

EXPEUEW7

Especificación de requisitos de software

Proyecto: [“Perfu” by Perfulandia SPA]

Revisión: [99.99]

[Martes 08 de Abril del 2025]

Perfume.

Contenido

Ficha del documento	3
1. Introducción	4
1.1. Propósito	4
1.2. Ámbito del Sistema	4
1.3. Definiciones, Acrónimos y Abreviaturas	5
1.4. Referencias	5
1.5. Visión General del Documento	6
2. Descripción General	7
2.1. Perspectiva del Producto	7
2.2. Funciones del Producto	7
2.2. Historias de Usuarios	8
3. Diseño de la Nueva Arquitectura	10
3.1. Casos De Uso	10
3.2. Diagrama de Clases	14
3.3. Diagrama de Microservicios	15
4. Plan de Migración	16
4.1. Objetivo 9	16
4.2. Alcance 9	16

4.3.	Fase 9	17
4.4.	Estrategia 9	18
4.5.	Planificación Detallada	18
4.6.	Gestión de Riesgo	18
4.7.	Criterios de Aceptación	19
4.8.	Comunicación	19
5.	Consideraciones Éticas	20
5.1.	Privacidad y Seguridad	20
5.2.	Inclusión y Accesibilidad	20
5.3.	Transparencia y Responsabilidad	21
5.4.	Equidad y No Discriminacion	21
5.5.	Impacto Social y Ambiental	21

Ficha del documento

Fecha	Revisión	Autor	Modificación

Documento validado por las partes en fecha:

Por el cliente

Por la empresa suministradora

[Firma]

[Firma]

Sr./Sra.

Sr./Sra.

1. Introducción

Perfulandia SPA es una firma chilena nueva que ha crecido rápido por su gama de productos de buena calidad a precios bajos. Hoy tiene tiendas en Santiago, Concepción y Viña del Mar. Su crecimiento ha mostrado las fallas del sistema de software grande actual, que tiene problemas con rendimiento, escalabilidad y disponibilidad;

El actual documento ERS tiene el fin de definir los requisitos del nuevo sistema que se basa en una arquitectura de microservicios, esto llevará al cambio del sistema vigente, en línea con la estrategia de digitalización y aumento de tamaño de la empresa.

1.1. Propósito

El propósito de este documento de Especificación de Requisitos del Sistema es definir de una forma detallada y estructurada los requisitos funcionales y no funcionales del nuevo sistema “Perfu”, el cual busca modernizar y optimizar los procesos operativos de Perfulandia SPA a través de una arquitectura basada en microservicios. Además, este documento se utilizará como guía principal para el equipo de desarrollo, con el fin de garantizar que todas las partes involucradas comprendan perfectamente las funcionalidades esperadas, los objetivos del sistema y las condiciones en las que debe operar.

Está dirigido a las siguientes audiencias:

- **Equipo de desarrollo:** para implementar correctamente los módulos requeridos.
- **Cientes internos (usuarios del sistema):** para validar que las funcionalidades correspondan a sus necesidades.
- **Jefatura de proyecto y stakeholders:** para seguimiento del alcance y validación del cumplimiento de los objetivos del sistema.
- **Equipo de QA:** para definir criterios de prueba basados en los requisitos.

1.2. Ámbito del Sistema

Este nuevo sistema llamado “perfu”, estará basado en una arquitectura de microservicios. Esta elección permitirá que el sistema sea escalable, modular y fácilmente mantenible, facilitando futuras mejoras, integración con nuevos servicios y adaptaciones.

“Perfu” podrá gestionar usuarios, inventario, ventas, pedidos, reportes y atención al cliente, todo mediante servicios distribuidos que se comunicarán entre sí. Además tendrá una mayor disponibilidad del sistema, reducción de tiempos de respuesta, facilidad de mantenimiento y se espera sustituir el sistema monolítico actual, garantizar tiempos de respuesta menores a 2 segundos en operaciones críticas, y reducir los errores e incidencias técnicas reportadas en al menos un 80%.

1.3. Definiciones, Acrónimos y Abreviaturas

En esta subsección se definirán todos los términos, acrónimos y abreviaturas utilizadas en la ERS.

- **API:** Application Programming Interface
- **ERS:** *Especificación de Requisitos del Sistema* – Documento que detalla los requisitos funcionales y no funcionales de un sistema.
- **SPA:** *Sociedad por Acciones* – Tipo de entidad legal en Chile, usada comúnmente por empresas.
- **CRUD:** *Create, Read, Update, Delete* – Operaciones básicas de almacenamiento de datos.
- **UI:** *User Interface* – Interfaz de usuario.
- **UX:** *User Experience* – Experiencia del usuario.
- **MS:** *Microservicio* – Servicio independiente dentro de una arquitectura modular.
- **DB:** *Database* o base de datos.
- **Docker:** Plataforma para contenerización de aplicaciones.
- **Spring Boot:** Framework de desarrollo en Java para crear microservicios y aplicaciones web.
- **MySQL:** Sistema de gestión de bases de datos relacional.
- **GitHub:** Plataforma para control de versiones y colaboración en el desarrollo de software.

1.4. Referencias

Caso de Estudio - Perfulandia SPA: Documento base entregado por la asignatura para el desarrollo del proyecto.

Ley N° 19.628 sobre Protección de la Vida Privada: Legislación chilena referente al tratamiento de datos personales.

Documentación Oficial de Spring Boot: <https://spring.io/projects/spring-boot>

Documentación Oficial de Docker: <https://docs.docker.com/>

Guía para la Elaboración de Documentos ERS: – IEEE 830-1998

Documentación de MySQL: <https://dev.mysql.com/doc/>

Documentacion de Spring Data JPA: <https://spring.io/projects/spring-data-jpa>

1.5. Visión General del Documento

Este documento de Especificación de Requisitos del Sistema (ERS) está organizado en secciones que permiten comprender de forma clara el funcionamiento esperado del nuevo sistema “Perfu”. En primer lugar, se presenta una introducción con el contexto general, alcance, objetivos y beneficios del sistema.

Posteriormente, en la sección de **Análisis de Requerimientos**, se detallan los requerimientos funcionales y no funcionales según los distintos perfiles de usuarios. A continuación, se analiza el sistema actual para entender sus limitaciones y justificar la necesidad de cambio.

En la sección de **Diseño de la Nueva Arquitectura**, se describe la propuesta técnica basada en microservicios, incluyendo los servicios identificados y la planificación para su implementación. También se contempla un apartado para **diagramas técnicos** que serán incluidos en la versión final.

La sección de **Planificación de la Migración** expone la estrategia a seguir para llevar a cabo la transición del sistema monolítico al nuevo entorno distribuido.

Finalmente, se presentan **consideraciones éticas** relacionadas con el desarrollo del sistema y se concluye resumiendo los beneficios esperados y próximos pasos del proyecto.

2. Descripción General

Esta sección presenta una visión contextual del sistema **Perfu**, abordando los factores que influyen en sus requisitos, pero sin describirlos directamente. Se establece el entorno en el que se desarrollará el sistema, su relación con otros sistemas, los perfiles de usuarios que lo utilizarán, las restricciones existentes y supuestos que se consideran válidos al momento del desarrollo.

2.1. Perspectiva del Producto

Perfu es un sistema de software empresarial desarrollado bajo una arquitectura de microservicios, cuyo propósito es reemplazar el sistema monolítico actual utilizado por Perfulandia SPA. El sistema estará conformado por múltiples servicios distribuidos, cada uno encargado de funcionalidades específicas como la gestión de usuarios, ventas, pedidos, inventario y reportes.

El sistema interactúa con una base de datos MySQL, APIs internas entre microservicios y herramientas externas para la contenerización y despliegue (Docker, Spring Boot, etc.). El sistema será accesible a través de una interfaz web compatible con navegadores modernos.

2.2. Funciones del Producto

El sistema **Perfu** permitirá realizar las siguientes funciones generales:

- Registrar y autenticar usuarios según sus roles.
- Gestionar el inventario y productos de cada sucursal.
- Registrar ventas y generar facturas electrónicas.
- Controlar el estado de pedidos y envíos.
- Generar reportes de rendimiento, ventas e inventario.
- Permitir a los clientes consultar, comprar y realizar seguimiento a sus pedidos.

2.3. Historia de los Usuarios

Épica 1: Gestión de Usuarios

- **Historia de usuario 1:** Como Administrador del Sistema, quiero gestionar las cuentas de usuario (crear, actualizar, desactivar, eliminar) para mantener el control del acceso al sistema.
- **Historia de usuario 2:** Como Administrador del Sistema, quiero configurar los permisos de acceso a los diferentes módulos y funciones del sistema para asegurar la seguridad y el control de las acciones de los usuarios.
- **Historia de usuario 3:** Como Administrador del Sistema, quiero monitorizar el estado del sistema y recibir alertas sobre posibles fallos para asegurar el rendimiento y la disponibilidad del sistema.
- **Historia de usuario 4:** Como Administrador del Sistema, quiero respaldar y restaurar los datos del sistema para prevenir la pérdida de información.

Épica 2: Gestión de Sucursal

- **Historia de usuario 5:** Como Gerente de Sucursal, quiero gestionar el inventario (agregar, actualizar, eliminar productos, ajustar stock) para mantener el control de los productos disponibles.
- **Historia de usuario 6:** Como Gerente de Sucursal, quiero generar reportes de ventas, inventario y rendimiento de la sucursal para analizar el desempeño del negocio.
- **Historia de usuario 7:** Como Gerente de Sucursal, quiero gestionar los detalles específicos de la sucursal (horarios, personal, políticas) para configurar la operación de la sucursal.
- **Historia de usuario 8:** Como Gerente de Sucursal, quiero gestionar los pedidos de productos para reabastecer el inventario de la sucursal.

Épica 3: Gestión de Ventas

- **Historia de usuario 9:** Como Empleado de Ventas, quiero registrar las ventas (procesar transacciones, aplicar descuentos/ofertas) para gestionar las ventas de la sucursal.
- **Historia de usuario 10:** Como Empleado de Ventas, quiero atender las devoluciones y reclamaciones de los clientes para gestionar la satisfacción del cliente.
- **Historia de usuario 11:** Como Empleado de Ventas, quiero consultar el inventario (verificar disponibilidad, buscar productos) para proporcionar información precisa a los clientes.
- **Historia de usuario 12:** Como Empleado de Ventas, quiero generar facturas electrónicas y enviarlas por correo electrónico a los clientes para gestionar las transacciones de venta.

Épica 4: Gestión de Logística

- **Historia de usuario 13:** Como Logística, quiero gestionar los envíos de productos (crear, actualizar, seguir envíos) desde la bodega hasta las sucursales y los clientes para asegurar la entrega de los productos.
- **Historia de usuario 14:** Como Logística, quiero optimizar las rutas de entrega para los pedidos locales y regionales para mejorar la eficiencia de las entregas.
- **Historia de usuario 15:** Como Logística, quiero actualizar el estado de los pedidos en el sistema (desde el procesamiento hasta la entrega final) para proporcionar información precisa a los clientes.
- **Historia de usuario 16:** Como Logística, quiero gestionar los proveedores (mantener información, realizar pedidos, gestionar la recepción de mercancías) para asegurar el suministro de productos.

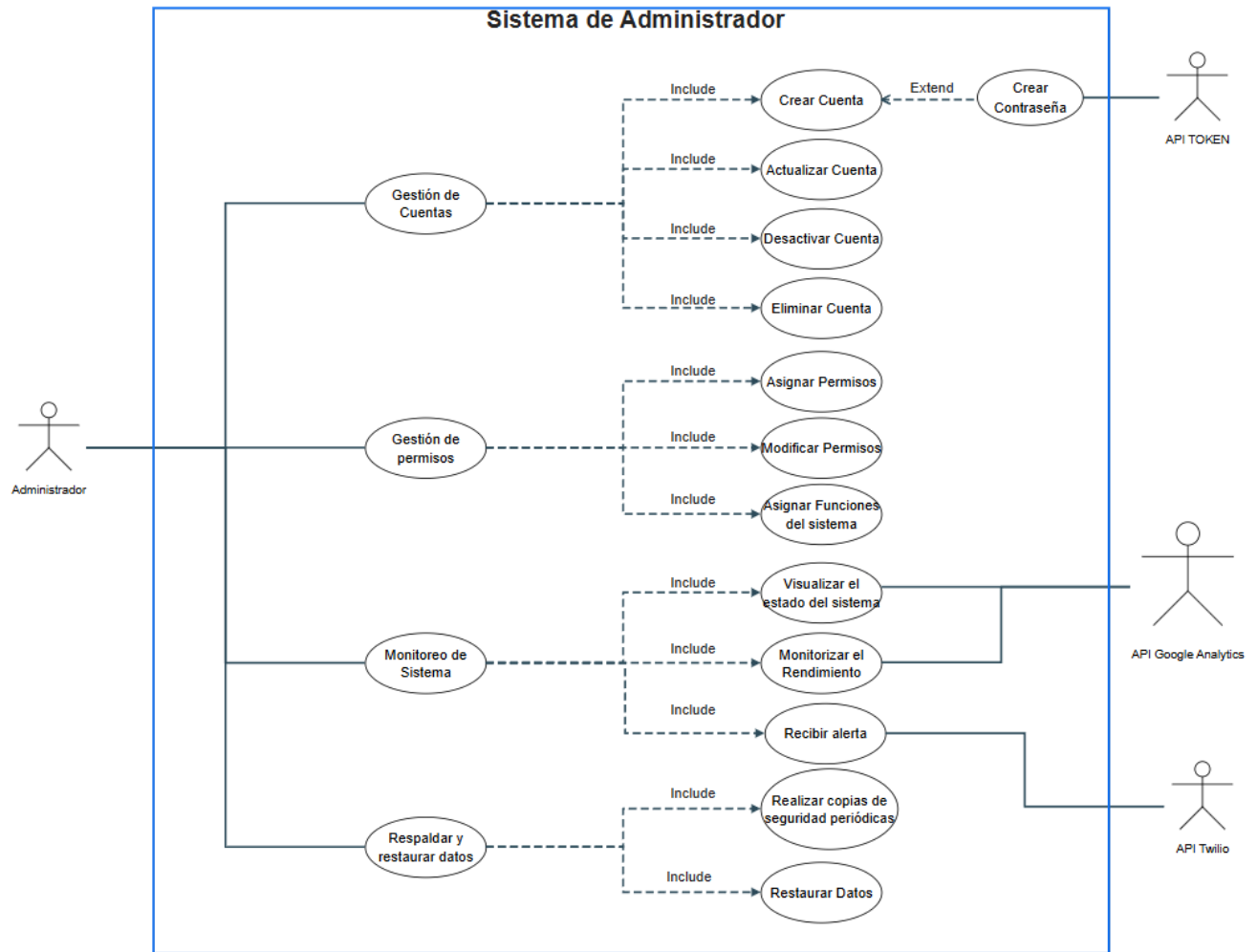
Épica 5: Acciones de los Clientes vía Web

- **Historia de usuario 17:** Como Cliente, quiero crear una cuenta en la plataforma web para registrarme y acceder a los servicios.
- **Historia de usuario 18:** Como Cliente, quiero iniciar sesión en mi cuenta para acceder a mis datos y realizar compras.
- **Historia de usuario 19:** Como Cliente, quiero navegar y buscar productos en el catálogo para encontrar los productos que necesito.
- **Historia de usuario 20:** Como Cliente, quiero agregar productos al carrito de compras para realizar una compra.
- **Historia de usuario 21:** Como Cliente, quiero realizar pedidos, proporcionando detalles de pago y seleccionando opciones de envío, para completar mi compra.
- **Historia de usuario 22:** Como Cliente, quiero consultar mi historial de pedidos y el estado actual de mis pedidos para realizar un seguimiento de mis compras.
- **Historia de usuario 23:** Como Cliente, quiero gestionar mi perfil (actualizar información personal, direcciones de envío, detalles de pago) para mantener mi información actualizada.
- **Historia de usuario 24:** Como Cliente, quiero solicitar soporte a través de un formulario de contacto o chat en línea para resolver mis dudas o problemas.
- **Historia de usuario 25:** Como Cliente, quiero dejar reseñas y calificaciones de los productos comprados para compartir mi opinión con otros usuarios.
- **Historia de usuario 26:** Como Cliente, quiero aplicar cupones y descuentos durante el proceso de compra para obtener beneficios.

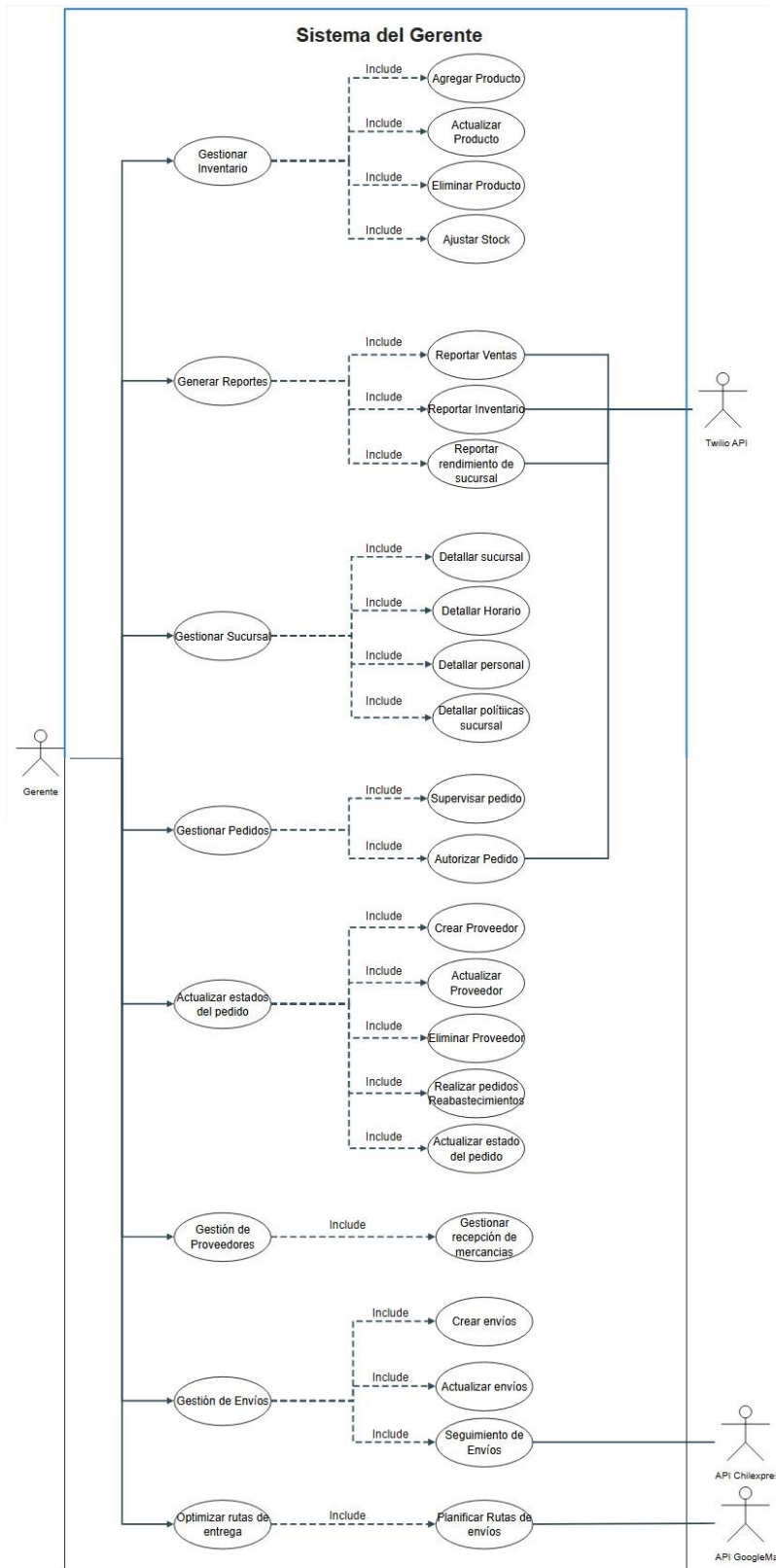
3. Diseño de la nueva arquitectura

3.1 Casos de Uso

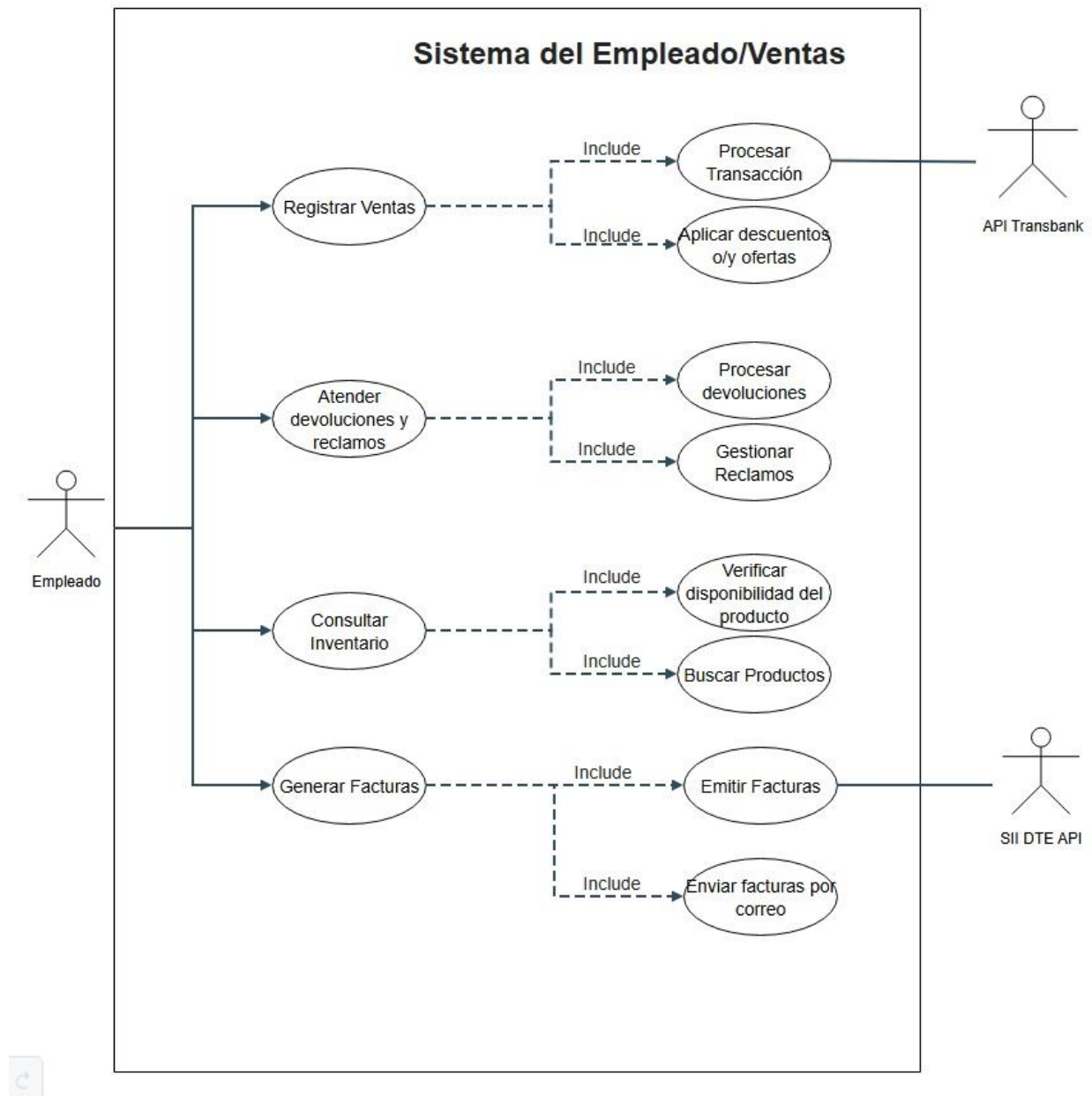
Administrador



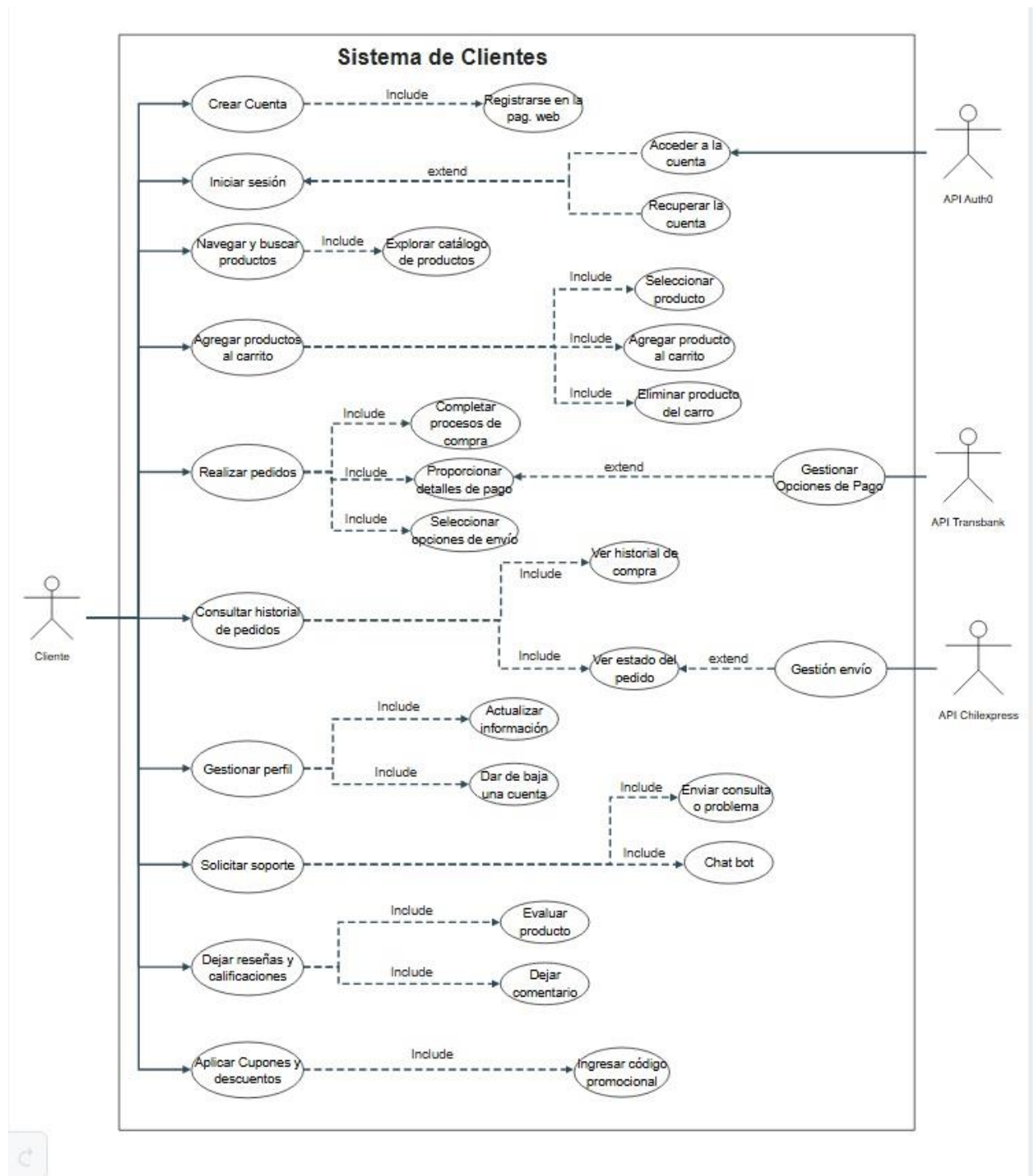
Gerente



Empleado Ventas

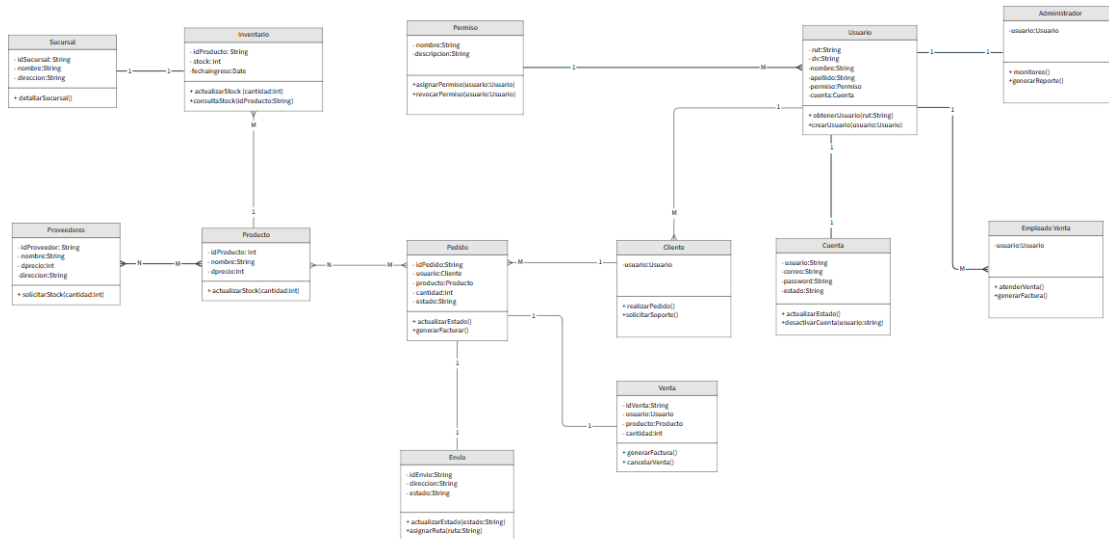


Cliente

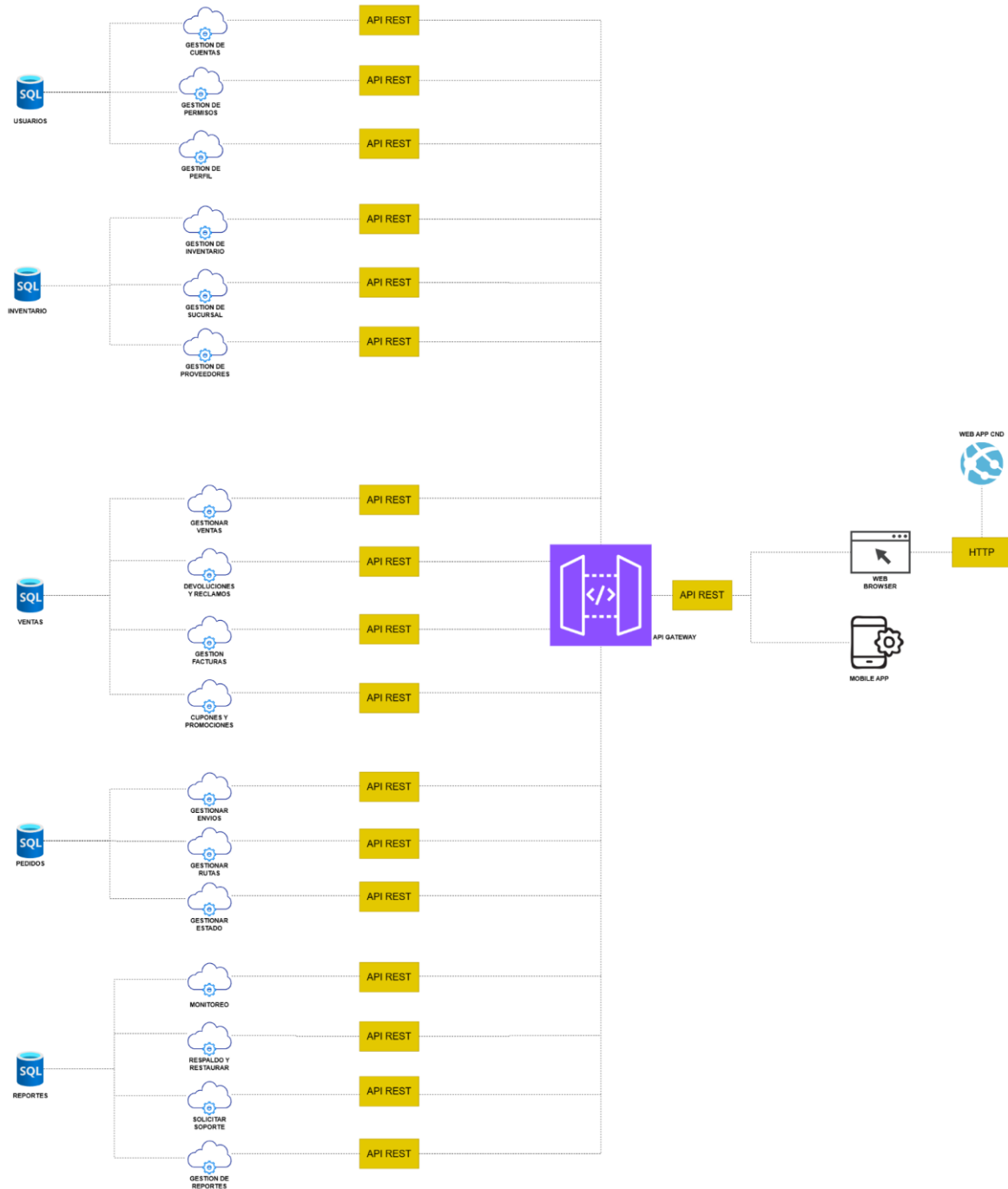


3.2 Diagrama de Clases

UML Class Diagram



3.3 Diagrama de Microservicios



4. Plan de migración

4.1. Objetivos de la Migración

- Modernizar la infraestructura tecnológica de Perfulandia SPA.
- Superar las limitaciones del sistema monolítico actual (fallos de rendimiento y disponibilidad por su actual aumento de demanda).
- Soportar el crecimiento continuo de la empresa y el aumento de clientes.
- Mejorar la escalabilidad, el rendimiento y la mantenibilidad del sistema.
- Implementar una gestión eficiente de ventas, inventario, pedidos y clientes.

4.2. Alcance de la Migración

- Migrar desde un sistema monolítico a una arquitectura de microservicios.
- Implementar un motor de base de datos MySQL.
- Desarrollar los microservicios definidos:
 - Autenticación (manejo de usuarios, login con JWT)
 - Usuario (gestión de usuarios y roles)
 - inventario(control de productos y stock)
 - Ventas (registro de ventas y facturación)
 - Pedidos(gestión de pedidos y envíos)
 - Reportes (generación de informes de ventas e inventario)
 - Notificaciones (envío de alertas a clientes y administradores)
- Integrar APIs externas:
 - Transbank Webpay (pagos en línea)
 - Google Maps / Chilexpress (rutas de entrega y rastreo de envíos)
 - SII (Servicio de impuestos internos) (Facturación electrónica)
 - Twilio/Firebase (notificaciones SMS, Email, Push)
 - Google Analytics/Power BI (análisis y reportes).
 - Keycloak /Auth0 (autenticación)
- Migrar los datos del sistema monolítico a la nueva arquitectura.
- Implementar las funcionalidades descritas para los diferentes perfiles de usuario:
 - Administrador del Sistema
 - Gerente de Sucursal
 - Empleado de Ventas.
 - Cliente vía Web

4.3. Fases de la Migración

● Fase 1: Planificación y Preparación

- Análisis detallado del sistema monolítico existente
- Diseño de la nueva arquitectura de microservicios (API Gateway + Microservicios RESTful)
- Selección de tecnologías y herramientas:
 - Frontend: HTML, CSS, JavaScript.
 - Backend: Spring Boot (Java) con microservicios.
 - Base de Datos: MySQL.
 - Gestión de Código: GitHub + Visual Studio.
 - Metodología de Trabajo: SCRUM (con posibilidad de integrar Kanban).
 - Mensajería: RabbitMQ/Kafka (para comunicación entre microservicios).
- Definición de roles y responsabilidades del equipo de migración.
- Establecimiento de los criterios de éxito de la migración.

● Fase 2: Desarrollo de los Microservicios

- Desarrollo de cada microservicio y sus APIs RESTful.
- Implementación de la comunicación entre microservicios (RabbitMQ/Kafka).
- Desarrollo de la API Gateway para centralizar las peticiones del frontend.
- Desarrollo de las integraciones con las APIs externas.
- Desarrollo de pruebas unitarias e integración.
- Configuración del entorno de desarrollo, pruebas y producción.

● Fase 3: Migración de Datos

- Análisis de los datos del sistema monolítico.
- Diseño y desarrollo de los scripts de migración de datos.
- Pruebas de la migración de datos en un entorno de pruebas.
- Ejecución de la migración de datos en el entorno de producción.
- Verificación de la integridad y consistencia de los datos migrados.

● Fase 4: Implementación y Pruebas

- Despliegue de los microservicios y la API Gateway en el entorno de producción.
- Pruebas de integración y de sistema de la nueva arquitectura.
- Pruebas de rendimiento y escalabilidad del sistema.
- Pruebas de seguridad del sistema.

- Validación de los criterios de aceptación de la migración.

- **Fase 5: Despliegue y Post-Migración**

- Despliegue de la nueva arquitectura de microservicios en producción.
- Monitoreo y seguimiento del sistema en producción.
- Soporte y mantenimiento del sistema.
- Optimización y mejora continua del sistema.

4.4. Estrategia de Migración

- Se recomienda una estrategia de migración **incremental**. Esto permite migrar funcionalidades por partes, reduciendo el riesgo y permitiendo que la empresa comience a utilizar las nuevas funcionalidades antes de que la migración completa finalice.

4.5. Planificación Detallada

- Esta sección requiere una planificación más detallada que se realizará durante el proyecto. Debe incluir:
 - Listado de tareas específicas para cada fase.
 - Asignación de responsables para cada tarea.
 - Cronograma detallado con fechas de inicio y fin para cada tarea.
 - Estimación de los recursos necesarios (personal, hardware, software, etc).
 - Uso de herramientas de gestión de proyectos (Trello) para el seguimiento de las tareas.
 - Implementación de la metodología SCRUM para la gestión del proyecto (backlog, sprint, sprint review).

4.6. Gestión de Riesgos

- **Riesgos Identificados:**
 - Fallos en el sistema durante la migración.
 - Pérdida o corrupción de datos durante la migración.
 - Problemas de rendimiento en la nueva arquitectura.
 - Dificultades en la integración de los microservicios.
 - Retrasos en el cronograma.
 - Problemas de seguridad.

● Planes de Mitigación:

- Realizar pruebas exhaustivas en un entorno de pruebas antes de la migración a producción.
- Implementar un plan de rollback en caso de problemas.
- Realizar copias de seguridad de los datos antes de la migración y validar su integridad después.
- Realizar pruebas de rendimiento y escalabilidad durante el desarrollo y las pruebas.
- Definir claramente las APIs y los protocolos de comunicación entre los microservicios.
- Establecer un cronograma realista y flexible, con hitos claros y entregables definidos.
- Implementar medidas de seguridad en todas las etapas del desarrollo y la migración.

4.7. Criterios de Aceptación

- El sistema cumple con todos los requerimientos funcionales y no funcionales definidos.
- El sistema tiene un rendimiento y una escalabilidad adecuados.
- El sistema es seguro y confiable.
- La migración de datos se ha realizado correctamente y los datos son consistentes.
- Los usuarios han sido capacitados y pueden utilizar el nuevo sistema.

4.8. Comunicación

- Establecer un plan de comunicación con los stakeholders del proyecto (equipo de desarrollo, gerencia de Perfulandia SPA, usuarios del sistema).
- Comunicar el progreso del proyecto de forma regular.
- Proporcionar capacitación y soporte a los usuarios durante la migración.
- Gestionar las expectativas de los stakeholders.

5. Consideraciones Éticas

5.1. Privacidad y Seguridad

- **Privacidad de Datos:**

- El sistema debe proteger la información personal de los usuarios (clientes, administradores, gerentes, empleados).
- Se deben implementar medidas de seguridad para prevenir el acceso no autorizado, la alteración, la divulgación o la destrucción de los datos.
- Se debe cumplir con las regulaciones de protección de datos aplicables (por ejemplo, la Ley de Protección de Datos Personales).
- Se debe informar a los usuarios sobre cómo se recopilan, utilizan y almacenan sus datos, y obtener su consentimiento cuando sea necesario.

- **Seguridad del Sistema:**

- El sistema debe ser seguro y resistente a ataques cibernéticos (por ejemplo, inyección SQL, ataques de denegación de servicio).
- Se deben implementar mecanismos de autenticación y autorización robustos para controlar el acceso a las diferentes funcionalidades del sistema.
- Se deben realizar auditorías de seguridad periódicas para identificar y corregir posibles vulnerabilidades.
- Se debe garantizar la confidencialidad, integridad y disponibilidad de la información.

5.2. Inclusión y Accesibilidad

- **Accesibilidad:**

- El sistema debe ser accesible para personas con discapacidades (por ejemplo, personas con discapacidad visual, auditiva, motriz o cognitiva).
- Se deben seguir las pautas de accesibilidad web (por ejemplo, WCAG) para garantizar que el sistema sea utilizable por todos los usuarios.
- Se deben proporcionar alternativas para el contenido no textual (por ejemplo, descripciones de imágenes, transcripciones de audio).
- Se debe asegurar que la interfaz de usuario sea fácil de usar y navegar.

- **Inclusión:**

- El sistema debe ser inclusivo y no discriminar a ningún grupo de personas (por ejemplo, por motivos de género, raza, religión, orientación sexual, etc.).
- Se debe considerar la diversidad de los usuarios al diseñar el sistema.
- Se debe evitar el uso de lenguaje excluyente o estereotipado.

5.3. Transparencia y Responsabilidad

- **Transparencia:**

- El sistema debe ser transparente en su funcionamiento.
- Se debe informar a los usuarios sobre las decisiones automatizadas que toma el sistema (por ejemplo, recomendaciones de productos, personalización de contenido).
- Se debe proporcionar a los usuarios la capacidad de comprender y cuestionar estas decisiones.

- **Responsabilidad:**

- Los desarrolladores y la empresa son responsables del impacto que el sistema pueda tener en los usuarios y en la sociedad.
- Se deben considerar las posibles consecuencias negativas del sistema y tomar medidas para mitigarlas.
- Se debe establecer un mecanismo para que los usuarios puedan reportar problemas o quejas.

5.4. Equidad y No Discriminación

- **Equidad:**

- El sistema debe tratar a todos los usuarios de manera justa y equitativa.
- Se debe evitar la discriminación algorítmica, que puede ocurrir cuando los algoritmos utilizados en el sistema perpetúan o amplifican los sesgos existentes.
- Se deben realizar pruebas para detectar y corregir posibles sesgos en el sistema.

- **No Discriminación:**

- El sistema no debe discriminar a ningún usuario por ningún motivo.
- Se deben considerar las necesidades de todos los usuarios al diseñar el sistema.

5.5. Impacto Social y Ambiental

- **Impacto Social:**

- Se debe considerar el impacto social que el sistema puede tener en la comunidad.
- Se debe evaluar si el sistema puede contribuir al bienestar social o si puede tener consecuencias negativas (por ejemplo, desplazamiento de trabajadores).
- Se debe promover el uso responsable del sistema.

- **Impacto Ambiental:**

- Se debe considerar el impacto ambiental del sistema.
- Se debe buscar formas de reducir el consumo de energía y los residuos generados por el sistema.
- Se debe promover el uso de tecnologías sostenibles.