

Nota: caso de uso creación de clientes

Tabla 14

Edición clientes

Descripción	Editar clientes

Precondición	Iniciar sesión y acceder al módulo clientes, seleccionar cliente
Entrada	Formulario cliente: nombre, dirección, teléfono, email
Proceso	Actualiza el cliente en la fuente de datos

Nota: caso de uso actualización de clientes

Tabla 15

Eliminación clientes

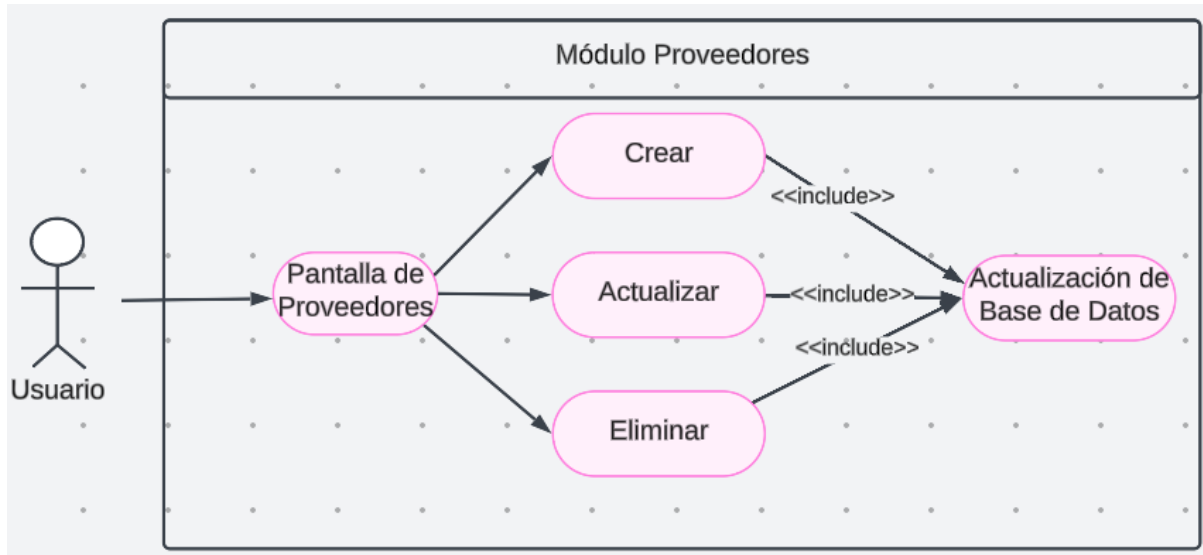
Descripción	Eliminar clientes
Precondición	Iniciar sesión y acceder al módulo clientes, seleccionar cliente
Entrada	Ninguna
Proceso	Elimina el cliente de la fuente de datos

Nota: caso de uso eliminación de clientes

Caso de Uso: Gestión de Proveedores

Ilustración 5

Caso de uso vista proveedores



Nota: Grafico UML sobre el módulo proveedores

Tabla 16

Creación proveedores

Descripción	Creación proveedores
Precondición	Iniciar sesión y acceder al módulo proveedores
Entrada	Formulario proveedor: identificación, nombre, dirección, teléfono, email
Proceso	Crea al proveedor en la fuente de datos

Nota: caso de uso creación de proveedores

Tabla 17

Edición proveedores

Descripción	Editar proveedores
Precondición	Iniciar sesión y acceder al módulo proveedores, seleccionar proveedor

Entrada	Formulario proveedor: nombre, dirección, teléfono, email
Proceso	Actualiza el proveedor en la fuente de datos

Nota: caso de uso actualización de proveedores

Tabla 18

Eliminación proveedores

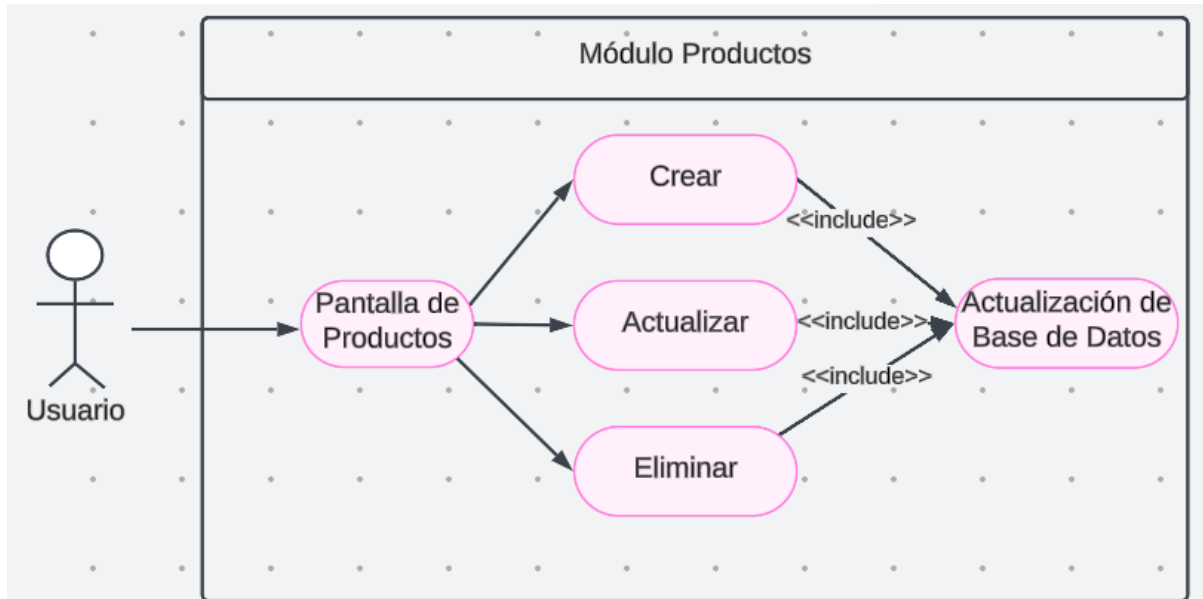
Descripción	Eliminar proveedores
Precondición	Iniciar sesión y acceder al módulo proveedores, seleccionar proveedor
Entrada	Ninguna
Proceso	Elimina el proveedor de la fuente de datos

Nota: caso de uso eliminación de proveedores

Caso de Uso: Gestión de Productos y Servicios

Ilustración 6

caso de uso modulo productos



Nota: Grafico UML sobre el módulo productos

Tabla 19

Creación de productos

Descripción	Creación de productos
Precondición	Iniciar sesión y acceder al módulo productos
Entrada	Formulario de productos: nombre, descripción, precio compra, precio venta, cantidad, categoría, marca, modelo y proveedores
Proceso	Crea el producto en la fuente de datos

Nota: caso de uso creación de productos

Tabla 20

Edición productos

Descripción	Editar productos
-------------	------------------

Precondición	Iniciar sesión y acceder al módulo productos, seleccionar producto
Entrada	Formulario de productos: nombre, descripción, precio compra, precio venta, cantidad, categoría, marca, modelo y proveedores
Proceso	Actualiza el producto en la fuente de datos

Nota: caso de uso actualización del producto

Tabla 21

Eliminación productos

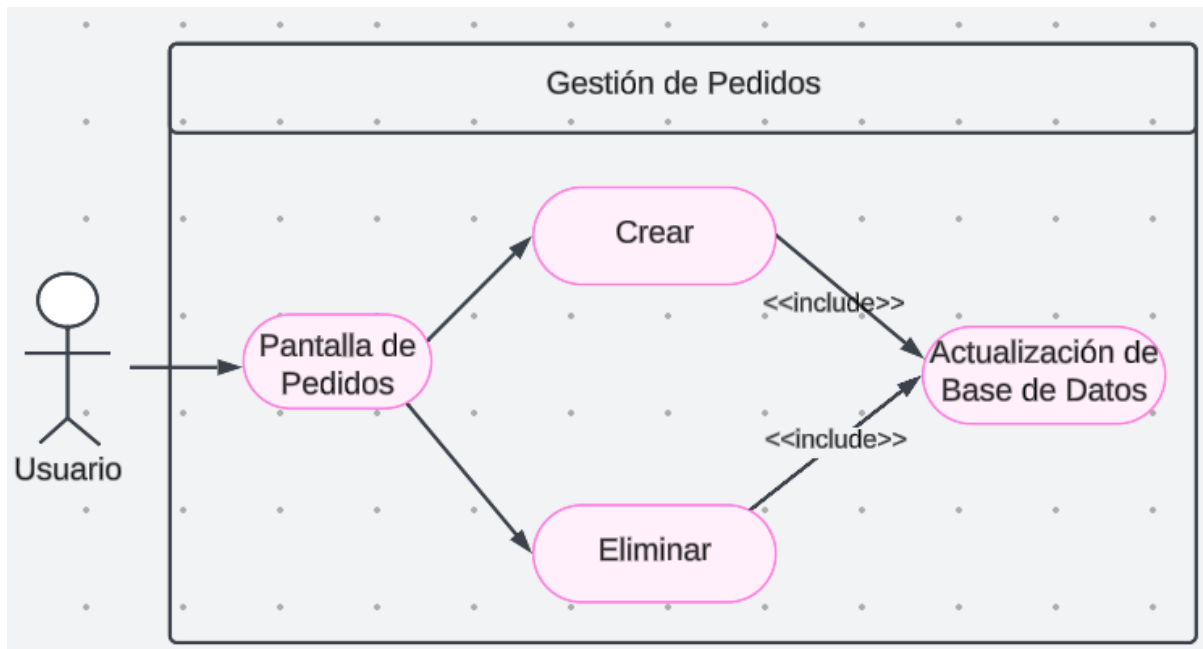
Descripción	Eliminar productos
Precondición	Iniciar sesión y acceder al módulo productos, seleccionar producto
Entrada	Ninguna
Proceso	Elimina el producto de la fuente de datos

Nota: caso de uso eliminación del producto

Caso de Uso: Gestión de Pedidos

Ilustración 7

caso de uso gestión de pedidos



Nota: Grafico UML sobre la gestión de pedidos

Tabla 22

Creación pedidos

Descripción	Creación pedidos
Precondición	Iniciar sesión y acceder al módulo pedidos
Entrada	Proveedor elegido y producto del proveedor
Proceso	Crea el pedido en la fuente de datos

Nota: caso de uso creación del pedido

Tabla 23

Eliminación pedidos

Descripción	Eliminar pedidos
Precondición	Iniciar sesión y acceder al módulo pedidos, seleccionar pedido

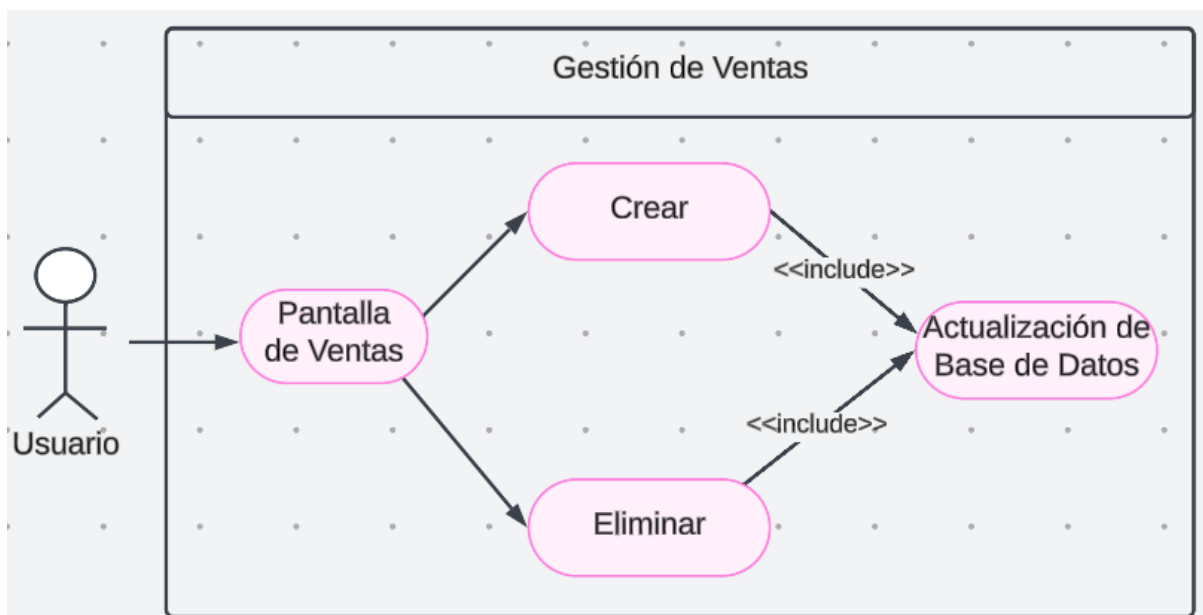
Entrada	Ninguna
Proceso	Elimina el pedido de la fuente de datos

Nota: caso de uso eliminación del pedido

Caso de Uso: Gestión de Ventas

Ilustración 8

Caso de uso ventas



Nota: Grafico UML sobre la gestión de Ventas

Tabla 24

Creación ventas

Descripción	Creación ventas
Precondición	Iniciar sesión y acceder al módulo ventas
Entrada	identificación, nombre, dirección, teléfono, email y producto de la venta
Proceso	Mostrar la información de las ventas, se crea un cliente en caso de que no exista y

	resta un producto del stock en inventario generando la venta
--	---

Nota: caso de uso creación de la venta

Tabla 25

Eliminación ventas

Descripción	Eliminar ventas
Precondición	Iniciar sesión y acceder al módulo ventas, seleccionar venta
Entrada	Ninguna
Proceso	Eliminar la venta

Nota: caso de uso eliminación de la venta

Capítulo III Desarrollo

Requerimientos del sistema

Realizado el análisis del proyecto e identificando los principales procesos del sistema es posible identificar lo que el aplicativo web necesita solventar de manera activa y pasiva a estos requisitos se los categoriza como funcionales aquellos que impactan directamente en la funcionalidad del sistema y no funcionales cuyos no afectan la funcionalidad, pero si atacan a la usabilidad y experiencia del usuario.

Requerimientos funcionales:

Modulo Clientes

El sistema debe ser capaz de actualizar y eliminar clientes validando sus datos en el formulario de ingreso.

Modulo Proveedores

El proyecto debe crear, actualizar y eliminar proveedores y los mismos deben estar preparados para asociarse a un producto.

Modulo Ventas

En este módulo el programa debe poder crear ventas y clientes si es que se ingresan los datos de un comprador que no esté registrado, así como realizar el proceso respectivo para que el stock de ese producto disminuya acorde a la venta.

Modulo Pedidos

El aplicativo web debe poder registrar los pedidos de cada producto, así como realizar el proceso respectivo para que el stock de ese producto aumente acorde al pedido.

Modulo Reportes

La herramienta debe mostrar el total de ventas y pedidos con la suma total monetaria, se debería poder filtrar por fechas y generar un documento Excel de esta información.

Modulo Caja

Para la gestión de caja se deberá presentar el total del valor de ventas en efectivo y transferencia con la posibilidad de filtrar por fechas.

Modulo Usuarios

Los usuarios deberán poder crearse, editarse y eliminarse en el presente modulo, cada uno de los usuarios creado deberá tener acceso al sistema y su propia visualización de módulos de acuerdo con el rol.

Requerimientos no funcionales:

El sistema deberá proveer la información de caja y los reportes de ventas siempre actualizados al último registro correctamente filtrados de ser el caso. El tiempo de respuesta al actualizar cualquier producto, usuario, proveedor, cliente o venta debe ser inmediato.

La contraseña de cada uno de los usuarios debe ser encriptada tanto para la vista de los módulos como en la fuente de datos, deben existir roles que limiten la visualización de módulos a los correspondientes dependiendo del usuario ingresado. El interfaz debe ser fácil de usar e intuitivo al usuario para la fácil adaptación de los consumidores. La herramienta debe tener la posibilidad de expandirse o añadir otro modulo con facilidad para nuevas funcionalidades relacionadas los módulos anteriores.

Cronograma

Para la construcción del sistema la organización es esencial por lo cual a continuación se presentan los cronogramas correspondientes a las tareas y desarrollo del proyecto.

Tabla 26*Cronograma de trabajo del back end*

Módulo	Iteración	Día de inicio	Requerimiento	Prioridad	Tarea	Responsable	Duración (días)
Investigación	1	1	Análisis de requisitos	Alta	Investigación de herramientas	Daniela Meneses	5
Levantamiento del ambiente y fuente de datos	2	5	Levantamiento del ambiente y fuente de datos	Alta	Levantamiento de fuente de datos, dependencias y servidor en node.js	Daniela Meneses	10
Diseño de backend	3	8	Diseño e interactividad de módulos desde el backend	Alta	Desarrollo de modularidad y atar los módulos	Daniela Meneses	15
Configuración y ajustes	4	21	Configuración de entrada, mensajes de éxito y error	Alta	Disponibilidad de mensajes y entradas	Daniela Meneses	8

Pruebas	6	34	Pruebas de funcionamiento de todos los endpoints	Alta	Pruebas de flujo y validaciones	Daniela Meneses	6
---------	---	----	--	------	---------------------------------	-----------------	---

Nota: Cronograma del proyecto en el área del back end

Tabla 27

Cronograma de trabajo del front end

Módulo	Iteración	Día de inicio	Requerimiento	Prioridad	Tarea	Responsable	Duración (días)
Investigación	1	1	Análisis de requisitos	Alta	Investigación de herramientas	Joan Mera	5
Levantamiento del ambiente	2	5	Levantamiento del ambiente y estabilización de procesos	Alta	Levantamiento de librerías y sistema en general configurado con MUI	Joan Mera	3
Diseño del frontend	3	8	Diseño y interactividad de módulos	Alta	Desarrollo del diseño y atar los módulos	Joan Mera	15
Configuración y ajustes	4	21	Configuración de entrada, mensajes de éxito y error	Alta	Disponibilidad de mensajes y entradas	Joan Mera	8
Validaciones	5	29	Validaciones de los formularios	Alta	Poner validaciones dentro de los formularios de creación	Joan Mera	5
Pruebas	6	34	Pruebas de funcionamiento	Alta	Pruebas de flujo y validaciones	Joan Mera	3

Construcción

Dentro del sistema la lógica implementada fue fundamental para la construcción de un flujo fluido que permita la interoperabilidad entre distintos módulos conectados a través de postgresql, detalles como la sincronización del stock y la creación embebida de registros fue clave para el sistema de inventario

A continuación, un ejemplo de una de las secciones más relevantes en la construcción del programa:

Ilustración 9

Función para verificar la creación de clientes

```
346
347 // Verificar si el cliente ya existe o crear uno nuevo
348 if (!clienteIdFinal && newClient.identificacion) {
349     const clienteExistenteId = await checkIfClientExists(
350         newClient.identificacion
351     );
352     if (clienteExistenteId) {
353         clienteIdFinal = clienteExistenteId;
354     } else {
355         const clienteIdCreado = await handleCreateClient();
356         if (!clienteIdCreado) {
357             notifyError("No se puede crear un cliente");
358             return;
359         }
360         clienteIdFinal = clienteIdCreado;
361     }
362     setClienteId(clienteIdFinal);
363 }
364
```

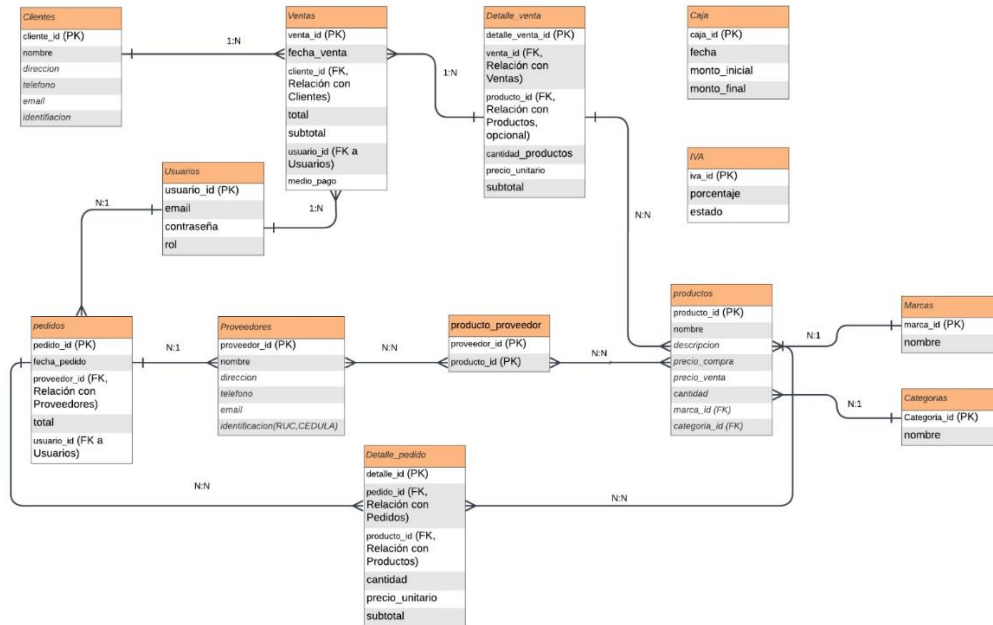
Nota: código que implica la verificación de creación de clientes

Diagrama de clases

Para desarrollar el backend, primero definiremos la estructura y la forma en que nuestras clases interactúan entre sí. A continuación, presentamos la organización de estas clases en la siguiente imagen.

Ilustración 10

Diagrama UML de clases



Nota: Diagrama de clases utilizado para el desarrollo

Diagrama de Secuencia

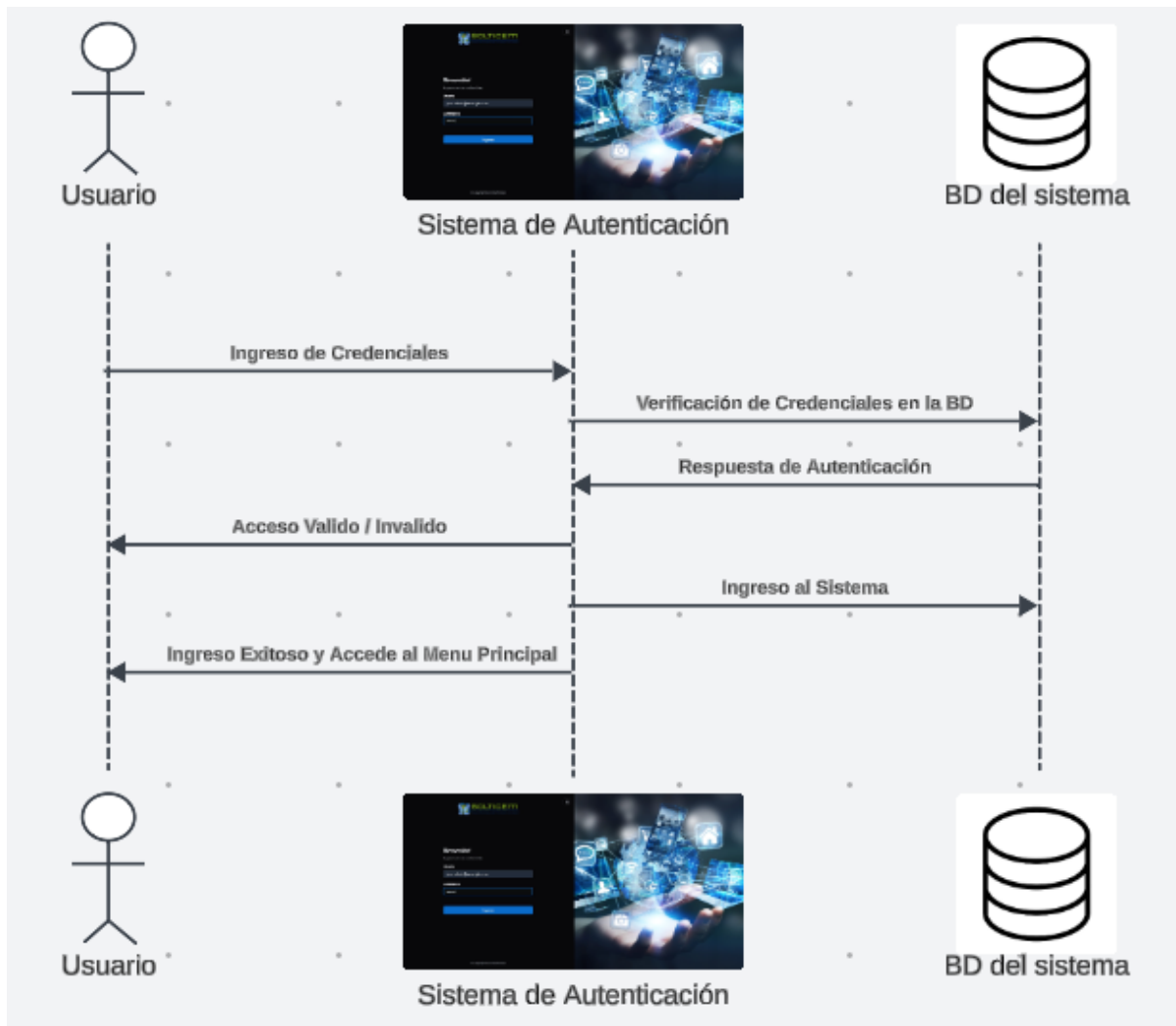
Otorga una vista sobre la interacción entre los objetos del sistema ante un proceso en un determinado tiempo de manera ordenada representando un flujo

Ingreso al sistema:

En el presente diagrama se muestra el proceso de ingreso al sistema verificando las credenciales del usuario.

Ilustración 11

Diagrama de secuencia para el inicio de sesión



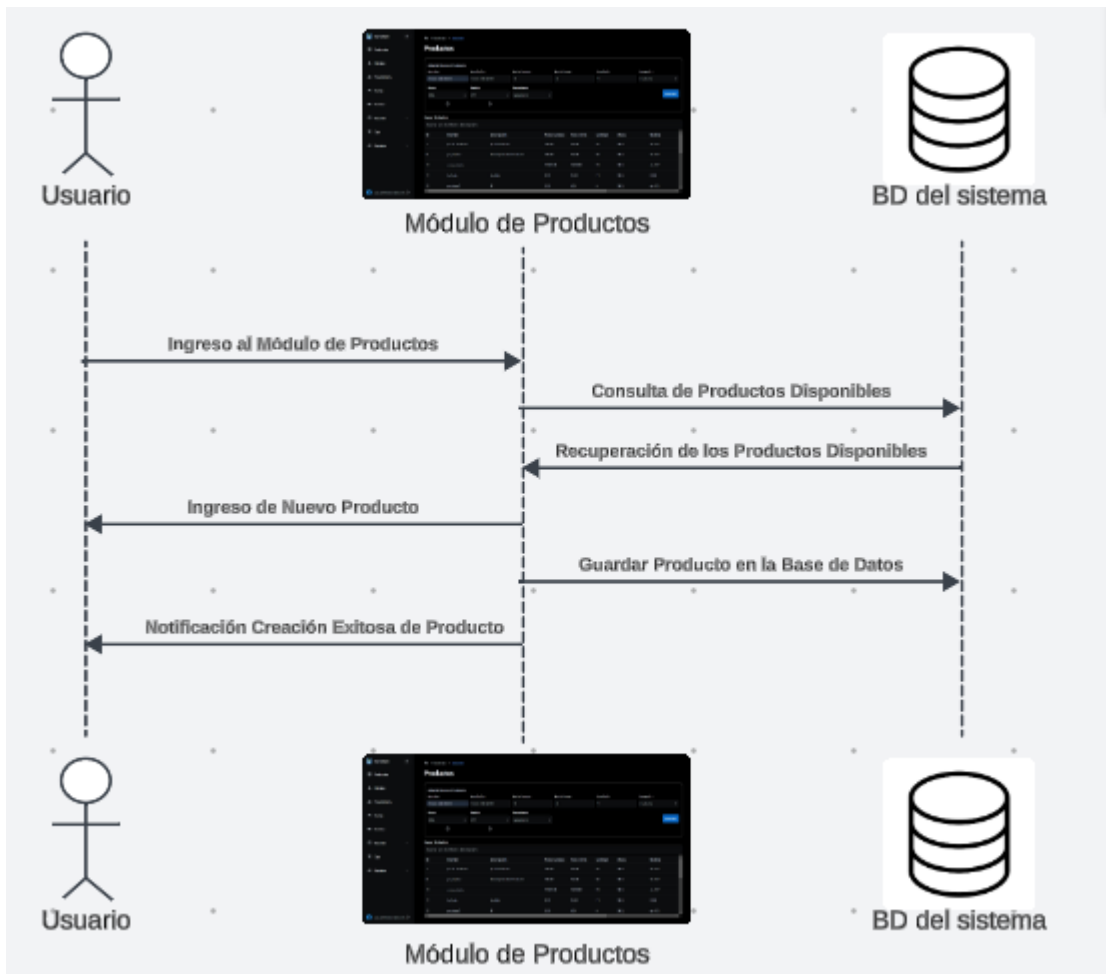
Nota: Diagrama de secuencia que debe seguirse a la hora de iniciar sesión

Ingreso de Nuevo Producto:

Se presenta a través del diagrama el proceso que debe seguirse a la hora de crear un nuevo producto dentro de la plataforma.

Ilustración 12

Diagrama de secuencia para el ingreso de nuevos productos



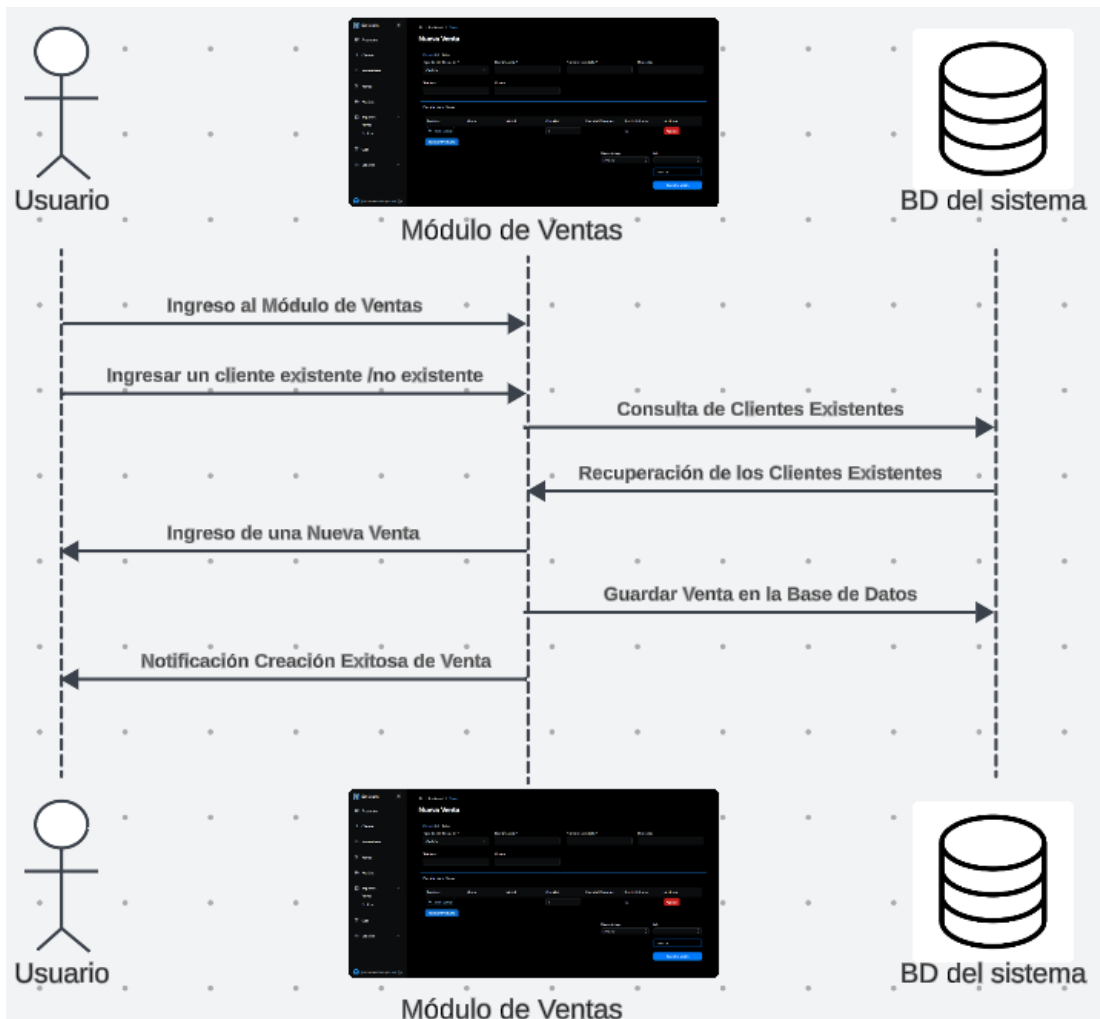
Nota: Diagrama de secuencia que debe seguirse a la hora de crear un producto

Ingreso de Nueva Venta:

Se presenta a través del diagrama el proceso que debe seguirse a la hora de crear una venta dentro de la plataforma.

Ilustración 13

Diagrama de secuencia para el ingreso de una venta



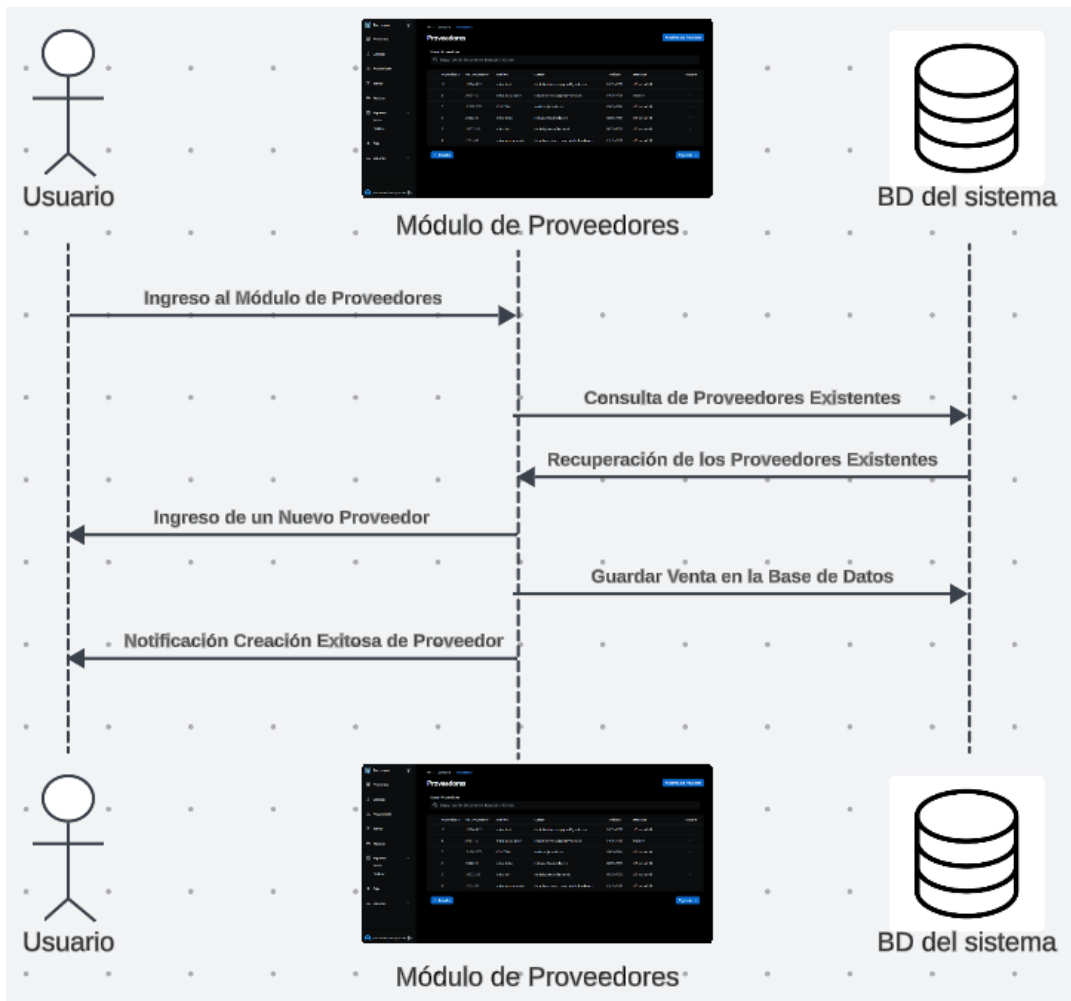
Nota: Diagrama de secuencia para la creación una venta

Ingreso de Nuevo proveedor:

Se presenta a través del diagrama el proceso que debe seguirse a la hora de crear un nuevo proveedor dentro de la plataforma.

Ilustración 14

Diagrama de secuencia para el ingreso de nuevos proveedores



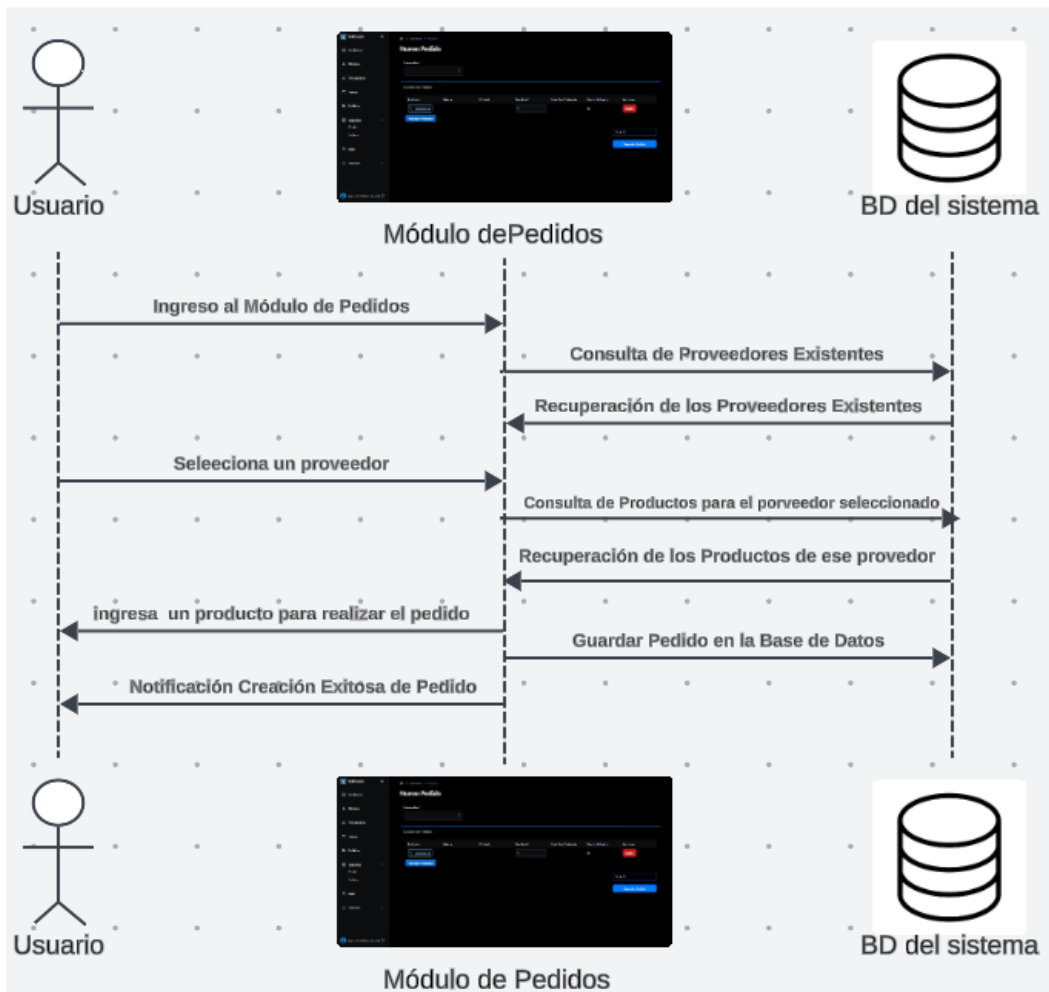
Nota: Diagrama de secuencia que debe seguirse a la hora de crear un proveedor

Ingreso de Nuevo Pedido:

Se presenta a través del diagrama el proceso que debe seguirse a la hora de crear un pedido dentro de la plataforma.

Ilustración 15

Diagrama de secuencia para el ingreso de un pedido



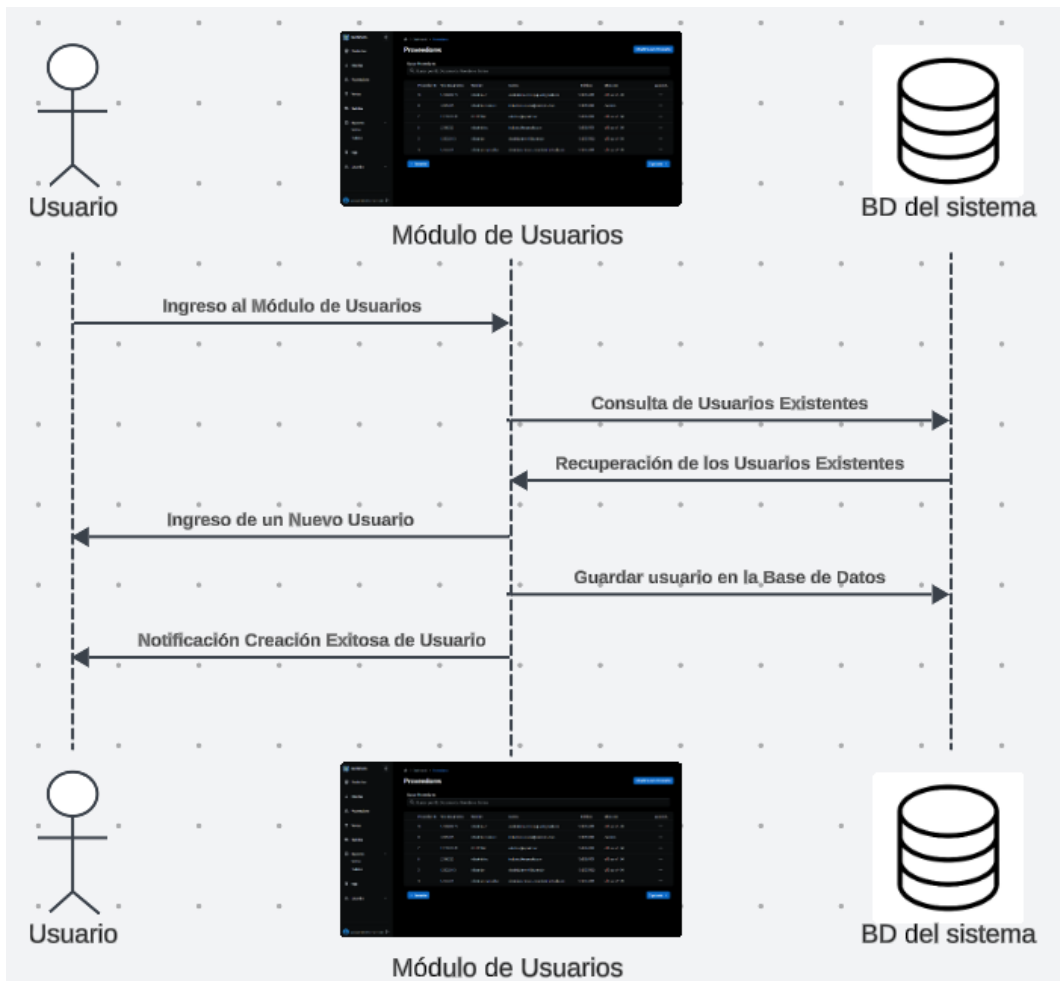
Nota: Diagrama de secuencia para la creación un pedido

Ingreso de Nuevo Usuario:

Se presenta a través del diagrama el proceso que debe seguirse a la hora de crear un usuario dentro de la plataforma

Ilustración 16

Diagrama de secuencia para el ingreso de nuevos usuarios



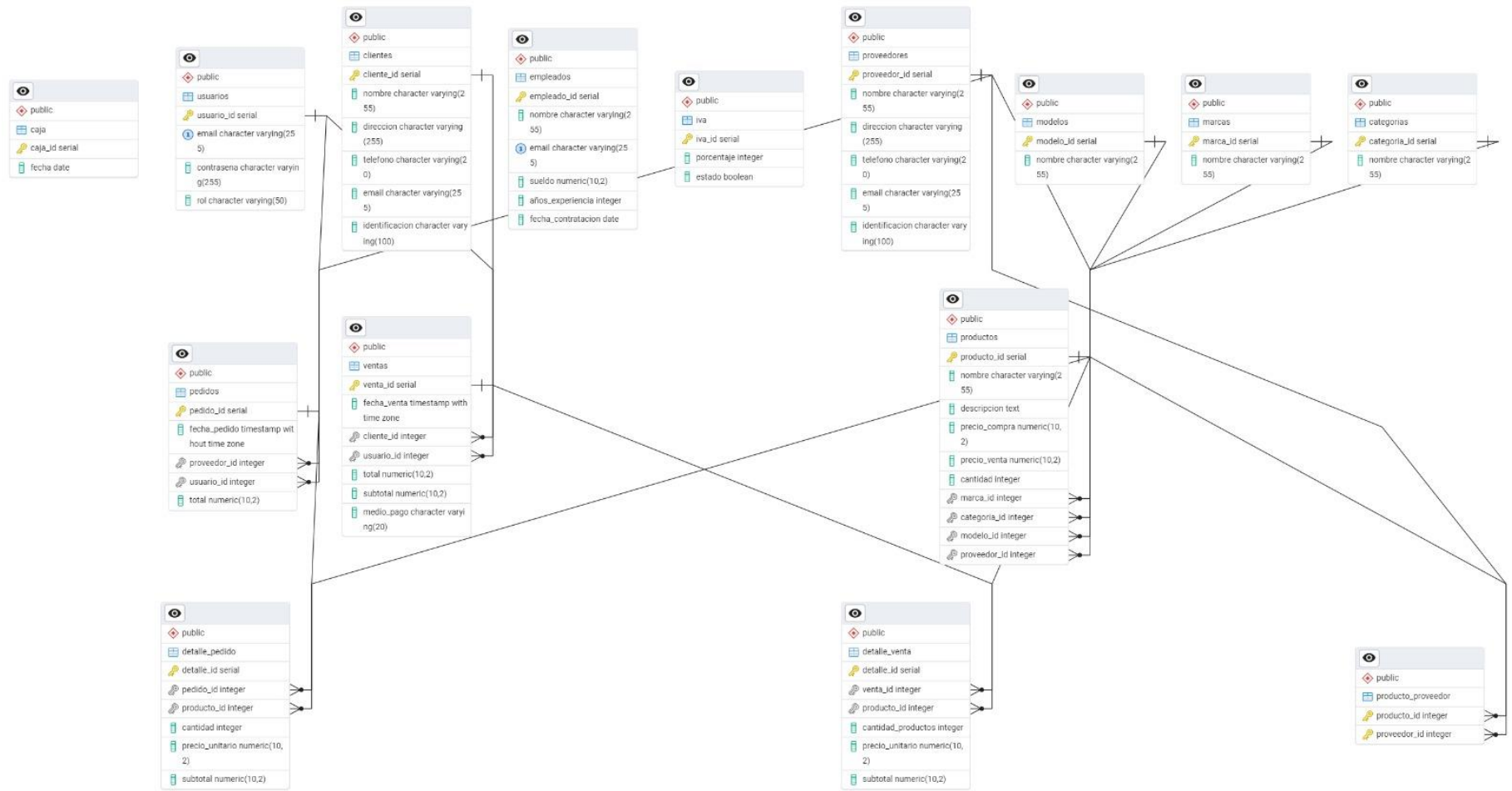
Nota: Diagrama de secuencia que debe seguirse a la hora de crear un usuario

Modelo BDD

A continuación, se mostrará un diagrama de cómo está construida la fuente de datos normalizada 1NF para el funcionamiento del programa.

Ilustración 17

Grafica de elaboración propia con ayuda de posgresql



Diccionario de datos

Este glosario de datos es un conjunto de información que permite una explicación detallada acerca de cómo está construida cada tabla utilizada para alojar información. Detalla cada tabla, los datos que guarda, el tipo de información que puede tener cada campo y cómo se conectan entre sí.

Tabla 28

Cientes

Columna	Tipo de Dato	Condiciones	Detalle
cliente_id	SERIAL	LLAVE PRIMARIA	Clave única del cliente en la tabla
nombre	CHARACTER VARYING(255)	NO NULO	Nombre del cliente
direccion	CHARACTER VARYING(255)	NULLABLE	Dirección del cliente
telefono	CHARACTER VARYING(20)	NULLABLE	Teléfono de contacto
email	CHARACTER VARYING(255)	NULLABLE	Correo electrónico del cliente
identificacion	CHARACTER VARYING(100)	NULLABLE	Identificación fiscal o RUC

Nota: Glosario de datos de la tabla Cientes

Tabla 29

Usuarios

Columna	Tipo de Dato	Condiciones	Detalle
---------	--------------	-------------	---------

usuario_id	SERIAL	LLAVE PRIMARIA	Clave única del usuario
email	CHARACTER VARYING(255)	UNICO, NO NULO	Correo electrónico del usuario
contrasena	CHARACTER VARYING(255)	NO NULO	Contraseña encriptada del usuario
rol	CHARACTER VARYING(50)	NO NULO	Rol del usuario en el sistema
usuario_id	SERIAL	LLAVE PRIMARIA	Identificador del usuario
email	CHARACTER VARYING(255)	UNICO, NO NULO	Dirección de correo del usuario

Nota: Glosario de datos de la tabla Usuarios

Tabla 30

Proveedores

Columna	Tipo de Dato	Condiciones	Detalle
proveedor_id	SERIAL	LLAVE PRIMARIA	Clave única del proveedor
nombre	CHARACTER VARYING(255)	NO NULO	Nombre del proveedor
direccion	CHARACTER VARYING(255)	NULLABLE	Dirección del proveedor
telefono	CHARACTER VARYING(20)	NULLABLE	Teléfono del proveedor

email	CHARACTER VARYING(255)	NULLABLE	Correo electrónico del proveedor
identificacion	CHARACTER VARYING(100)	NULLABLE	RUC o Identificación fiscal

Nota: Glosario de datos de la tabla Proveedores

Tabla 31

Categorías

Columna	Tipo de Dato	Condiciones	Detalle
categoria_id	SERIAL	LLAVE PRIMARIA	Clave única de la categoría
nombre	CHARACTER VARYING(255)	NO NULO	Nombre de la categoría

Nota: Glosario de datos de la tabla Categorías

Tabla 32

Marcas

Columna	Tipo de Dato	Condiciones	Detalle
marca_id	SERIAL	LLAVE PRIMARIA	Clave única de la marca
nombre	CHARACTER VARYING(255)	NO NULO	Nombre de la marca

Nota: Glosario de datos de la tabla Marcas

Tabla 33

Modelos

Columna	Tipo de Dato	Condiciones	Detalle
---------	--------------	-------------	---------

modelo_id	SERIAL	LLAVE PRIMARIA	Clave única del modelo
nombre	CHARACTER VARYING(255)	NO NULO	Nombre del modelo

Nota: Glosario de datos de la tabla Modelos

Tabla 34

Productos

Columna	Tipo de Dato	Condiciones	Detalle
producto_id	SERIAL	LLAVE PRIMARIA	Clave única del producto
nombre	CHARACTER VARYING(255)	NO NULO	Nombre del producto
descripcion	TEXT	NULLABLE	Detalle del producto
precio_compra	DECIMAL(10,2)	NULLABLE	Precio de compra del producto
precio_venta	DECIMAL(10,2)	NULLABLE	Precio de venta del producto
cantidad	INTEGER	NULLABLE	Cantidad disponible en stock
marca_id	INTEGER	LLAVE FORANEA (Marcas)	Relación con la tabla de marcas
categoria_id	INTEGER	LLAVE FORANEA (Categorias)	Relación con la tabla de categorías

modelo_id	INTEGER	LLAVE FORANEA (Modelos)	Relación con la tabla de modelos
-----------	---------	-------------------------------	----------------------------------

Nota: Glosario de datos de la tabla Productos

Tabla 35

Ventas

Columna	Tipo de Dato	Condiciones	Detalle
venta_id	SERIAL	LLAVE PRIMARIA	Clave única de la venta
fecha_venta	TIMESTAMP WITH TIME ZONE	NO NULO, DEFAULT NOW()	Fecha y hora de la venta
cliente_id	INTEGER	LLAVE FORANEA (Clientes)	Cliente que realizó la compra
usuario_id	INTEGER	LLAVE FORANEA (Usuarios)	Usuario que registró la venta
total	DECIMAL(10,2)	NULLABLE	Total de la venta
subtotal	DECIMAL(10,2)	NULLABLE	Subtotal antes de impuestos

Nota: Glosario de datos de la tabla Ventas

Tabla 36

Detalle venta

Columna	Tipo de Dato	Condiciones	Detalle
---------	--------------	-------------	---------

detalle_id	SERIAL	LLAVE PRIMARIA	Clave única del detalle de venta
venta_id	INTEGER	LLAVE FORANEA(Ventas) ON DELETE CASCADE	Relación con la tabla Ventas
producto_id	INTEGER	LLAVE FORANEA(Productos)	Producto vendido
cantidad_productos	INTEGER	NO NULO	Cantidad de productos vendidos
precio_unitario	DECIMAL(10,2)	NO NULO	Precio unitario del producto
subtotal	DECIMAL(10,2)	NULLABLE	Subtotal de la línea de venta

Nota: Glosario de datos de la tabla Detalle venta

Tabla 37

Pedidos

Columna	Tipo de Dato	Condiciones	Detalle
pedido_id	SERIAL	LLAVE PRIMARIA	Clave única del pedido
fecha_pedido	DATE	NO NULO	Fecha en que se realizó el pedido
proveedor_id	INTEGER	LLAVE FORANEA(Proveedores)	Proveedor del pedido

usuario_id	INT	LLAVE FORANEA(Usuarios)	Usuario que registró el pedido
total	DECIMAL(10,2)	NULLABLE	Total del pedido

Nota: Glosario de datos de la tabla Pedidos

Tabla 38

Detalle pedido

Columna	Tipo de Dato	Condiciones	Detalle
detalle_id	SERIAL	LLAVE PRIMARIA	Clave única del detalle de pedido
pedido_id	INTEGER	LLAVE FORANEA(Pedidos)	Relación con la tabla Pedidos
producto_id	INTEGER	LLAVE FORANEA(Productos)	Producto solicitado
cantidad	INTEGER	NO NULO	Cantidad solicitada del producto
precio_unitario	DECIMAL(10,2)	NO NULO	Precio unitario del producto
subtotal	DECIMAL(10,2)	NULLABLE	Subtotal del pedido

Nota: Glosario de datos de la tabla Detalle_pedido

Tabla 39

Caja

Columna	Tipo de Dato	Condiciones	Detalle
caja_id	SERIAL	LLAVE PRIMARIA	Clave única de la caja

fecha	DATE	NO NULO	Fecha de operación en caja
-------	------	---------	----------------------------

Nota: Glosario de datos de la tabla Caja

Tabla 40

Producto proveedor

Columna	Tipo de Dato	Condiciones	Detalle
producto_id	INTEGER	LLAVE FORANEA(Productos) ON DELETE CASCADE	Relación con Productos
proveedor_id	INTEGER	LLAVE FORANEA(Proveedores) ON DELETE CASCADE	Relación con Proveedores

Nota: Glosario de datos de la tabla Producto_proveedor

Tabla 41

IVA

Columna	Tipo de Dato	Condiciones	Detalle
iva_id	SERIAL	LLAVE PRIMARIA	Clave única del IVA
porcentaje	INTEGER	NO NULO	Porcentaje del IVA (ejemplo: 12%)
estado	BOOLEAN	NO NULO	Indica si el IVA está activo o inactivo

Nota: Glosario de datos de la tabla IVA

Arquitectura

Para la construcción del sistema en el front end se utilizó React junto con Vite, lo que permitió un entorno de desarrollo más rápido y optimizado. Vite destaca por su capacidad de proporcionar una recarga en caliente instantánea y una compilación eficiente, lo que reduce significativamente los tiempos de espera durante el desarrollo.

Además, se implementó Material UI (MUI) como biblioteca de componentes para React, lo que facilitó la creación de una interfaz moderna y responsiva. MUI ofrece una amplia colección de componentes preconstruidos, con estilos personalizables y compatibles con los principios de diseño de Google Material Design, lo que agiliza la construcción de interfaces consistentes y profesionales.

En el back end, se utilizó Express.js, framework simplista y eficiente para Node.js que facilita la gestión de rutas, middleware y peticiones HTTP. Su flexibilidad permite integrar fácilmente bases de datos, autenticación y otras funcionalidades esenciales para el sistema. Gracias a Express.js, se logró una arquitectura escalable y modular, facilitando el mantenimiento y la ampliación del sistema a futuro.

Capítulo IV Pruebas y Resultados

Pruebas y Resultados

Finalmente presentamos las pruebas que fueron necesarias para lograr un flujo correcto del sistema.

Tabla 42

Iteración 1 ingreso de un producto

Sección	Iteración	Prerrequisito	Entidad	Proceso	Proceso secundario	Acción	Resultado	Comentarios
Gestión de Productos	1	El sistema debe permitir ingresar un producto.	Administrador Usuario	Ingresar Producto	Insertar producto en BD.	Ingresar datos del producto.	50%	El sistema no detecta que el producto no tiene stock y se hace la venta con stock negativo.

Nota: tabla de la primera iteración de prueba para el ingreso de un producto

Tabla 43*Iteración 2 ingreso de un producto*

Sección	Iteración	Prerrequisito	Entidad	Proceso	Proceso secundario	Acción	Resultado	Comentarios
Gestión de Productos	2	El sistema debe permitir ingresar un producto y verificar si tiene stock.	Administrador Usuario	Ingresar Producto	Consultar existencia en BD	Ingresar datos del producto y verificar stock.	100%	El sistema detecta que el producto no tiene stock y no permite hacer la venta hasta seleccionar una cantidad valida del producto.

Nota: tabla de la segunda iteración de prueba para el ingreso de un producto

Tabla 44*Iteración 1 ingreso de una venta*

Sección	Iteración	Precondición	Entidad	Proceso	Proceso secundario	Acción	Resultado	Comentarios
Gestión de Ventas	1	El sistema debe registrar la venta correctamente.	Administrador Usuario	Registrar Venta	Insertar venta en BD	Seleccionar productos, ingresar datos del cliente y confirmar la venta.	70%	El sistema no valida si las credenciales y la información del cliente ingresado son correctas antes de hacer la venta.

Nota: tabla de la primera iteración de prueba para el ingreso de una venta

Tabla 45*Iteración 2 ingreso de una venta*

Sección	Iteración	Precondición	Entidad	Proceso	Proceso secundario	Acción	Resultado	Comentarios
Gestión de Ventas	2	El cliente debe crearse correctamente validando RUC o cédula.	Administrador Usuario	Registrar Venta	Validar RUC/Cédula en BD	Ingresar datos del cliente y validar RUC/cédula.	100%	El sistema valida el cliente y registra la venta.

Nota: tabla de la segunda iteración de prueba para el ingreso de una venta

Tabla 46*Ingreso de un proveedor*

Sección	Iteración	Precondición	Entidad	Proceso	Proceso secundario	Acción	Resultado	Comentarios
Gestión de Proveedores	1	El proveedor debe añadirse correctamente.	Administrador Usuario	Ingresar Proveedor	Insertar proveedor en BD	Completar datos y confirmar el registro.	70%	El sistema no valida si las credenciales y la información del proveedor ingresado son correctas antes de hacer la venta.

Nota: tabla de la primera iteración de prueba para el ingreso de un proveedor

Tabla 47*Iteración 2 ingreso de un proveedor*

Sección	Iteración	Precondición	Entidad	Proceso	Proceso secundario	Acción	Resultado	Comentarios
Gestión de Proveedores	2	El sistema debe permitir ingresar un proveedor y validar su RUC.	Administrador Usuario	Ingresar Proveedor	Validar RUC en BD	Ingresar datos del proveedor y verificar RUC.	100%	El sistema valida el proveedor y lo registra correctamente.

Nota: tabla de la segunda iteración de prueba para el ingreso de un proveedor

Tabla 48*Iteración 1 gestión de reportes*

Sección	Iteración	Precondición	Entidad	Proceso	Proceso secundario	Acción	Resultado	Comentarios
Gestión de Reportes	1	El sistema debe permitir buscar ventas por fecha.	Administrador Usuario	Generar Reporte	Filtrar ventas en BD	Buscar ventas por un rango de fechas.	90%	Las ventas filtradas de un mismo día (ej. 6 a 6) no aparecen debido a un problema en la filtración.

Nota: tabla de la primera iteración de prueba para reporte de ventas

Tabla 49

Iteración 2 gestión de reportes

Sección	Iteración	Precondición	Entidad	Proceso	Proceso secundario	Acción	Resultado	Comentarios
Gestión de Reportes	2	El sistema debe corregir el filtrado de fechas.	Administrador Usuario	Generar Reporte	Ajuste en consulta SQL	Buscar ventas por un rango de fechas.	100%	Ahora las ventas de un mismo día aparecen correctamente en el reporte.

Nota: tabla de la segunda iteración de prueba para reporte de ventas

Conclusiones

Se ha determinado que la implementación del sistema permitirá automatizar el control de ventas, ingresos e inventario, reduciendo la dependencia de registros manuales y mejorando la persistencia de los datos. Gracias a la digitalización, será posible una gestión más eficiente, minimizando errores humanos y agilizando los procesos de registro en la empresa SOLTICEM TECHNOLOGYS.

El uso de ReactJS con TypeScript en el frontend y Node.js en el backend permitió desarrollar una aplicación dinámica y escalable. Gracias a su arquitectura basada en tipado estático, componentes reutilizables y un backend eficiente, junto con PostgreSQL como fuente de datos, la aplicación ofrecerá flexibilidad y capacidad de adaptación para futuros crecimientos empresariales.

La aplicación mejorará la disponibilidad de información en tiempo real, permitiendo a los empleados acceder a datos actualizados sobre ventas e inventario. Esto resultará en una mayor agilidad operativa, optimizando el flujo de trabajo y mejorando la experiencia tanto para empleados como para clientes.

Recomendaciones

Para aprovechar al máximo las funcionalidades del sistema, se recomienda capacitar a los empleados en el uso eficiente de la plataforma, asegurando que puedan manejar correctamente las opciones de registro de ventas, inventario y clientes.

Se debe establecer un plan de mantenimiento que contemple revisiones periódicas del sistema, corrección de errores y la incorporación de nuevas funcionalidades según las necesidades del negocio. Además, se recomienda monitorear el rendimiento del servidor para garantizar un servicio estable.

Se recomienda considerar el uso de un ORM (Object-Relational Mapping) en el futuro para mejorar el manejo de los datos. Un ORM como Sequelize o TypeORM en Node.js facilitaría la interacción con la fuente de datos PostgreSQL al proporcionar una capa de abstracción que reduce la complejidad de las consultas SQL, mejora la seguridad y optimiza el mantenimiento del código. Además, permitiría una mayor escalabilidad y consistencia en la gestión de los datos a medida que la aplicación crezca.

Como mejora futura, se sugiere implementar un módulo de facturación electrónica para integrar la gestión de ventas con la emisión de comprobantes fiscales. Asimismo, se podría incluir herramientas avanzadas de análisis de datos que permitan generar reportes detallados sobre tendencias de ventas y stock.

Referencias bibliográficas

Por qué React JS es la tecnología front-end más favorecida por las startups. (2024, 12 julio). <https://www.ranktracker.com/es/blog/why-react-js-is-the-most-favored-front-end-technology-for-startups/>

Scrum: conceptos clave y cómo se aplica en la gestión de proyectos. (2024). Retrieved 15 December 2024, from <https://asana.com/es/resources/what-is-scrum>

auth0.com. (s. f.). *JWT.IO*. JSON Web Tokens - jwt.io. <https://jwt.io/>

Generator, B. (s. f.). *Bcrypt Generator - Online Hash Generator and Checker*. Bcrypt Generator. <https://bcrypt-generator.com/>

Inicio rápido – React. (s. f.). <https://es.react.dev/learn>

MUI. (s. f.). *MUI: The React component library you always wanted.* <https://mui.com/>

GeeksforGeeks. (2024, 18 octubre). *React Material UI*. GeeksforGeeks. <https://www.geeksforgeeks.org/react-material-ui/>

Kamada, R. (2022, 7 agosto). *Adding the Material UI CSS framework to a React application*. DEV Community. <https://dev.to/rodrigokamada/adding-the-material-ui-css-framework-to-an-react-application-4jkl>

GeeksforGeeks. (2023, 30 marzo). *Microsoft Azure Database for PostgreSQL*. GeeksforGeeks. <https://www.geeksforgeeks.org/microsoft-azure-database-for-postgresql/>