



**PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR**

**Unidad Académica de Formación Técnica y Tecnológica – PUCE TEC**

**APLICATIVO WEB PARA LA GESTION DE INVENTARIOS Y DESPACHO DE  
PRODUCTOS PARA CASA GUFLO**

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

**Autores: Wilson Oswaldo Mites Montalvo**

**Brandom Ariel Ortega Bermúdez**

**Tutor: Andrés Mauricio Chiliquina Tutacha**

**Quito, Ecuador 2026**

## Índice

Portada .....	1
Índice .....	2
Lista de tablas.....	4
Lista de figuras.....	5
Declaración y Autorización.....	1
Declaración y Autorización.....	2
Agradecimiento.....	1
Agradecimiento.....	2
Introducción .....	3
Antecedentes .....	4
Planteamiento del Problema.....	4
Justificación.....	5
Objetivo General .....	5
Objetivos Específicos.....	5
Capítulo I.....	7
13.1. Marco Teórico.....	7
13.2. Procesos Actuales.....	8
13.3. Infraestructura Actual.....	8
13.4. Marco de Trabajo Scrum.....	9
13.5. Sprints .....	11
Capítulo II.....	13
14.1. Análisis y Diseño .....	13
14.2. Análisis de Requisitos .....	14
14.3. Mapa de Navegación.....	15
14.4. Diseño de Arquitectura .....	16
14.5. Viabilidad Técnica .....	17
14.6. Viabilidad Económica.....	18
14.7 Casos de Uso .....	21
Capítulo III.....	30
15.1. Desarrollo.....	30
15.2. Requerimientos de Software .....	30
15.3. Cronograma.....	32
15.4. Diagrama de Clases.....	34
15.5. Diagrama de Secuencia .....	35
15.6. Código Fuente .....	35
15.7. Modelo Base de Datos .....	39
Capítulo IV .....	46
16.1. Pruebas y resultado .....	46

16.2. Tablas de iteración de cada prueba con sus versiones .....	47
Conclusiones .....	51
Recomendaciones.....	52
Referencias bibliográficas .....	53

## Lista de tablas

Tabla 1 Infraestructura Actual .....	8
Tabla 2 Sprints .....	11
Tabla 3 Análisis de Requisitos.....	14
Tabla 4 Viabilidad Técnica.....	17
Tabla 5 Viabilidad economica del sistema .....	19
Tabla 6 Costos de desarrollo.....	19
Tabla 7 Tabla de valores .....	20
Tabla 8 Caso de uso: Iniciar Sesión.....	22
Tabla 9 Caso de uso: Visualizar productos.....	22
Tabla 10 Caso de uso: Registrar producto .....	23
Tabla 11 Caso de uso: Editar producto .....	23
Tabla 12 Caso de uso: Consultar detalle de producto.....	24
Tabla 13 Caso de uso: Desactivar producto.....	24
Tabla 14 Caso de uso: Gestionar categorías .....	25
Tabla 15 Caso de uso: Gestionar bodegas .....	25
Tabla 16 Caso de uso: Gestionar proveedores.....	26
Tabla 17 Caso de uso: Gestionar clientes .....	26
Tabla 18 Caso de uso: Registrar ingreso de productos.....	27
Tabla 19 Caso de uso: Registrar despacho de productos.....	27
Tabla 20 Caso de uso: Consultar movimientos.....	28
Tabla 21 Caso de uso: Cerrar sesión.....	28
Tabla 22 Cronograma de trabajo.....	32
Tabla 23 Tabla roles.....	41
Tabla 24 Tabla usuarios .....	41
Tabla 25 Tabla unidades .....	41
Tabla 26 Tabla categorías .....	42
Tabla 27 Tabla proveedores.....	42
Tabla 28 Tabla bodegas .....	42
Tabla 29 Tabla clientes .....	42
Tabla 30 Tabla productos.....	43
Tabla 31 Tabla stock.....	44
Tabla 32 Tabla movimientos .....	44
Tabla 33 Tabla movimiento detalle .....	45
Tabla 34 Iteración 1 Ingreso de un producto .....	47
Tabla 35 Iteración 2 Ingreso de un producto .....	48
Tabla 36 Iteración 1 Despacho .....	48
Tabla 37 Iteración 2 Despacho .....	49
Tabla 38 Iteración 1 Inicio de sesión .....	49
Tabla 39 Iteración 2 Inicio de sesión .....	50

## Lista de figuras

Figura 1 Mapa de navegación .....	15
Figura 2 Grafico de la arquitectura .....	17
Figura 3 Diagrama de casos de uso.....	21
Figura 4 Diagrama de clases .....	34
Figura 5 Diagrama de secuencia .....	35
Figura 6 Función para redirigir al login .....	36
Figura 7 Interfaz del LogIn .....	37
Figura 8 Lógica despacho tipo caja .....	37
Figura 9 Interfaz de despacho tipo caja .....	38
Figura 10 MER .....	40

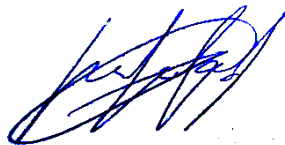
## Declaración y Autorización

Yo, Wilson Oswaldo Mites Montalvo con C.I. 1751524719 autor(a) del trabajo de titulación intitulado: “Aplicativo web para la gestión de inventarios y despacho de productos para Casa Guflo”, 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, 14 de febrero 2026



C.I. 1751524719

## **Declaración y Autorización**

Yo, Brandom Ariel Ortega Bermúdez con C.I. 1750649970 autor(a) del trabajo de titulación intitulado: “Aplicativo web para la gestión de inventarios y despacho de productos para Casa Guflo”, 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, 14 de febrero 2026



C.I. 175064997

## **Agradecimiento**

Agradezco a Dios por darme la fortaleza, constancia y salud para terminar esta etapa importante de mi vida.

Expreso mis más sinceros agradecimientos a la Pontificia Universidad Católica del Ecuador, en especial a la Unidad Académica de Formación Técnica y Tecnológica – PUCE TEC, por los conocimientos brindados a lo largo de mi estudio, los cuales han sido esenciales para mi desarrollo académico.

También agradezco de manera muy especial a nuestro tutor Andrés Mauricio Chiliquinga Tutacha, por su ayuda, orientación y compañía académica durante el desarrollo de mi tesis, así como por sus observaciones que contribuyeron a mejorar la calidad final del trabajo.

Finalmente, agradezco a mi familia por su apoyo incondicional, comprensión y por motivarme constantemente, así como a mi compañero de tesis por el trabajo colaborativo y su compromiso demostrado durante todo el tiempo de desarrollo del aplicativo y también agradezco de forma especial a mi novia por su apoyo incondicional y acompañarme en esas noches de desvelo y de igual manera a la mama de mi novia por permitirme hacer un aplicativo para su negocio ya que si no hubiera sido así, este trabajo de titulación no hubiera sido posible. Muchas gracias.

Wilson Oswaldo Mites Montalvo

## **Agradecimiento**

Agradezco a la universidad PUCE, especialmente al área de PUCE TEC, por la formación académica que me brindado a lo largo de este proceso, la cual ha fortalecido mis conocimientos profesionales y técnicos en el área del desarrollo de software.

De manera especial, expreso mi gratitud a nuestro tutor Andrés Mauricio Chiliquinga Tutacha, por su asesoría constante, paciencia y sobre todo por la disposición para orientar el desarrollo adecuado de este trabajo de titulación.

Para concluir y de forma especial agradezco a mi familia por el respaldo y apoyo brindado durante mi formación, a mi compañero de proyecto por el esfuerzo, destacando la responsabilidad y la colaboración demostrada, los cuales fueron fundamentales para la culminación exitosa de nuestro proyecto. Y de forma muy honorífica agradezco a mi novia por estar en los momentos donde sentía que no podía más ya que sin su motivación y acompañamiento esto no hubiese sido posible. Muchas gracias.

Brandom Ariel Ortega Bermúdez

## **Introducción**

El uso de un sistema digital de gestión de inventario en una empresa en crecimiento como Casa Guflo principalmente necesita de tres objetivos centrales: primero, la organización efectiva de un inventario de productos; segundo, la optimización en la administración y control de los despachos de productos; y, por último, el fortalecimiento de la ventaja competitiva en el entorno donde opera la empresa mediante la toma de decisiones más rápidas y precisas.

Contar con este tipo de servicios en la nube brinda fenomenales ventajas como el acceso remoto, la disminución de costos en equipamiento y servicios necesarios para sostener la infraestructura, y la escalabilidad del servicio en función del crecimiento de la empresa. Por ende, es necesaria la implementación de este tipo de soluciones en especial para Casa Guflo por sus características. Así mismo, el aplicativo que se va a desarrollar es amigable, es decir, muy fácil de usar y comprensible para aquellas personas que no poseen un nivel elevado de conocimiento técnico, por lo que su uso está al alcance y es rápido aprendizaje para todos sus empleados.

Para ajustar la solución a los requerimientos, se utilizarán metodologías ágiles en el proceso de desarrollo de manera que permita ejecutar muchas iteraciones y permitir el cambio de versiones de modo que se incorpore el feedback continuo. En cuanto a los productos, se usarán tecnologías actuales, por ejemplo, React para el desarrollo del interfaz.

## **Antecedentes**

La gestión de inventarios es algo esencial dentro de los procesos administrativos de los negocios, ya que permite controlar la disponibilidad de los artículos ofrecidos. La incorporación de tecnologías ha permitido a las empresas optimizar estos procesos mediante el uso de sistemas informáticos que ayudan al registro, almacenamiento y consulta de datos.

Con el paso del tiempo se ha evidenciado que el uso de aplicativos webs para la gestión de inventarios contribuye a reducir errores humanos, mejorar la organización de la información y apoyar la toma de decisiones estratégicas. Estas soluciones han sido ampliamente adoptadas en pequeños y medianos negocios debido a su accesibilidad, bajo costo de implementación y facilidad de uso. Muchos negocios comerciales continúan gestionando sus inventarios de forma manual o mediante herramientas básicas, como hojas de cálculo, lo que limita el control efectivo del stock y dificulta el seguimiento de ingresos y despachos. Casa Guflo, se encuentra dentro de este grupo de negocios que desean modernizar sus procesos administrativos para mejorar la eficiencia operativa y competitividad del negocio.

## **Planteamiento del Problema**

Casa Guflo no cuenta con un sistema informático que podría permitirle manejar su inventario de manera más eficaz. Hoy en día, el registro y control de mercancías se hace de manera manual, lo que provoca fallas, dificultad de control en el inventario y pérdida de tiempo en las gestiones administrativas. Esto impacta directamente al funcionamiento del negocio y en su capacidad de atender a sus clientes.

En el contexto actual que opera Casa Guflo muestra un incremento en la competencia a nivel local, en el cual el ofrecimiento de productos y atención al cliente son costos de

diferenciación. Al no contar con un sistema informático que facilite la gestión del inventario, la empresa no solo enfrenta problemas para mejorar su productividad interna, sino también compromete su posicionamiento competitivo actualmente.

Este proyecto propone desarrollar un aplicativo web que permita a Casa Guflo hacer la gestión de sus inventarios y despachos de productos de manera digital, segura y eficaz.

### **Justificación**

El desarrollo de un aplicativo web para la gestión de inventarios y despacho de productos en Casa Guflo se justifica por la necesidad de optimizar los procesos administrativos y mejorar el control de su inventario y movimientos. Al implementar esta solución se logrará agilizar el registro de movimientos de inventario y facilitar el acceso a información.

Desde el punto de vista operativo, el sistema ayudara a mejorar la organización del negocio, permitiendo un control más preciso del stock, el seguimiento de ingresos y despachos de los productos.

Desde el punto académico, el proyecto representa una oportunidad para aplicar los conocimientos adquiridos durante la formación en Desarrollo de Software.

### **Objetivo General**

Desarrollar un aplicativo web que permita a Casa Guflo gestionar el ingreso y despacho de productos de su inventario, mediante una plataforma web accesible y segura, con el propósito de optimizar el control de existencias y mejorar la eficiencia operativa de la empresa.

### **Objetivos Específicos**

- Diseñar la arquitectura del sistema web orientado a la gestión de inventario de Casa Guflo.

- Implementar un módulo para el registro de ingresos de productos al inventario.
- Implementar un módulo para el control de despachos de productos desde bodega.
- Desarrollar un módulo de reportes básicos que permita visualizar el estado del inventario en tiempo real.
- Validar la funcionalidad del sistema con pruebas técnicas y retroalimentación de usuario

## Capítulo I

### 13.1. Marco Teórico

La gestión de inventarios es un proceso clave dentro de los negocios dedicados a la comercialización, ya que permite mantener un control adecuado sobre las existencias y a su vez garantizar la disponibilidad de los productos. Un sistema de inventarios bien controlado ayuda a la reducción de pérdidas, mejora la toma de decisiones y fortalece la eficiencia operativa de un negocio.

Las aplicaciones web han ganado terreno en los últimos años debido a su accesibilidad, facilidad de mantenimiento y capacidad de ser utilizadas desde cualquier dispositivo solo con conexión a internet y ayuda a reducir el uso de registros manuales y mejorar la comunicación entre los distintos procesos administrativos de un negocio.

El modelo de arquitectura cliente-servidor es muy utilizado en el desarrollo de aplicaciones web, ya que separa la interfaz de usuario de la lógica de negocio y el almacenamiento de datos, dicha separación facilita la escalabilidad, el mantenimiento del código y la seguridad de la información.

En cuanto al manejo de datos, las bases de datos relacionales ayudan a estructurar la información de manera organizada, garantizando la integridad de los datos. PostgreSQL es un sistema de gestión de bases de datos relacional que resalta por su estabilidad, seguridad y cumplimiento de estándares SQL.

La metodología ágil Scrum prioriza el desarrollo paulatino de software mediante ciclos cortos de trabajo llamados sprints. Esta metodología promueve la colaboración del equipo, la adaptación a cambios y la entrega continua de funcionalidades, ya que son aspectos clave para el desarrollo eficiente.

### 13.2. Procesos Actuales

En la actualidad, Casa Guflo realiza su control de inventarios de manera manual, apoyándose en registros físicos básicos. Este enfoque genera dificultades para mantener información actualizada, aumenta el riesgo de errores y limita el registro del stock real disponible.

Las actividades relacionadas con el control de productos, ingresos y despachos recaen principalmente en una sola persona, lo que dificulta la continuidad del proceso en caso de ausencias.

Además, no se cuenta con flujos de procesos claramente definidos, lo que ocasiona inconsistencias en el manejo de la información.

### 13.3. Infraestructura Actual

La empresa cuenta con dispositivos básicos y acceso a internet, lo que permite la implementación de una solución basada en tecnologías web.

La infraestructura actual no busca plataformas de gestión centralizada y no ofrece soporte tecnológico para un control eficiente del inventario ni para la automatización de los procesos administrativos.

**Tabla 1**

*Infraestructura Actual*

Recurso	Descripción	Estado actual	Observaciones
Dispositivos	Computador portátil para gestión administrativa	Operativo	Uso compartido

Conectividad	Conexión a Internet doméstica	Operativo	Sin redundancia
Software de gestión	No existe sistema especializado	No disponible	Procesos manuales
Almacenamiento de datos	Archivos físicos y hojas de cálculo	Limitado	Riesgo de pérdida de información

### 13.4. Marco de Trabajo Scrum

Scrum es un marco de trabajo ágil orientado al desarrollo incremental de software, permitiendo una mejor adaptación a cambios y una comunicación constante entre los involucrados del proyecto (Schwaber & Sutherland, 2020).

Dentro del marco de trabajo Scrum se identifican los siguientes roles principales:

- **Product Owner:** responsable de definir y priorizar los requisitos del sistema. (Alaimo labs, 2024)
- **Scrum Master:** encargado de asegurar que se sigan las prácticas de Scrum y eliminar impedimentos. (Alaimo labs, 2024)
- **Equipo de desarrollo:** responsable de diseñar, desarrollar y probar el sistema. (Alaimo labs, 2024)

Las principales ceremonias de Scrum utilizadas en el proyecto son:

- **Sprint Planning:** planificación de las actividades a realizar en cada sprint. (Laoyan, 2026)
- **Daily Scrum:** reuniones breves para revisar el avance del trabajo. (Laoyan, 2026)

- **Sprint Review:** revisión de los resultados obtenidos al final del spring. (Laoyan, 2026)
- **Sprint Retrospective:** análisis de mejoras para el siguiente sprint. (Laoyan, 2026)

La adopción de Scrum permite una mejor organización, una respuesta rápida a los cambios y una entrega progresiva.

### 13.5. Sprints

El desarrollo del sistema se organizó en sprints, y se detallan en la siguiente tabla:

**Tabla 2**

*Sprints*

Sprint	Duración	Objetivo	Actividades principales	Entregables	Responsable
Sprint 1	Semanas 1 a 3	Establecer las bases del proyecto y definir los requerimientos	Configuración del entorno de desarrollo, creación del repositorio en GitHub, definición de requerimientos y diseño de la arquitectura	Entorno configurado, estructura base del proyecto, diseño inicial de arquitectura	Wilson Mites Brandom Ortega
Sprint 2	Semanas 4 a 6	Diseñar la experiencia de usuario del sistema	Creación de bocetos en Figma, diseño de vistas principales, definición de navegación y validación con usuarios	Diseño de navegación y experiencia de usuario	Brandom Ortega
Sprint 3	Semanas 7 a 9	Implementar la interfaz principal del sistema	Desarrollo de vistas en React, y autenticación visual, creación de componentes reutilizables	Interfaz funcional del sistema, navegación operativa	Brandom Ortega

---

Sprint 4	Semanas 10 a 12	Desarrollar los servicios del backend	Implementación de API REST, configuración de PostgreSQL, desarrollo de endpoints principales	API funcional, base de datos operativa, servicios conectados	Wilson Mites
Sprint 5	Semanas 13 a 15	Integrar y validar el sistema completo	Integración frontend-backend, pruebas funcionales y retroalimentación del usuario	Sistema integrado, resultados de pruebas	Wilson Mites
Sprint 6	Semanas 16 a 18	Finalizar el proyecto y preparar la entrega	Correcciones finales, documentación técnica y manual de usuario	Sistema finalizado, documentación completa, entrega académica	Wilson Mites

---

## Capítulo II

### 14.1. Análisis y Diseño

El desarrollo de este aplicativo web se basa en criterios técnicos orientados a la eficiencia, escalabilidad y mantenimiento del sistema. Una solución basada en tecnologías web responde a la necesidad de implementar una plataforma de fácil acceso, flexible y compatible con distintos entornos.

Desde una vista arquitectónica, el uso de un modelo cliente-servidor permite separar claramente las responsabilidades del sistema, donde la capa inicial se encarga de la interacción con el usuario y la lógica gestiona las reglas del negocio y los datos.

Asimismo, el uso de un aplicativo web resulta técnicamente esencial debido a su capacidad para operar en tiempo real, procesar múltiples solicitudes de manera concurrente y centrar la información. Esto ayuda a garantizar la integridad de los datos, la consistencia de las operaciones y un correcto uso de la información entre los distintos módulos del sistema.

El diseño modular del sistema ayuda a que cada componente funcione de manera independiente, ayudando a una escalabilidad futura. Este plan técnico también facilita la realización de pruebas, detección de errores y una futura adaptación del sistema a nuevas tecnologías.

Todo esto junto, el análisis y diseño del aplicativo se apoyan en principios de ingeniería de software, asegurando una solución alineada con buenas prácticas actuales en el desarrollo de aplicaciones web.

## 14.2. Análisis de Requisitos

El análisis de requisitos permitió analizar los aspectos clave que el sistema debe resolver.

Entre los principales puntos que se busca resolver se encuentran:

**Tabla 3**

### *Análisis de Requisitos*

Código	Necesidad Identificada	Descripción	Área Impactada
AR-01	Control centralizado de inventario	Registrar y consultar productos, categorías y stock desde un solo sistema	Inventarios
AR-02	Gestión de múltiples bodegas	Permitir el control de stock por bodega física	Logística
AR-03	Registro de ingresos y despachos	Automatizar entradas y salidas de productos para evitar errores manuales	Operaciones
AR-04	Visualización en tiempo real	Consultar existencias actualizadas sin cálculos manuales	Administración
AR-05	Seguridad de acceso	Restringir el acceso al sistema mediante autenticación de usuario administrador	Seguridad
AR-06	Historial de movimientos	Restringir el acceso al sistema mediante autenticación de usuario administrador	Control

Los puntos presentados sirvieron como base para definir el alcance del sistema y el diseño de sus funcionalidades.

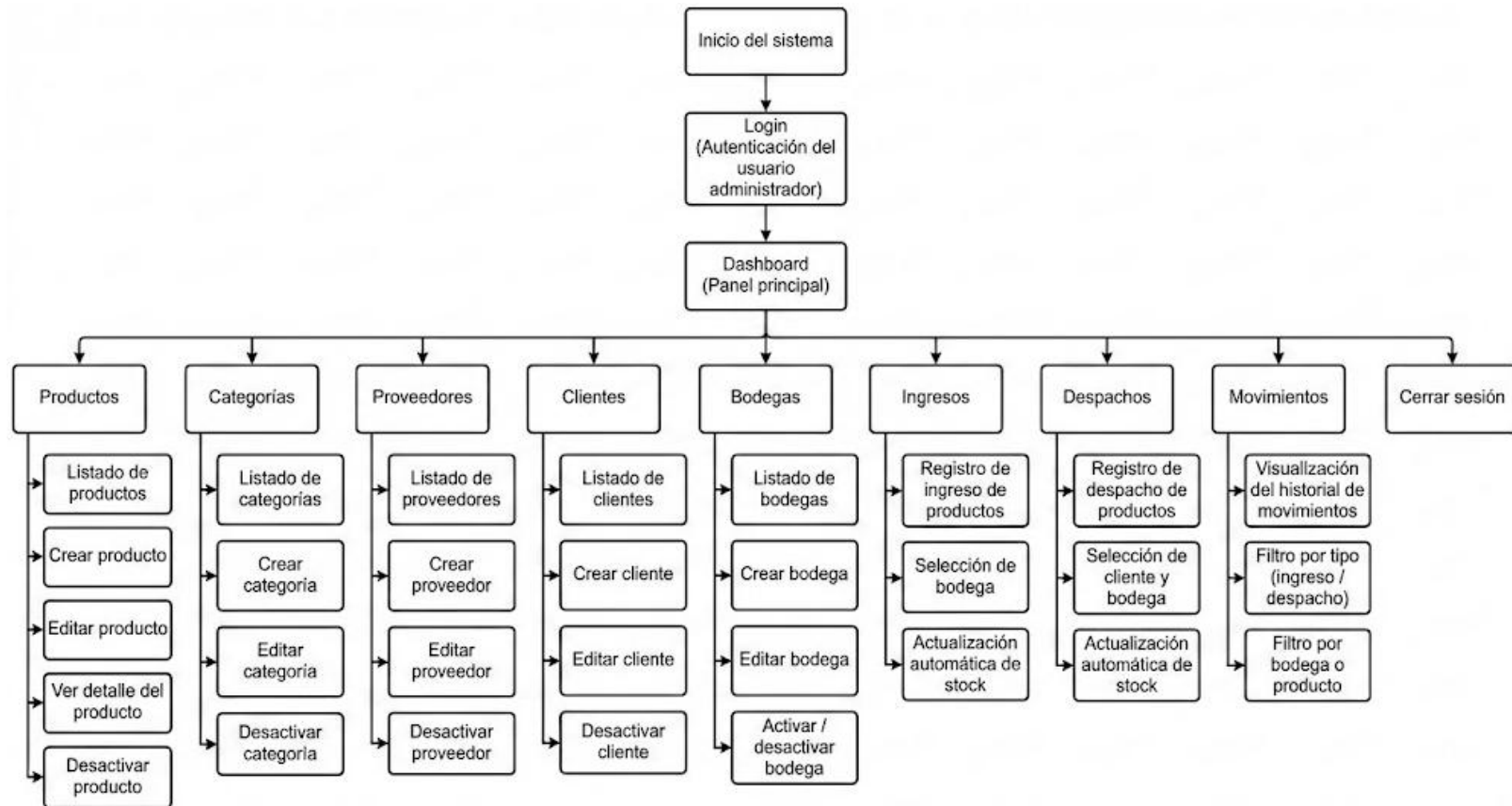
### 14.3. Mapa de Navegación

El mapa de navegación es un esquema, que se puede pensar como un árbol jerárquico, que representa la arquitectura de las páginas de un sitio web. (Fetecua, 2025)

En el sistema propuesto para Casa Guflo, el mapa de navegación contiene las siguientes secciones principales:

**Figura 1**

*Mapa de navegación*



## **14.4. Diseño de Arquitectura**

El sistema fue diseñado bajo una arquitectura cliente-servidor, separando las responsabilidades del frontend, backend y base de datos. Dicha separación facilita el mantenimiento, escalabilidad y desarrollo del sistema.

### **14.4.2 Frontend**

El frontend del aplicativo web fue desarrollado utilizando React.js, una librería de JavaScript orientado en el desarrollo de interfaces de usuario dinámicas. También nos ayuda a la creación de componentes reutilizables, el desarrollo de formularios, tablas y vistas. A su vez, contribuye a una experiencia de usuario fluida gracias a su eficaz manejo del estado y actualización de la interfaz.

### **14.4.3 Backend**

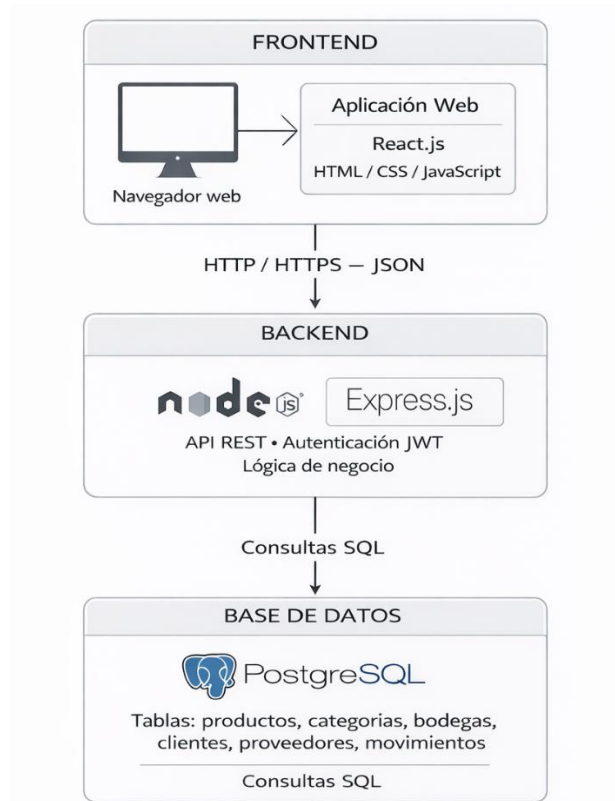
El backend del aplicativo fue creado utilizando Node.js como entorno de ejecución, junto con el framework Express.js para la construcción de una API REST. Esto se encarga de procesar las solicitudes del cliente, aplicar la lógica de negocio y gestionar la comunicación con la base de datos. También, se utilizó un sistema de autenticación mediante tokens JWT para garantizar el acceso seguro al aplicativo.

### **14.4.4 Base de datos**

Para guardar la información se utilizó PostgreSQL, un sistema de gestión de datos relacional reconocido por su estabilidad, seguridad y soporte. La base de datos almacena información relacionada con los productos, categorías, bodegas, stock, movimientos y usuario asegurando la integridad y consistencia de datos.

**Figura 2**

*Grafico de la arquitectura*



### 14.5. Viabilidad Técnica

El proyecto es técnicamente viable, ya que utiliza tecnologías ampliamente documentadas y compatibles. React.js, Node.js y PostgreSQL ayudan a desarrollar un sistema robusto, escalable y mantenible. De igual manera, el equipo de desarrollo posee conocimientos previos en estas tecnologías, lo que reduce riesgos técnicos durante la implementación del aplicativo.

**Tabla 4**

*Viabilidad Técnica*

Componente	Tecnología / Servicio	Descripción
------------	-----------------------	-------------

---

Servidor de la aplicación	Node.js + Express.js	Ejecuta lógica del backend y expone los servidores REST del sistema
Servidor Web	Navegador web moderno	Permite el acceso al sistema desde cualquier dispositivo
Servidor de base de datos	PostgreSQL	Almacena la información del inventario, productos y movimientos
Frontend	React.js	Interfaz gráfica para la gestión del sistema
Control de versiones	GitHub	Gestión del código fuente y trabajo colaborativo
Diseño de interfaz	Figma	Prototipado y validación de la interfaz gráfica
Sistema Operativo	Windows	Entorno de ejecución para desarrollo y pruebas

---

En base al análisis, se concluye que el proyecto es técnicamente viable ya que la solución propuesta utiliza herramientas y tecnologías de código abierto, ampliamente utilizadas.

#### **14.6. Viabilidad Económica**

La viabilidad económica es favorable, porque el desarrollo del aplicativo web se basa principalmente en el uso de herramientas libres. Esto permite reducir a gran escala los costos asociados a licencias. Los recursos tecnológicos necesarios, como los dispositivos básicos y acceso a internet, ya se encuentran disponibles, lo que hace aún más viable la implementación del sistema.

**Tabla 5***Viabilidad economica del sistema*

Recurso	Tipo de licencia
React.js	Software libre
Node.js	Software libre
Express.js	Software libre
PostgreSQL	Software libre
Sequelize	Software libre
GitHub	Software libre
Figma	Software libre
Navegador web	Software libre
Equipo de computo	Recurso existente

**Tabla 6***Costos de desarrollo*

Concepto	Descripción	Costo Estimado (USD)
Análisis y diseño del sistema	Levantamiento de información y diseño de la arquitectura	\$500
Desarrollo frontend	Implementación de interfaz web en React	\$550

Desarrollo backend	Implementación de API REST con Node.js y Express	\$550
Diseño e implementación de base de datos	Modelado y configuración en PostgreSQL	\$550
Pruebas y validaciones	Pruebas funcionales y verificación del sistema	\$550
Documentación técnica	Elaboración de manuales y documentación del proyecto	\$300
Costo total del desarrollo		\$3000

*Nota.* Los costos de desarrollo serán asumidos por los titulares del proyecto

#### 14.6.2 Tablas de Valores

Los costos del proyecto se limitan solo al tiempo de desarrollo, pruebas e implementación, sin requerir inversión en software propietario.

**Tabla 7**

*Tabla de valores*

Recurso	Tipo	Descripción
Servidor de la aplicación	Local	Ejecuta el backend desarrollado en node.js y express.js
Servidor de base de datos	Local	Almacena la información en PostgreSQL

Navegador web	Cliente	Permite el acceso al aplicativo
Red local / Internet	Servicio	Comunicación entre cliente y servicio
Sistema operativo	Software base	Entorno donde se ejecuta el sistema

### 14.7 Casos de Uso

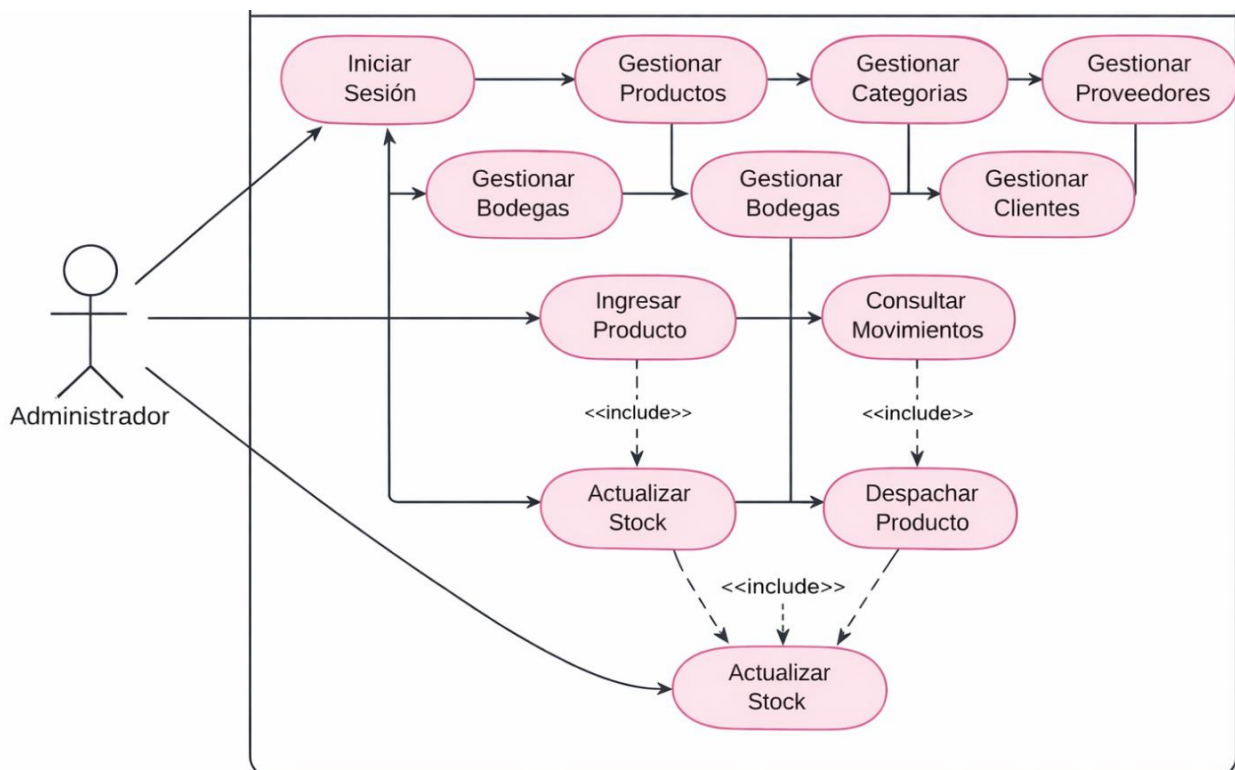
Un caso de uso explica cómo los usuarios interactúan con un producto o sistema.

Describe el flujo de entradas del usuario estableciendo caminos exitosos y fallidos. (Figma, 2025)

En el presente sistema, el aplicativo web es utilizado solo por un usuario con rol de Administrador, quien es responsable de todas las operaciones relacionadas con la gestión del inventario, clientes, proveedores, bodegas, ingresos y despachos.

**Figura 3**

*Diagrama de casos de uso*



**Tabla 8***Caso de uso: Iniciar Sesión*

Descripción	Iniciar sesión
Actor	Administrador
Objetivo	Permitir el acceso al sistema mediante autenticación segura
Precondición	El Administrador no debe haber iniciado sesión
Entrada	Correo electrónico y contraseña
Proceso	El sistema valida las credenciales y autoriza el acceso
Salida	Acceso al panel principal
Postcondición	Sesión activa del Administrador

**Tabla 9***Caso de uso: Visualizar productos*

Descripción	Visualizar productos
Actor	Administrador
Objetivo	Consultar el listado de productos registrados
Precondición	Sesión iniciada
Entrada	Solicitud de acceso al módulo productos
Proceso	El sistema muestra los productos activos con su información
Salida	Listado de productos

Postcondición	Información visualizada sin modificaciones
---------------	--

**Tabla 10**

*Caso de uso: Registrar producto*

Descripción	Registrar producto
Actor	Administrador
Objetivo	Registrar un nuevo producto en el inventario
Precondición	Sesión iniciada
Entrada	Datos del producto
Proceso	El sistema valida y guarda el producto
Salida	Producto registrado
Postcondición	Producto disponible en el sistema

**Tabla 11**

*Caso de uso: Editar producto*

Descripción	Editar producto
Actor	Administrador
Objetivo	Actualizar información de un producto
Precondición	Producto existente
Entrada	Nuevos datos del producto
Proceso	El sistema actualiza la información

Salida	Producto actualizado
Postcondición	Cambios reflejados en el sistema

**Tabla 12**

*Caso de uso: Consultar detalle de producto*

Descripción	Consultar detalle de producto
Actor	Administrador
Objetivo	Visualizar información detallada del producto
Precondición	Producto existente
Entrada	Selección de producto
Proceso	El sistema muestra el detalle completo
Salida	Información detallada
Postcondición	Consulta finalizada

**Tabla 13**

*Caso de uso: Desactivar producto*

Descripción	Desactivar producto
Actor	Administrador
Objetivo	Desactivar un producto sin eliminarlo
Precondición	Producto activo
Entrada	Confirmación de desactivación
Proceso	El sistema cambia el estado a inactivo

Salida	Producto desactivado
Postcondición	Producto no disponible para operaciones

**Tabla 14**

*Caso de uso: Gestionar categorías*

Descripción	Gestionar categorías
Actor	Administrador
Objetivo	Administrar categorías del sistema
Precondición	Sesión iniciada
Entrada	Datos de la categoría
Proceso	El sistema crea o actualiza la categoría
Salida	Categoría registrada
Postcondición	Catálogo actualizado

**Tabla 15**

*Caso de uso: Gestionar bodegas*

Descripción	Gestionar bodegas
Actor	Administrador
Objetivo	Administrar bodegas de almacenamiento
Precondición	Sesión iniciada
Entrada	Datos de la bodega

Proceso	El sistema guarda o actualiza la bodega
Salida	Bodega registrada
Postcondición	Bodegas disponibles para inventario

**Tabla 16**

*Caso de uso: Gestionar proveedores*

Descripción	Gestionar proveedores
Actor	Administrador
Objetivo	Administrar proveedores del sistema
Precondición	Sesión iniciada
Entrada	Datos del proveedor
Proceso	El sistema registra o actualiza proveedor
Salida	Proveedor registrado
Postcondición	Proveedor disponible

**Tabla 17**

*Caso de uso: Gestionar clientes*

Descripción	Gestionar clientes
Actor	Administrador
Objetivo	Administrar clientes para despachos
Precondición	Sesión iniciada
Entrada	Datos del cliente
Proceso	El sistema guarda la información

Salida	Cliente registrado
Postcondición	Cliente disponible

**Tabla 18**

*Caso de uso: Registrar ingreso de productos*

Descripción	Registrar ingreso de productos
Actor	Administrador
Objetivo	Registrar entradas de productos al inventario
Precondición	Producto y bodega existentes
Entrada	Datos de ingreso
Proceso	El sistema incrementa el stock
Salida	Ingreso registrado
Postcondición	Stock actualizado

**Tabla 19**

*Caso de uso: Registrar despacho de productos*

Descripción	Registrar despacho de productos
Actor	Administrador
Objetivo	Registrar salidas de productos
Precondición	Stock suficiente
Entrada	Datos del despacho
Proceso	El sistema reduce el stock

Salida	Despacho registrado
Postcondición	Stock actualizado

**Tabla 20**

*Caso de uso: Consultar movimientos*

Descripción	Consultar movimientos
Actor	Administrador
Objetivo	Visualizar historial de movimientos
Precondición	Sesión iniciada
Entrada	Filtros de búsqueda
Proceso	El sistema consulta registros
Salida	Historial mostrado
Postcondición	Consulta finalizada

**Tabla 21**

*Caso de uso: Cerrar sesión*

Descripción	Cerrar sesión
Actor	Administrador
Objetivo	Finalizar la sesión activa
Precondición	Sesión iniciada
Entrada	Solicitud de cierre

---

Proceso	El sistema invalida la sesión
Salida	Sesión cerrada
Postcondición	Acceso restringido

---

## Capítulo III

### 15.1. Desarrollo

Se realizó con una metodología ágil basada en Scrum, lo que permitió utilizar una construcción progresiva e incremental del aplicativo web para la gestión de inventarios de Casa Guflo. Este planteamiento permitió validar funcionalidades de una forma continua realizando ajustes.

El desarrollo inició con la configuración del entorno de trabajo, incluyendo las herramientas tanto para el frontend como para el backend. Posteriormente, se implementaron los módulos principales de la aplicación, priorizando aquellos relacionados con la gestión de productos, categorías, proveedores, clientes y bodegas, debido a su impacto directo en el control del inventario.

Durante esta etapa se desarrolló el frontend que se encarga de la interacción con el usuario mediante una interfaz web intuitiva y responsiva, mientras que el backend gestiona la lógica de negocio, la validación de datos.

Finalmente, el desarrollo incluyó la integración entre frontend y backend, asegurando que las operaciones de ingreso y despacho de productos actualicen el stock de manera automática y consistente. Cada funcionalidad fue testada de forma continua para garantizar su correcto funcionamiento previo al avance con la siguiente etapa.

### 15.2. Requerimientos de Software

El sistema responde a requerimientos definidos anteriormente en las fases de análisis y diseño. Estos requerimientos permitieron establecer el alcance funcional del aplicativo y garantizó que la solución resuelva las necesidades operativas del negocio.

Uno de los principales requerimientos del software se encuentra la capacidad de gestionar

la información relacionada con productos, categorías, proveedores, clientes y bodegas.

Asimismo, el sistema debe permitir el registro de ingresos y despachos de productos, asegurando la actualización automática del stock y el registro de todos los movimientos realizados.

Adicionalmente, el software debe garantizar un acceso seguro, permitiendo únicamente el acceso al sistema por parte del usuario administrador. La aplicación debe ser accesible desde un navegador web, sin la necesidad de instalaciones adicionales, y ofrecer una interfaz clara que facilite su uso en las actividades diarias del negocio.

### 15.3. Cronograma

El cronograma del proyecto se lo realizó a partir de la planificación de sprints. Cada sprint tuvo una duración de tres semanas incluyendo objetivos claros y específicos.

Esta planificación distribuyó las actividades de desarrollo, pruebas e integración de forma ordenada, facilitando el seguimiento del proyecto y el cumplimiento de los tiempos establecidos.

**Tabla 22**

*Cronograma de trabajo*

Sprint	Duración	Objetivo	Actividades principales	Entregables	Responsable
Sprint 1	Semanas 1 a 3	Establecer las bases del proyecto y definir los requerimientos	Configuración del entorno de desarrollo, creación del repositorio en GitHub, definición de requerimientos y diseño preliminar de la arquitectura	Entorno configurado, estructura base del proyecto, diseño inicial de arquitectura	Wilson Mites Brandom Ortega
Sprint 2	Semanas 4 a 6	Diseñar la experiencia de usuario del sistema	Creación de prototipos en Figma, diseño de vistas principales,	Diseño de navegación y experiencia de usuario	Brandom Ortega

---

			definición de navegación y validación con usuarios		
Sprint 3	Semanas 7 a 9	Implementar la interfaz principal del sistema	Desarrollo de vistas en React, y autenticación visual, creación de componentes reutilizables	Interfaz funcional del sistema, navegación operativa	Brandom Ortega
Sprint 4	Semanas 10 a 12	Desarrollar los servicios del backend	Implementación de API REST, configuración de PostgreSQL, desarrollo de endpoints principales	API funcional, base de datos operativa, servicios conectados	Wilson Mites
Sprint 5	Semanas 13 a 15	Integrar y validar el sistema completo	Integración frontend-backend, pruebas funcionales y retroalimentación del usuario	Sistema integrado, resultados de pruebas	Wilson Mites
Sprint 6	Semanas 16 a 18	Finalizar el proyecto y preparar la entrega	Correcciones finales, documentación técnica y manual de usuario	Sistema finalizado, documentación completa, entrega académica	Wilson Mites

---

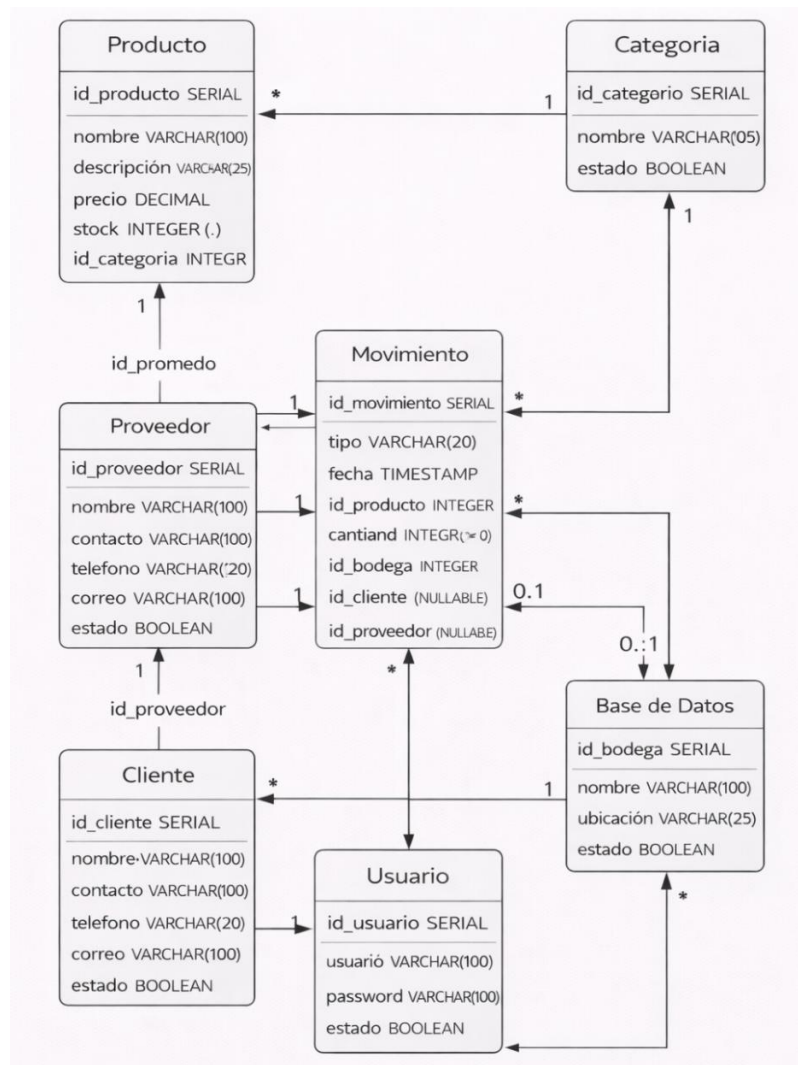
## 15.4. Diagrama de Clases

Este diagrama presenta la estructura lógica del sistema y muestra las principales entidades que conforman el modelo del aplicativo, sus atributos y relaciones.

También permite visualizar la organización de los datos y la interacción entre las diferentes áreas, como productos, categorías, proveedores, clientes, bodegas, movimientos e inventario. Su elaboración facilitó la correcta implementación del modelo de datos y ayudó a mantener la coherencia entre la base de datos y la lógica del negocio.

**Figura 4**

*Diagrama de clases*

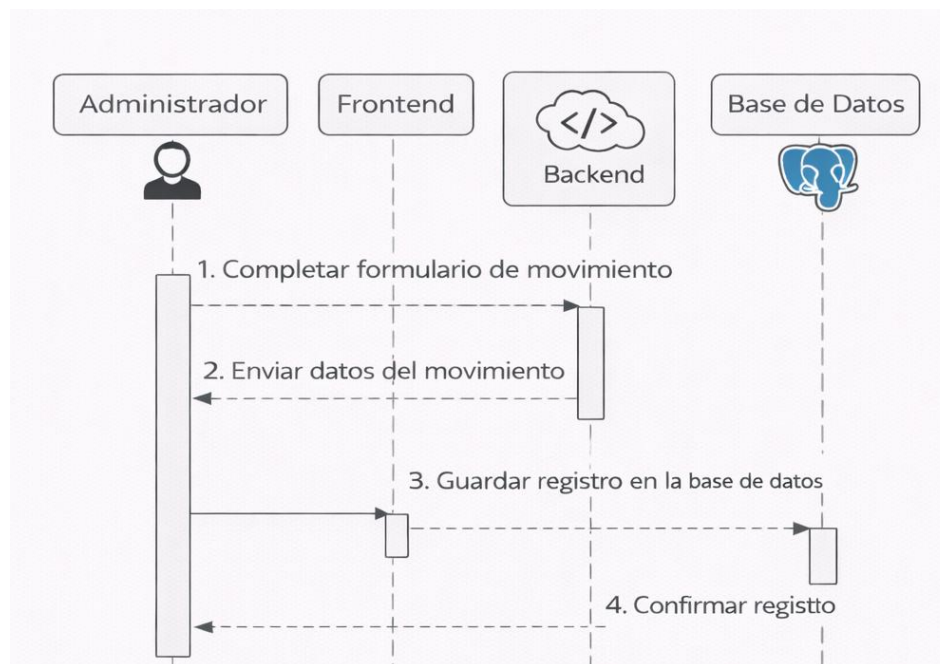


## 15.5. Diagrama de Secuencia

El siguiente diagrama indica el flujo de interacción entre el usuario administrador y los componentes del sistema durante la ejecución de procesos, como el registro de ingresos o despachos de productos.

Además, ayuda a identificar el orden de las operaciones, las solicitudes realizadas desde la interfaz, la respuesta del backend y las actualizaciones en la base de datos. Su uso ayudó a comprender el comportamiento del sistema y a validar de forma correcta la comunicación entre los módulos.

**Figura 5**  
*Diagrama de secuencia*



## 15.6. Código Fuente

La estructura del código se organizó de manera modular, lo que facilita su mantenimiento y futura escalabilidad.

En esta sección se presentan capturas de fragmentos representativos del código, acompañadas de una breve explicación de su función. A su vez, se incluyen imágenes de la

interfaz del aplicativo para evidenciar el resultado final del desarrollo.

## Figura 6

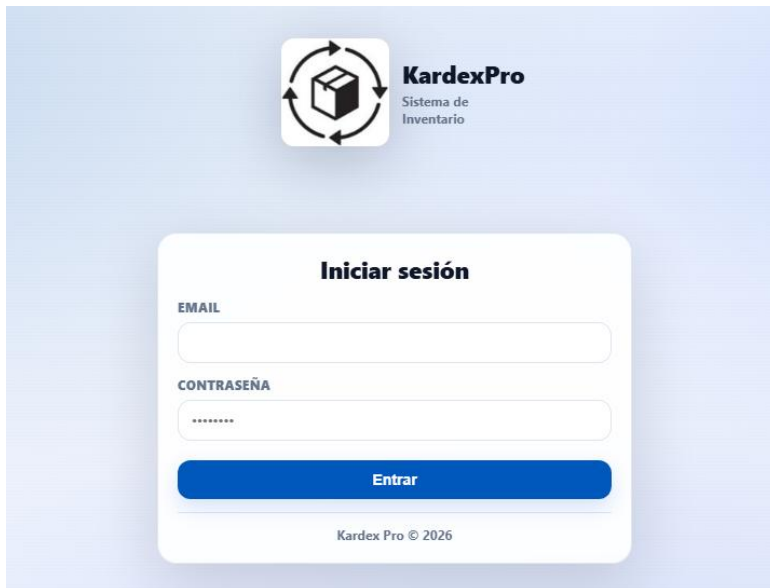
*Función para redirigir al login*

```
10
11 export const useAuthStore = create<AuthState>((set) => {
12   // Leer valores guardados (si existen)
13   const savedToken = sessionStorage.getItem('token');
14   const savedUser = sessionStorage.getItem('user');
15
16   return {
17     token: savedToken || null,
18     user: savedUser ? (JSON.parse(savedUser) as User) : null,
19
20     login: (token, user) => {
21       sessionStorage.setItem('token', token);
22       sessionStorage.setItem('user', JSON.stringify(user));
23       set({ token, user });
24     },
25
26     logout: () => {
27       sessionStorage.removeItem('token');
28       sessionStorage.removeItem('user');
29       set({ token: null, user: null });
30     },
31   };
32 });
```

*Nota:* Código que implica mediante un sessionStorage el regreso a la pestaña de login por más que se cierre la pestaña de la página web aun estando iniciado sesión.

**Figura 7**

*Interfaz del LogIn*



**Figura 8**

*Lógica despacho tipo caja*

```
{detalles.map((d, i) => {
  const prod = productos.find((p) => p.id === d.producto_id);
  const upc = prod?.unidades_por_caja || 0;

  const opcionesProducto = (() => {
    const base = [...productosFiltradosBase];
    const seleccionado = productos.find((p) => p.id === d.producto_id);
    if (seleccionado && !base.some((p) => p.id === seleccionado.id)) {
      return [seleccionado, ...base];
    }
    return base;
  })();

  const totalLinea = getLineaTotal(d);

  let cajasEq = 0;
  let restoEq = 0;
  if (d.precio_tipo === 'CAJA' && upc > 0 && d.cantidad > 0) {
    cajasEq = Math.floor(d.cantidad / upc);
    restoEq = d.cantidad % upc;
  }
}
```

Nota: Esta parte del código hace la conversión de cajas a unidades, esto se hace por el cómo manejan la venta de tipo caja en el negocio.

## Figura 9

### Interfaz de despacho tipo caja

**Línea #1 - 4432 - Canillera**

<b>BODEGA</b>	2do Piso	<b>PRODUCTO</b>	4432 - Canillera
<b>TIPO DE PRECIO</b>	Caja	<b>CANTIDAD (UNIDADES)</b>	215
<b>PRECIO UNITARIO (CAJA)</b>	1,25	1 caja = 200 unidades Equivale a 1 caja(s) + 15 unidad(es)	
<b>MOTIVO DESCUENTO (OPCIONAL)</b>	Solo si usas DESCUENTO		

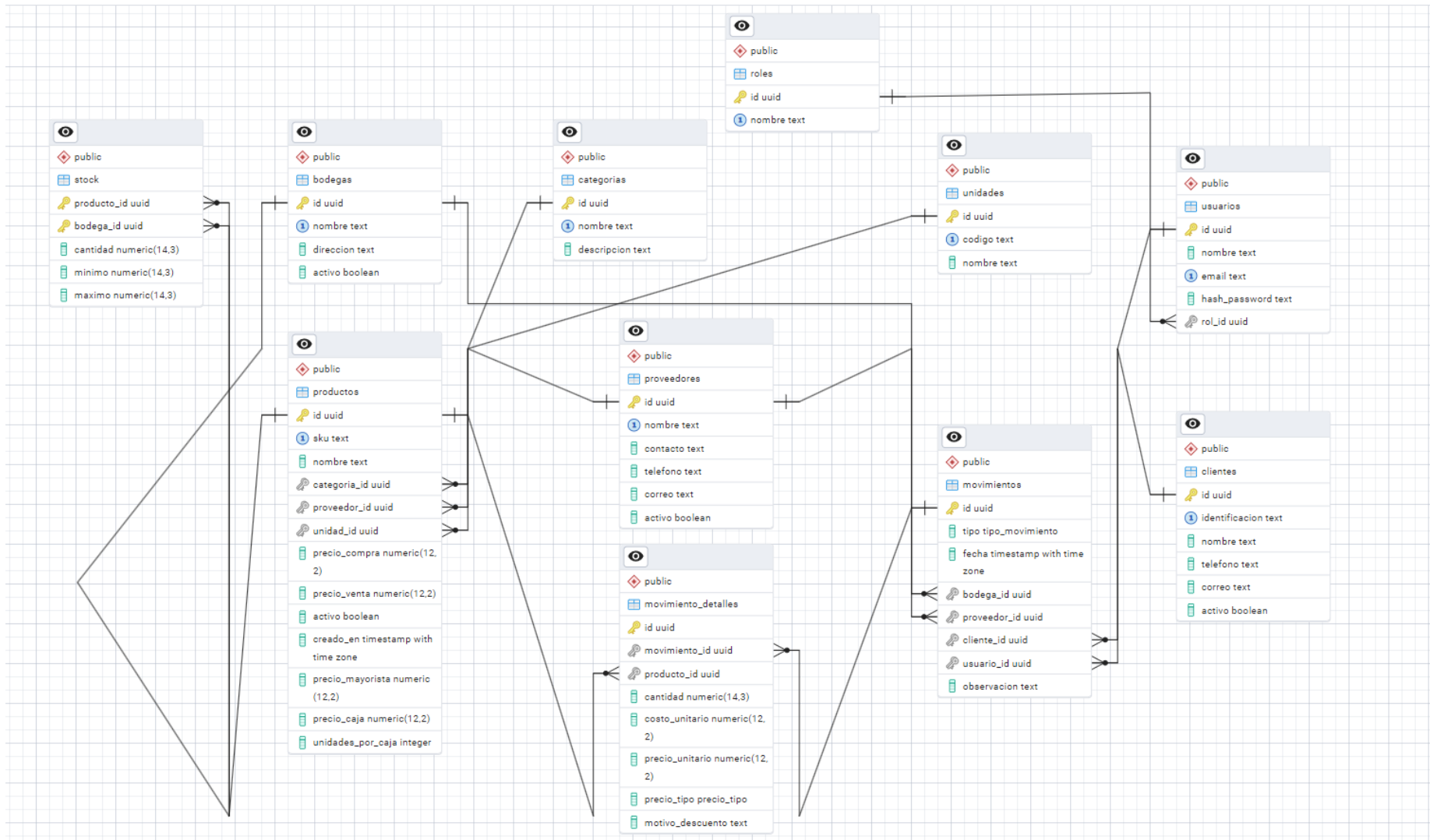
## **15.7. Modelo Base de Datos**

Se utiliza una base de datos relacional diseñada para almacenar la información relacionada con el inventario y los movimientos de productos. También, garantiza la integridad y consistencia de la información mediante el uso de claves primarias, claves foráneas y relaciones entre las tablas.

### **15.7.2 MER**

Aquí se representa gráficamente las entidades principales del sistema y sus relaciones. Este modelo ayudó a definir la estructura de la base de datos, asegurando una correcta organización de la información.

**Figura 10**  
**MER**



### 15.7.3 Diccionario de Datos

Describe cada una de las tablas del sistema, con sus campos respectivos, tipos de datos y restricciones. Esta documentación nos facilita la comprensión de la estructura de la base de datos y una futura evolución del sistema.

**Tabla 23**

*Tabla roles*

Atributo	Tipo	Descripción	PK	FK
id	uuid	Identificador único del rol	Si	
nombre	text	Nombre del rol (ej. ADMIN)		

**Tabla 24**

*Tabla usuarios*

Atributo	Tipo	Descripción	PK	FK
id	uuid	Identificador único del usuario	Si	
nombre	text	Nombre del usuario		
email	text	Correo del usuario		
hash_password	text	Contraseña cifrada (hash)		
rol_id	uuid	Rol asignado al usuario		Si roles.id

**Tabla 25**

*Tabla unidades*

Atributo	Tipo	Descripción	PK	FK
id	uuid	Identificador único de la unidad	Si	
codigo	text	Código corto de unidad (ej. UND, CAJ)		
nombre	text	Nombre de la unidad		

**Tabla 26***Tabla categorías*

<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>	<b>PK</b>	<b>FK</b>
id	uuid	Identificador único de la categoría	Si	
nombre	text	Nombre de la categoría		
descripcion	text	Descripción de la categoría		

**Tabla 27***Tabla proveedores*

<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>	<b>PK</b>	<b>FK</b>
id	uuid	Identificador único del proveedor	Si	
nombre	text	Nombre del proveedor		
contacto	text	Nombre/Persona de contacto		
telefono	text	Teléfono del proveedor		
correo	text	Correo del proveedor		
activo	boolean	Estado lógico del proveedor		

**Tabla 28***Tabla bodegas*

<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>	<b>PK</b>	<b>FK</b>
id	uuid	Identificador único de la bodega	Si	
nombre	text	Nombre de la bodega		
direccion	text	Dirección de la bodega		
activo	boolean	Estado lógico de la bodega		

**Tabla 29***Tabla clientes*

<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>	<b>PK</b>	<b>FK</b>
id	uuid	Identificador único del cliente	Si	

identificacion	text	Cédula/RUC/identificación
nombre	text	Nombre del cliente
telefono	text	Teléfono del cliente
correo	text	Correo del cliente
activo	boolean	Estado lógico del cliente

**Tabla 30**

*Tabla productos*

Atributo	Tipo	Descripción	PK	FK
id	uuid	Identificador único del producto	Si	
sku	text	Código SKU del producto		
nombre	text	Nombre del producto		
categoria_id	uuid	Categoría asociada		Si categorias.id
proveedor_id	uuid	Proveedor asociado		Si proveedores.id
unidad_id	uuid	Unidad de medida		Si unidades.id
precio_compra	numeric(12,2)	Precio de compra		
precio_venta	numeric(12,2)	Precio de venta		
activo	boolean	Estado lógico del producto		
creado_en	timestamp with time zone	Fecha/hora de creación		
precio_mayorista	numeric(12,2)	Precio al por mayor		
precio_caja	numeric(12,2)	Precio por caja		
unidades_por_caja	integer	Cantidad de unidades por caja		

**Tabla 31***Tabla stock*

Atributo	Tipo	Descripción	PK	FK
producto_id	uuid	Producto relacionado al stock		Si productos.id
bodega_id	uuid	Bodega relacionada al stock		Si bodegas.id
cantidad	numeric(14,3)	Cantidad disponible en bodega		
minimo	numeric(14,3)	Stock mínimo permitido		
maximo	numeric(14,3)	Stock máximo permitido		

**Tabla 32***Tabla movimientos*

Atributo	Tipo	Descripción	PK	FK
id	uuid	Identificador único del movimiento	Si	
tipo_tipo_movimiento	(enum)	Tipo de movimiento (Ingreso/Despacho/Ajuste)		
fecha	timestamp with time zone	Fecha/hora del movimiento		
bodega_id	uuid	Bodega asociada		Si bodegas.id
proveedor_id	uuid	Proveedor (si aplica)		Si proveedores.id
cliente_id	uuid	Cliente (si aplica)		Si clientes.id
usuario_id	uuid	Usuario que registra el movimiento		Si usuarios.id
observacion	text	Observación o nota del movimiento		

**Tabla 33***Tabla movimiento detalle*

Atributo	Tipo	Descripción	PK	FK
id	uuid	Identificador único del detalle	Si	
movimiento_id	uuid	Movimiento padre		Si movimientos.id
producto_id	uuid	Producto involucrado		Si productos.id
cantidad	numeric(14,3)	Cantidad del producto en el movimiento		
costo_unitario	numeric(12,2)	Costo unitario (si aplica)		
precio_unitario	numeric(12,2)	Precio unitario (si aplica)		
precio_tipo	(enum)	Tipo de precio usado (si aplica)		
motivo_descuento	text	Motivo/nota de descuento (si aplica)		

## Capítulo IV

### 16.1. Pruebas y resultado

Estas pruebas ayudaron a verificar el funcionamiento correcto de cada uno de los módulos y a validar que las funcionalidades desarrolladas cumplan con los requerimientos establecidos y respondan de forma correcta ante distintos escenarios de uso.

Se ejecutaron pruebas sobre los procesos de autenticación, gestión de productos, control de inventario, ingresos y despachos de productos, y en la visualización de movimientos. Los resultados evidenciaron un comportamiento estable del sistema y actualización correcta de la base de datos.

## 16.2. Tablas de iteración de cada prueba con sus versiones

Para documentar el proceso de validación, se desarrollaron tablas de iteración en cada versión del software. Estas tablas permiten identificar las funcionalidades evaluadas, resultados y observaciones.

El uso de estas tablas facilitó el seguimiento, arreglo de errores y mejoras durante el desarrollo, asegurando una evolución controlada del sistema hasta su versión final.

### Tabla 34

#### *Iteración 1 Ingreso de un producto*

Sección	Iteración	Prerrequisito	Entidad	Proceso	Proceso secundario	Acción	Resultado	Comentarios
Gestión de Ingresos	1	El sistema debe permitir registrar un ingreso y actualizar el stock.	Administrador	Registrar ingreso	Insertar movimiento en BD	Registrar producto + cantidad + bodega	50%	El sistema registra el movimiento, pero no actualiza correctamente el stock por bodega (queda igual o se duplica).

**Tabla 35***Iteración 2 Ingreso de un producto*

Sección	Iteración	Prerrequisito	Entidad	Proceso	Proceso secundario	Acción	Resultado	Comentarios
Gestión de Ingresos	2	El sistema debe registrar ingresos y reflejar el stock en tiempo real.	Administrador	Registrar ingreso	Actualizar stock en BD	Registrar producto + cantidad + bodega	100%	Se valida la entrada y el sistema actualiza el stock por bodega y el stock total correctamente.

**Tabla 36***Iteración 1 Despacho*

Sección	Iteración	Prerrequisito	Entidad	Proceso	Proceso secundario	Acción	Resultado	Comentarios
Gestión de Despachos	1	El sistema debe impedir despachos si no hay stock suficiente.	Administrador	Registrar despacho	Validar existencia en BD	Registrar cliente + bodega + producto + cantidad	50%	El sistema permite el despacho, aunque no exista stock suficiente, generando stock negativo o inconsistencia.

**Tabla 37***Iteración 2 Despacho*

Sección	Iteración	Prerrequisito	Entidad	Proceso	Proceso secundario	Acción	Resultado	Comentarios
Gestión de Despachos	2	El sistema debe validar stock antes del despacho.	Administrador	Registrar despacho	Consultar y validar stock en BD	Registrar cliente + bodega + producto + cantidad	100%	El sistema bloquea el despacho si la cantidad supera el stock y muestra mensaje de error; con stock válido, descuenta correctamente.

**Tabla 38***Iteración 1 Inicio de sesión*

Sección	Iteración	Prerrequisito	Entidad	Proceso	Proceso secundario	Acción	Resultado	Comentarios
Autenticación	1	El sistema debe permitir iniciar sesión al Administrador.	Administrador	Iniciar sesión	Validar credenciales en BD	Ingresar usuario y contraseña	50%	La autenticación responde, pero no mantiene sesión.

**Tabla 39***Iteración 2 Inicio de sesión*

Sección	Iteración	Prerrequisito	Entidad	Proceso	Proceso secundario	Acción	Resultado	Comentarios
Autenticación	2	El sistema debe permitir sesión estable para acceder a módulos.	Administrador	Iniciar sesión	Generar y persistir token	Ingresar usuario y contraseña	100%	El sistema genera token, lo guarda en sesión/local storage y lo envía en cada request; acceso correcto a dashboard y módulos.

## **Conclusiones**

El desarrollo del aplicativo web demostró que la automatización de procesos generó una solución eficaz para mejorar el control, organización y trazabilidad de la información relacionada con productos, bodegas, ingresos y despachos. La implementación del sistema se acopla perfectamente a la solución de las necesidades de la empresa, reduciendo el uso de procesos manuales y minimizando errores.

La aplicación de la metodología ágil Scrum fue fundamental para la planificación y ejecución del proyecto, logrando ajustes progresivos conforme se avanzaba. Así se logró un seguimiento del progreso, identificación de mejoras y entrega de funcionalidades a tiempo.

Para finalizar, se concluye que el aplicativo web cumple con los objetivos planteados al inicio del proyecto, ofreciendo una herramienta tecnológica eficiente, segura y accesible que aporta valor al negocio de Casa Guflo.

## **Recomendaciones**

Se recomienda que para futuros proyectos de software similares se considere desde las etapas iniciales una correcta planificación de la arquitectura del sistema. Esto no solo permite reducir costos de implementación, también permite facilitar el mantenimiento, la evolución del sistema y la integración de nuevas funcionalidades conforme crecen las necesidades del negocio.

También se debe aplicar metodologías ágiles durante el proceso de desarrollo, como Scrum, ya que ayudan a tener mejor organización del trabajo, comunicación constante entre los integrantes del equipo y detección temprana de errores.

También se recomienda implementar mecanismos de validación y control de errores desde las primeras fases del desarrollo, especialmente en módulos críticos ya que estos se conectan de forma directa al proyecto que se esté desarrollando.

De igual manera recomiendo a la carrera a que nos oriente más en el tema de la ciberseguridad de aplicativos y aplicaciones, como por ejemplo roles de usuario, ya que para poder hacer un aplicativo mucho más robusto se necesita tener una seguridad muy bien hecha y estructurada para poder evitar pérdida de datos claves y sensibles a futuro.

## Referencias bibliográficas

- Schwaber, K., & Sutherland, J. (2020). The Scrum Guide. Scrum.org.
- Fetecua, A. (2025, 1 diciembre). ¿Qué es un mapa de navegación web? ¡Lo que debes saber! DesignPlus.
- Qué es un caso de uso: Escritura, ejemplo, plantilla | Figma. (s. f.). Figma.
- Alaimo Labs. (2024). Roles y responsabilidades del equipo Scrum | Alaimo Labs.
- Laoyan, S. (2026, 6 enero). Conoce a fondo las 4 ceremonias ágiles. Asana