
Especificación de requisitos de software

Para

Sistema de Gestión de Paquetes

Versión 1.0 aprobado

**Preparado por Valery Aurela Perez. Gabriel León Simancas.
Nilson Rivera Gutiérrez. Dilson Rivera Gutiérrez**

Universidad tecnológica de Bolívar

01/03/2025

Tabla de contenido

1. Introducción	1
2. Descripción general.....	2
2.1 Características del producto.....	2
2.2 Clases de usuarios y características	2
2.3 Limitaciones de diseño y aplicación	3
2.4 Documentación del Usuario.....	4
2.5 Supuestos y dependencias.....	6
3. Características del sistema	7
3.1 Características de Sistema	¡Error! Marcador no definido.
3.1.1 Descripción y prioridad	¡Error! Marcador no definido.
3.1.2 Secuencias de estímulo/respuesta	¡Error! Marcador no definido.
3.1.3 Requisitos funcionales.....	¡Error! Marcador no definido.
4. Requisitos de la interfaz	10
4.1 Interfaces de Usuario	¡Error! Marcador no definido.
4.2 Interfaces de hardware	¡Error! Marcador no definido.
4.3 Interfaces de software	¡Error! Marcador no definido.
4.4 Interfaces de Comunicación.....	¡Error! Marcador no definido.
5. Otros requisitos No funcionales	12
5.1 Requerimientos de desempeño	¡Error! Marcador no definido.
5.2 Requisitos de seguridad	¡Error! Marcador no definido.
5.3 Atributos de calidad del software.....	¡Error! Marcador no definido.
6. Otros Requerimientos	14
7. Referencias.....	¡Error! Marcador no definido.

Historial de revisiones

Nombre	Fecha	Motivo de los cambios	Versión
Revisión 1	02 de marzo 2025	Versión inicial del código. Se implementaron las clases base 'Paquete', 'Envío', 'Usuario' y 'SistemaLogistica', así como el menú principal para gestionar los envíos.	-1 +1,226
Revisión 2	03 de marzo 2025	Mejoras en la autenticación de usuarios. Se agregó validación para verificar si un usuario está registrado antes de autenticarse.	-1 +1,226
Revisión 3	04 de marzo 2025	Optimización de la función para agregar paquetes a un envío. Se mejoró la búsqueda de paquetes en la lista y se agregó un mensaje de confirmación más detallado.	-1 +1,226
Revisión 4	05 de marzo 2025	Manejo de errores en el registro de envíos. Se implementaron verificaciones para evitar la creación de envíos duplicados y se corrigieron posibles fallos.	-1 +1,226
Versión Final	06 de marzo 2025	Publicación de la versión final. Se hicieron ajustes finales y pruebas para garantizar la estabilidad del código, sin cambiar el número de líneas.	-1 +1,226

1. Introducción

El software desarrollado es un sistema de logística diseñado para gestionar envíos y paquetes de manera eficiente. Permite registrar paquetes, crear envíos, asignar paquetes a envíos y realizar un seguimiento de su estado. Además, cuenta con un módulo de autenticación para garantizar la seguridad en el acceso a las funciones del sistema. Su objetivo principal es optimizar la logística de envíos, reduciendo errores manuales y mejorando el control centralizado de los paquetes.

Este sistema ofrece beneficios como la automatización de procesos, mayor seguridad en la gestión de usuarios y un flujo de trabajo más eficiente. Está alineado con estrategias empresariales enfocadas en mejorar la eficiencia operativa y optimizar los tiempos de entrega. Su diseño modular permite una futura escalabilidad e integración con sistemas avanzados de rastreo y optimización de rutas.

2. Descripción general

El Sistema de Logística es un software autónomo desarrollado para la gestión eficiente de paquetes y envíos. No forma parte de una familia de productos previa ni sustituye sistemas existentes; más bien, ha sido diseñado desde cero con un enfoque modular y escalable. Su objetivo principal es automatizar y optimizar los procesos logísticos, permitiendo la gestión de usuarios, el registro de paquetes y la asignación de estos a envíos, con un sistema de seguimiento del estado de los mismos.

Este software funciona como un sistema independiente, aunque su diseño permite la integración con otros sistemas empresariales mediante futuras interfaces de conexión. La interacción entre los módulos internos (usuarios, paquetes y envíos) se gestiona a través de estructuras de datos eficientes, garantizando un flujo de trabajo fluido. Su implementación mejora la seguridad y el control en la gestión logística, reduciendo el margen de error en la administración de envíos.

2.1 Características del producto

El Sistema de Logística ofrece una solución integral para la gestión eficiente de paquetes y envíos. Entre sus principales funcionalidades, permite el registro y autenticación de usuarios, garantizando un acceso seguro al sistema. Además, los usuarios pueden registrar paquetes, ingresando información clave como su peso, categoría y descripción. También facilita la creación de envíos, permitiendo asignar paquetes a destinatarios específicos y dar seguimiento a su estado en tiempo real.

El sistema organiza los envíos en una estructura de datos optimizada, permitiendo la actualización del estado de los envíos y la visualización de la información en cualquier momento. Su diseño modular permite su integración con otros sistemas logísticos en el futuro. La interfaz basada en consola ofrece una interacción clara y sencilla, asegurando una experiencia de usuario intuitiva y eficiente.

2.2 Clases de usuarios y características

El Sistema de Logística está diseñado para ser utilizado por diferentes clases de usuarios, cada una con funciones y niveles de acceso específicos. A continuación, se detallan las clases de usuarios previstas:

1. Administrador del sistema

- **Descripción:** Usuario con el máximo nivel de privilegios dentro del sistema.
- **Características:**
 - Puede **registrar nuevos usuarios** y gestionar credenciales.
 - Tiene acceso a **todas las funcionalidades del sistema**, incluida la modificación de envíos y paquetes.
 - Puede **actualizar el estado de los envíos** en cualquier momento.
 - Requiere conocimientos básicos en logística y administración de sistemas.

2. Operador logístico

- **Descripción:** Usuario encargado de la gestión operativa de los envíos y paquetes.
- **Características:**
 - Puede **registrar paquetes** e ingresarlos en el sistema.

- Puede **crear y gestionar envíos**, asignando paquetes a destinatarios.
- Puede **actualizar el estado de los envíos** a medida que se procesan.
- No puede registrar nuevos usuarios ni modificar credenciales.
- Requiere conocimientos básicos de gestión de logística.
- **3. Cliente o destinatario**
- **Descripción:** Usuario que recibe los paquetes y realiza consultas sobre sus envíos.
- **Características:**
 - Puede **autenticarse en el sistema** para consultar el estado de sus envíos.
 - No puede modificar paquetes ni envíos, solo visualizar información.
 - No requiere conocimientos técnicos previos.

Clases de usuarios prioritarias

- **Primarias:** Administradores del sistema y operadores logísticos, ya que realizan las funciones esenciales del sistema.
- **Secundarias:** Clientes o destinatarios, quienes solo requieren acceso a la consulta de información.

2.3 Limitaciones de diseño y aplicación

El desarrollo del **Sistema de Logística** está sujeto a diversas restricciones y consideraciones que impactarán su implementación. A continuación, se detallan las principales limitaciones:

1. Restricciones técnicas

- **Lenguaje de programación:** El sistema está desarrollado en **C++**, lo que limita la integración con tecnologías modernas basadas en web o bases de datos avanzadas.
- **Memoria y procesamiento:** Dado que se almacena la información en estructuras de datos en memoria (vectores y mapas), el sistema puede presentar **limitaciones de escalabilidad** para manejar grandes volúmenes de datos.
- **Paralelismo:** Actualmente, el sistema no maneja **procesamiento concurrente** ni acceso multiusuario simultáneo.

2. Interoperabilidad e integración

- **Base de datos:** No se incluye una integración con bases de datos, lo que significa que los datos no son persistentes entre sesiones.
- **Interfaces con otras aplicaciones:** No existen mecanismos de integración con **sistemas externos de logística, API de terceros o plataformas de seguimiento de envíos**.
- **Protocolos de comunicación:** No se han definido protocolos como **REST API, MQTT o WebSockets** para interacción con otros sistemas.

3. Seguridad y control de acceso

- **Autenticación limitada:** Actualmente, la autenticación de usuarios es **básica** y sin cifrado, lo que representa un riesgo de seguridad.
- **Roles de usuario:** No hay un **sistema de permisos** para restringir funciones según el tipo de usuario.
- **Protección de datos:** No se implementan medidas de **encriptación** o almacenamiento seguro de credenciales.

4. Normativas y políticas

- **Cumplimiento normativo:** Si el sistema se implementa en una empresa de logística real, deberá ajustarse a normativas como **GDPR** (protección de datos) y **ISO 9001** (gestión de calidad).
- **Mantenimiento del software:** Si el cliente final debe encargarse del mantenimiento, es recomendable seguir **convenciones de programación** y documentación clara.

5. Consideraciones de diseño

- **Interfaz de usuario:** La interacción es **únicamente por consola**, lo que podría dificultar su uso para usuarios no técnicos.
- **Escalabilidad:** Para manejar grandes volúmenes de datos, se necesitaría una arquitectura basada en **almacenamiento en bases de datos y procesamiento distribuido**.

2.4 Documentación del Usuario

Manual Técnico (para desarrolladores y soporte TI)

Introducción

Este manual está dirigido a desarrolladores y personal de soporte técnico encargados del mantenimiento y mejora del Sistema de Logística. Contiene información detallada sobre la arquitectura del software, estructura del código, tecnologías utilizadas y procedimientos clave para futuras modificaciones o ampliaciones.

Estructura del Código

El sistema está desarrollado en **C++**, utilizando la biblioteca estándar para entrada/salida y estructuras de datos. Se organiza en varias clases clave:

- **Paquete:** Representa un paquete con atributos como ID, peso, categoría y descripción.
- **Envío:** Gestiona un conjunto de paquetes, manteniendo el estado del envío.
- **Usuario:** Maneja la autenticación de usuarios mediante un sistema básico de credenciales.
- **Sistema Logística:** Clase principal que administra paquetes, envíos y usuarios, proporcionando funciones para la interacción del usuario.

Funcionamiento de la Autenticación

El sistema utiliza un map de la STL para almacenar usuarios registrados y sus contraseñas. Para autenticarse:

1. El usuario ingresa su nombre y contraseña.
2. Se verifica si el usuario existe en el mapa.
3. Si la contraseña coincide, la autenticación es exitosa.

- **Limitaciones actuales:** No se utiliza cifrado para las contraseñas, lo que implica riesgos de seguridad si se implementa en entornos reales. Se recomienda usar **hashing (SHA-256)** en futuras versiones.

Almacenamiento de Datos

Actualmente, el sistema almacena la información en **memoria volátil** (vectores y mapas en RAM). Esto significa que los datos se pierden al cerrar la aplicación.

Posibles mejoras:

- Implementar persistencia de datos usando archivos **JSON, CSV o SQLite**.
- Incluir serialización/deserialización para recuperar información tras reiniciar el sistema.

Funcionalidades Clave

Registro de Paquetes

- Se solicita ID, peso, categoría y descripción.
- Se almacena en un vector de objetos Paquete.

Creación de Envíos

- Se genera un ID de envío y se asigna un destinatario.
- Los paquetes pueden agregarse a un envío existente.

Gestión de Envíos

- Se pueden consultar los envíos creados y actualizar su estado.

Posibles Mejoras y Escalabilidad

- **Interfaz gráfica (GUI):** Implementar una interfaz en Qt o GTK.
- **Persistencia de datos:** Uso de bases de datos en lugar de estructuras en memoria.
- **Seguridad:** Implementar autenticación con hashing y almacenamiento seguro de credenciales.
- **Soporte para red:** Posibilitar conexión cliente-servidor con sockets para acceso remoto.

Instalación y Ejecución

-
- **Requisitos previos:**
- Compilador C++ compatible con C++11 o superior (GCC, Clang o MSVC).
- Sistema operativo: Windows, Linux o MacOS.

Comandos para compilar y ejecutar:

```
g++ -o logistica main.cpp  
./logística
```

2.5 Supuestos y dependencias

Para el correcto funcionamiento del **Sistema de Logística** se asumen las siguientes condiciones:

- **Uso de un entorno compatible con C++:** Se supone que el sistema se ejecutará en un entorno con un compilador que soporte **C++11 o superior** (GCC, Clang o MSVC).
- **Datos almacenados en memoria:** Actualmente, la aplicación no utiliza bases de datos ni almacenamiento persistente, por lo que los datos se pierden al cerrar el programa. Se asume que esto es aceptable en esta versión.
- **Autenticación básica sin cifrado:** Se asume que los usuarios ingresarán credenciales correctamente y que no es necesario cifrar contraseñas en esta versión, aunque se recomienda en futuras implementaciones.
- **Interacción por consola:** Se considera que los usuarios pueden interactuar con la aplicación a través de una terminal o consola sin necesidad de una interfaz gráfica.
- **Cantidad de datos manejable:** Se asume que la cantidad de paquetes y envíos gestionados será **moderada**, ya que el uso de estructuras en memoria puede volverse ineficiente a gran escala.

El sistema tiene las siguientes dependencias técnicas y operativas:

- **Librerías estándar de C++:** El código utiliza `<iostream>`, `<string>`, `<vector>` y `<map>`, por lo que cualquier entorno de desarrollo debe contar con estas librerías disponibles.
- **Compilador de C++:** Se requiere un compilador compatible con C++11 o superior. Se recomienda **GCC 7.0+, Clang 6.0+ o MSVC 2017+**.
- **Ejecución en un sistema operativo compatible:** Se supone que la aplicación correrá en **Windows, Linux o macOS** con soporte para entrada/salida estándar (`cin/cout`).
- **Entrada manual de datos:** El sistema depende de la **interacción manual** para registrar paquetes y envíos. No está diseñado para integrarse con sistemas de gestión automática de datos.

Si alguno de estos supuestos cambia o alguna dependencia no está disponible, el funcionamiento del sistema puede verse afectado.

3. Características del sistema

Gestión de Paquetes o Descripción y prioridad

- *La gestión de paquetes incluye la creación, actualización, eliminación y clasificación de paquetes en el sistema. Es una característica de alta prioridad, ya que es fundamental para el funcionamiento del sistema de gestión de envíos.*
- **Secuencias de estímulo/respuesta**
- **Usuario:** Registra un nuevo paquete en el sistema.
 - ❖ *Sistema: Almacena la información del paquete y lo clasifica automáticamente en una de las tres categorías (básico, estándar, dimensionado).*
- **Usuario:** Actualiza la información de un paquete existente.
 - ❖ *Sistema: Modifica la información del paquete en el sistema.*
- **Usuario:** Elimina un paquete.
 - ❖ *Sistema: Elimina la información del paquete del sistema.*
- **Requisitos funcionales**
- *REQ-1: El sistema debe permitir a los usuarios registrar nuevos paquetes especificando sus dimensiones, peso y tipo.*
- *REQ-2: El sistema debe permitir a los usuarios actualizar la información de los paquetes existentes.*
- *REQ-3: El sistema debe permitir a los usuarios eliminar paquetes del sistema.*
- *REQ-4: El sistema debe clasificar automáticamente los paquetes en tres categorías (básico, estándar, dimensionado) basándose en sus características.*

Gestión de Envíos y Trazabilidad

- **Descripción y prioridad**
- *La gestión de envíos y trazabilidad incluye la creación de envíos, asignación de paquetes a destinatarios y seguimiento del progreso del envío. Esta característica es de alta prioridad para asegurar la transparencia y eficiencia en la entrega de paquetes.*
- **Secuencias de estímulo/respuesta**
 - **Usuario:** Crea un nuevo envío asignando paquetes a un destinatario.
 - ❖ *Sistema: Guarda la información del envío y asigna los paquetes al destinatario especificado.*
 - **Usuario:** Consulta el estado de un envío.
 - ❖ *Sistema: Proporciona información de trazabilidad del envío desde la recolección hasta la entrega.*

- **Requisitos funcionales**

- REQ-5: El sistema debe permitir a los usuarios crear envíos asignando paquetes a destinatarios específicos con direcciones detalladas.
- REQ-6: El sistema debe proporcionar información de trazabilidad para cada envío, permitiendo a los usuarios seguir el progreso del envío desde la recolección hasta la entrega.
- REQ-7: El sistema debe permitir la consulta y visualización del estado de los envíos en cualquier momento.

Registro y Autenticación de Usuarios

- **Descripción y prioridad**

- El registro y autenticación de usuarios incluye la creación de cuentas de usuario y el control de acceso a las funcionalidades del sistema. Es una característica de alta prioridad para garantizar la seguridad y personalización del sistema.

- **Secuencias de estímulo/respuesta**

- **Usuario:** Se registra en el sistema.

❖ Sistema: Crea una cuenta de usuario y almacena la información de registro. ▪ Usuario: Inicia sesión en el sistema.

❖ Sistema: Verifica las credenciales del usuario y permite o deniega el acceso según corresponda.

Requisitos funcionales.

- REQ-8: El sistema debe permitir a los usuarios registrar nuevas cuentas proporcionando la información necesaria.
- REQ-9: El sistema debe permitir a los usuarios autenticarse proporcionando sus credenciales.
- REQ-10: El sistema debe garantizar que solo los usuarios autenticados puedan acceder a sus respectivas funcionalidades. Facturación y Gestión de Pagos o Descripción y prioridad La facturación y gestión de pagos incluye la generación de facturas y la realización de pagos por los servicios de envío utilizados. Es una característica de alta prioridad para asegurar la monetización de los servicios y la satisfacción del cliente. o

- **Secuencias de estímulo/respuesta**

- **Usuario:** Solicita la generación de una factura.

❖ Sistema: Genera una factura basada en los servicios de envío utilizados y la presenta al usuario.

- **Usuario:** Realiza un pago.

❖ Sistema: Procesa el pago a través de métodos seguros integrados en la plataforma. o

- REQ-11: El sistema debe generar facturas basadas en los servicios de envío utilizados por el usuario
- REQ-12: El sistema debe permitir a los usuarios realizar pagos a través de métodos seguros integrados en la plataforma.
- REQ-13: El sistema debe registrar y almacenar el historial de facturación y pagos de cada usuario.

Búsqueda y Filtrado de Envíos y Paquetes

- ***Descripción y prioridad***

- *La búsqueda y filtrado de envíos y paquetes permite a los usuarios encontrar información específica de manera rápida y eficiente. Es una característica de alta prioridad para mejorar la usabilidad y eficiencia del sistema.*

- ***Secuencias de estímulo/respuesta***

- ***Usuario:*** Realiza una búsqueda de envíos basándose en criterios específicos.
 - ❖ ***Sistema:*** Proporciona una lista de envíos que cumplen con los criterios de búsqueda.
- ***Usuario:*** Filtra paquetes según características específicas
 - ❖ ***Sistema:*** Presenta una lista de paquetes que cumplen con los criterios de filtrado.

- ***Requisitos funcionales***

- *REQ-14: El sistema debe permitir a los usuarios buscar envíos basándose en criterios como fecha de envío, estado del paquete, destinatario, entre otros.*
- *REQ-15: El sistema debe permitir a los usuarios filtrar paquetes según características específicas como dimensiones, peso y tipo.*
- *REQ-16: El sistema debe mostrar resultados de búsqueda y filtrado de manera eficiente y comprensible.*

4. Requisitos de la interfaz

4.1 Interfaces de Usuario

El sistema se ejecuta en una **interfaz de línea de comandos (CLI)**, donde los usuarios interactúan mediante un menú textual. Las principales características de la interfaz de usuario son:

- **Menú principal** que permite acceder a las funciones del sistema mediante la selección numérica.
- **Mensajes de confirmación y error**, indicando el éxito o fallo de cada operación.
- **Entrada manual de datos** a través de cin, donde el usuario proporciona información como el ID del paquete, peso, destinatario, etc.
- **Formato estructurado de salida**, con información detallada sobre paquetes y envíos.
- **Flujo secuencial de navegación**, sin atajos de teclado ni interfaz gráfica avanzada.

En futuras versiones, se podría implementar una interfaz gráfica basada en **Qt**, **GTK** o una **aplicación web** para mejorar la experiencia del usuario.

4.2 Interfaces de Hardware

El sistema no tiene dependencias específicas de hardware, pero se asume que se ejecutará en un **computador estándar** con teclado y pantalla. Otras consideraciones:

- **Procesador:** Compatible con cualquier CPU moderna capaz de ejecutar un compilador C++.
- **Memoria:** Se recomienda al menos **512 MB de RAM**, aunque el uso de memoria es mínimo.
- **Dispositivo de Entrada/Salida:** Se requiere un **teclado** para la entrada de datos y una **pantalla** para la visualización del menú y la información de los envíos.

4.3 Interfaces de Software

El sistema interactúa únicamente con las librerías estándar de C++:

- **<iostream>** para la entrada y salida de datos.
- **<string>** para manipulación de texto.
- **<vector>** para almacenar listas de paquetes y envíos.
- **<map>** para gestionar usuarios registrados.

No se conecta con bases de datos, servicios web ni otros sistemas en esta versión. En versiones futuras, se podría integrar con:

- **Bases de datos** (MySQL, SQLite) para almacenamiento persistente.
- **APIs web** para rastreo de envíos en tiempo real.

4.4 Interfaces de Comunicación

El sistema **no requiere conexión a redes** en su versión actual, ya que opera de forma local. Sin embargo, en futuras versiones se podrían considerar:

- **Protocolo HTTP/HTTPS** para integración con servicios de rastreo de envíos.
- **Correo electrónico/SMS** para notificaciones de estado de envíos.
- **Cifrado de credenciales** para mejorar la seguridad de la autenticación de usuarios.

5. Otros requisitos No funcionales

5.1 Requerimientos de Desempeño

El sistema debe cumplir con los siguientes requisitos de rendimiento:

- **Tiempo de respuesta:** Las operaciones de consulta y actualización de datos (registro de paquetes/envíos, autenticación de usuario) deben ejecutarse en un tiempo no mayor a **0.5 segundos** en equipos estándar.
- **Escalabilidad:** El sistema debe ser capaz de manejar al menos **1,000 paquetes y envíos** sin una degradación notable del rendimiento.
- **Consumo de memoria:** El uso de memoria debe mantenerse por debajo de **50 MB** en ejecuciones estándar.
- **Eficiencia algorítmica:** Se optimizarán las búsquedas de paquetes y envíos utilizando estructuras de datos eficientes, como map para usuarios y vector para listas de paquetes y envíos.

En futuras versiones, si se implementa una base de datos o una API web, se evaluarán nuevos requisitos de rendimiento.

5.2 Requisitos de Seguridad

El sistema maneja información sensible de usuarios y paquetes, por lo que se establecen las siguientes medidas de seguridad:

- **Autenticación de usuarios:** Se requiere que cada usuario ingrese su nombre de usuario y contraseña antes de realizar operaciones en el sistema.
- **Almacenamiento seguro de contraseñas:** En futuras versiones, las contraseñas se almacenarán de forma segura mediante **hashing y salting** en una base de datos (actualmente, las contraseñas se almacenan en texto plano en memoria).
- **Protección contra accesos no autorizados:** No se permitirá el acceso a funciones sensibles sin autenticación previa.
- **Seguridad en la comunicación (futuro):** En caso de implementarse una API o conexión remota, se usará **HTTPS y cifrado TLS** para proteger los datos.
-

*Si el sistema se amplía para integrarse con otros servicios, se evaluará la conformidad con normativas de seguridad como **ISO 27001** o **GDPR**.*

5.3 Atributos de Calidad del Software

El sistema debe cumplir con los siguientes atributos de calidad:

- **Fiabilidad:** *El sistema debe funcionar de manera estable y no generar fallos inesperados en operaciones comunes.*
- **Usabilidad:** *La interfaz de usuario basada en menú debe ser clara e intuitiva para usuarios sin experiencia técnica.*
- **Mantenibilidad:** *El código debe estar estructurado en clases bien definidas para facilitar futuras modificaciones.*
- **Portabilidad:** *El software debe compilar y ejecutarse sin problemas en cualquier sistema con un compilador de C++ compatible (g++/MSVC).*
- **Escalabilidad:** *El diseño debe permitir la integración futura con bases de datos y sistemas de seguimiento en línea sin necesidad de una reestructuración completa.*
- **Eficiencia:** *Se deben evitar redundancias en la manipulación de datos para optimizar el uso de memoria y CPU.*

6. Otros Requerimientos

6.1 Requisitos de la Base de Datos

En futuras versiones, el sistema podrá integrar una base de datos para el almacenamiento de información. Se consideran los siguientes requisitos:

- **Tipo de base de datos:** Se recomienda una base de datos relacional (ej. MySQL, PostgreSQL) o una solución ligera como SQLite.
- **Estructura:** Se contemplan tablas para **usuarios, paquetes y envíos** con claves primarias y relaciones entre ellas.
- **Seguridad:** Implementación de control de accesos y cifrado en datos sensibles como contraseñas.

6.2 Requisitos de Internacionalización

El sistema debe permitir la posibilidad de soportar múltiples idiomas en futuras versiones. Para ello:

- Se recomienda la separación de cadenas de texto en archivos de configuración externos.
- Implementación de soporte para codificación UTF-8 para caracteres especiales.

6.3 Requisitos Legales y Normativos

El sistema debe cumplir con normativas de privacidad y seguridad de datos, tales como:

- **Reglamento General de Protección de Datos (GDPR)** si se maneja información personal de usuarios.
- **Normas ISO 27001** en caso de implementación en un entorno de seguridad empresarial.
- **Normativa local de protección de datos**, dependiendo del país en el que se implemente.

6.4 Objetivos de Reutilización

El código debe diseñarse de forma modular para facilitar su reutilización en futuros proyectos o versiones. Se recomienda:

- Separar la lógica del negocio en clases independientes.
- Usar patrones de diseño como **MVC** si en el futuro se integra con interfaces gráficas o APIs.

7. Referencias

International Organization for Standardization. (2011). *ISO/IEC 25010:2011 - Systems and software engineering — Systems and software quality requirements and evaluation (SQuaRE) — System and software quality models*. Retrieved from <https://www.iso.org/standard/35733.html>

C++ Reference. (n.d.). *C++ Standard Library documentation*. Retrieved March 6, 2025, from <https://en.cppreference.com>

OWASP Foundation. (2021). *OWASP Top Ten Web Application Security Risks*. Retrieved from <https://owasp.org/www-project-top-ten/>

GitHub. (n.d.). *DevOps CI/CD with GitHub*. Retrieved March 6, 2025, from <https://github.com/resources/articles/devops/ci-cd>

Docker. (n.d.). *What are containers?* Retrieved March 6, 2025, from <https://savio.utb.edu.co/mod/url/view.php?id=390992>

SonarQube. (n.d.). *SonarQube: Code quality tool*. Retrieved March 6, 2025, from <https://savio.utb.edu.co/mod/url/view.php?id=390993>

DevOps Institute. (n.d.). *What is DevOps?* Retrieved March 6, 2025, from <https://savio.utb.edu.co/mod/url/view.php?id=391000>

Universidad Tecnológica de Bolívar. (2025). *Guía de diseño de interfaz de usuario*.

Universidad Tecnológica de Bolívar. (2025). *Documentos internos del proyecto: Casos de uso y manuales de usuario*.

Apéndice A: Glosario

- **SRS (Software Requirements Specification):** Documento que describe los requisitos del software.
- **Paquete:** Unidad de envío que contiene productos y tiene atributos como peso, categoría y descripción.
- **Envío:** Grupo de paquetes dirigido a un destinatario.
- **Autenticación:** Proceso de verificación de credenciales de usuario.
- **Base de datos:** Sistema que almacena información estructurada para su consulta y manipulación.
- **UTF-8:** Formato de codificación de caracteres que permite representar diferentes alfabetos y símbolos.

Apéndice B: Lista de Problemas

Actualmente, quedan pendientes los siguientes puntos por resolver:

1. **Implementación de persistencia:** Actualmente, los datos se manejan en memoria, lo que implica pérdida de información al cerrar la aplicación. Se necesita una solución de almacenamiento (archivo o base de datos).
2. **Seguridad de contraseñas:** Evaluar el uso de hashing en futuras versiones.
3. **Internacionalización:** El sistema solo está en español, falta soporte para otros idiomas.
4. **Interfaz gráfica:** Actualmente, el sistema funciona en modo consola. Se evaluará la integración con una interfaz gráfica en el futuro.