

Especificación de Requisitos de Software

Proyecto: The Hub

09/09/2025

Contenido

FICHA DEL DOCUMENTO	3
1. INTRODUCCIÓN.....	4
1.1. PROPÓSITO	4
1.2. ÁMBITO DEL SISTEMA	4
1.3. DEFINICIONES, ACRÓNIMOS Y ABREVIATURAS	4
1.4. REFERENCIAS.....	4
1.5. VISIÓN GENERAL DEL DOCUMENTO.....	4
2. DESCRIPCIÓN GENERAL	5
2.1. PERSPECTIVA DEL PRODUCTO	5
2.2. FUNCIONES DEL PRODUCTO	6
2.3. CARACTERÍSTICAS DE LOS USUARIOS	7
2.4. RESTRICCIONES	8
2.5. SUPOSICIONES Y DEPENDENCIAS.....	9
2.6. REQUISITOS FUTUROS.....	9
3. REQUISITOS ESPECÍFICOS	11
3.1 REQUISITOS COMUNES DE LAS INTERFACES	12
3.1.1 Interfaces de usuario	<i>¡Error! Marcador no definido.</i>
3.1.2 Interfaces de hardware.....	<i>¡Error! Marcador no definido.</i>
3.1.3 Interfaces de software	13
3.1.4 Interfaces de comunicación.....	14
3.2 REQUISITOS FUNCIONALES	14
3.3 REQUISITOS NO FUNCIONALES.....	14
3.3.1 Requisitos de rendimiento.....	16
3.3.2 Seguridad	17
3.3.3 Fiabilidad.....	18
3.3.4 Disponibilidad.....	18
3.3.5 Mantenibilidad	19
3.3.6 Portabilidad.....	20
3.4 OTROS REQUISITOS.....	¡ERROR! MARCADOR NO DEFINIDO.

Ficha del documento

Fecha	Revisión	Autor	Modificación
07/09/2025	1.0	Victor Ramos	Creacion inicial del documento
09/09/2025	1.1	Victor Ramos	Finalizacion del documento

Documento validado por las partes en fecha: [\[Fecha\]](#)

Por el cliente

Por la empresa suministradora

[Firma]

[Firma]

Sr./Sra. [\[Nombre\]](#)

Sr./Sra. [\[Nombre\]](#)

1. Introducción

En esta sección se proporcionará una introducción a todo el documento de Especificación de Requisitos Software (ERS). Consta de varias subsecciones: propósito, ámbito del sistema, definiciones, referencias y visión general del documento.

1.1. Propósito

Este documento tiene como propósito establecer la definición y el alcance del proyecto The Hub. Está destinado a ser entregado y revisado por el cliente, conteniendo todos los datos necesarios para su comprensión.

1.2. Ámbito del Sistema

- Nombre: The Hub
- Nuestro sistema se encargara de poder vender y enviar productos de accesorios electrónicos a todo el país
- Esto abarcará desde la mejora de la plataforma de ventas online y la gestión de envíos nacionales, hasta el incremento en la satisfacción del cliente y el volumen de negocio.

1.3. Definiciones, Acrónimos y Abreviaturas

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

ERS: Especificación de Requerimientos de Software.

Comanda: detalle del pedido del plato o menú.

1.4. Referencias

1.5. Visión General del Documento

En esta subsección se describe brevemente los contenidos y la organización del resto de la ERS.

Este documento consta de un área de definición del negocio, un área de especificación de requisitos, se proporciona el detalle de los requerimientos a través de formularios de caso de uso como anexos.

2. Descripción General

En esta sección se describen todos aquellos factores que afectan al producto y a sus requisitos. No se describen los requisitos, sino su contexto. Esto permitirá definir con detalle los requisitos en la sección 3, haciendo que sean más fáciles de entender.

Normalmente, esta sección consta de las siguientes subsecciones: Perspectiva del producto, funciones del producto, características de los usuarios, restricciones, factores que se asumen y futuros requisitos.

2.1. Perspectiva del Producto

Perspectiva del producto

- The Hub es una nueva plataforma de e-commerce diseñada para vender accesorios electrónicos.
- Es un sistema independiente que funcionará como la principal herramienta de venta del negocio. No depende de otros sistemas (salvo de los servicios de pago o de envío externos).
- Su principal objetivo es ser el canal de venta online que permita al negocio llegar a clientes a nivel nacional.

Funciones del producto

- El sistema permitirá a los usuarios explorar y buscar productos por categoría.
- Los usuarios podrán agregar productos al carrito de compras y realizar el pago.
- Los clientes podrán gestionar sus pedidos y ver su estado de envío.
- El sistema deberá incluir un panel de administración para que el personal de The Hub pueda gestionar el inventario, procesar pedidos y actualizar la información de los productos.

Características de los usuarios

- **Clientes:** Son los usuarios finales que compran productos. Necesitan una interfaz simple e intuitiva para buscar, comprar y seguir sus pedidos.
- **Administradores:** Son los empleados de The Hub. Necesitan un panel seguro y fácil de usar para gestionar el catálogo de productos, el inventario, los pedidos y los datos de los clientes.

Restricciones

- **Tecnológicas:** El sitio debe ser compatible con los navegadores web modernos (Chrome, Firefox, Safari, Edge).

- **Seguridad:** El sistema debe cumplir con los estándares de seguridad para el manejo de transacciones con tarjeta de crédito (por ejemplo, PCI DSS).
- **Presupuesto/Tiempo:** El proyecto debe ser desarrollado dentro de un presupuesto y un cronograma definidos.

Factores que se asumen

- Se asume que The Hub ya tiene un proveedor logístico para los envíos a nivel nacional.
- Se asume que se utilizará una pasarela de pago (como Mercado Pago o PayPal) para las transacciones.
- Se asume que el contenido inicial del catálogo de productos y las imágenes estarán disponibles antes de comenzar el desarrollo.

2.2. Funciones del Producto

Módulo de Cliente (E-commerce)

- **Búsqueda y Navegación:**
 - Permite a los usuarios buscar productos por nombre o palabra clave.
 - Facilita la navegación por categorías de productos.
- **Visualización de Productos:**
 - Muestra la información de cada producto (descripción, precio, imágenes).
- **Carrito de Compras:**
 - Permite agregar, modificar y eliminar productos del carrito.
 - Muestra el subtotal y total de la compra.
- **Proceso de Pago (Checkout):**
 - Guía al usuario para ingresar la información de envío y de pago.
 - Se conecta con la pasarela de pago para procesar la transacción.
- **Seguimiento de Pedido:**
 - Permite al cliente ver el estado actual de su pedido (pendiente, en preparación, enviado, entregado).

Módulo de Administración

- **Gestión de Productos:**
 - Permite al administrador agregar, editar o eliminar productos del catálogo.
 - Permite actualizar el inventario y los precios.
- **Gestión de Pedidos:**
 - Muestra todos los pedidos recibidos.
 - Permite cambiar el estado de los pedidos (ej. de "pendiente" a "enviado").
- **Reportes de Ventas:**
 - Genera reportes sobre las ventas, productos más vendidos y otras métricas clave.

2.3. Características de los Usuarios

1. Usuario Cliente (Usuario Final)

- **Nivel educacional y experiencia:** No se requiere un nivel educativo o técnico específico. Se asume que el usuario tiene un nivel de experiencia **básico** en el uso de internet y en la navegación de sitios de e-commerce.
- **Experiencia técnica:** Conocimiento general de navegación web y uso de dispositivos como computadoras o teléfonos móviles. No se espera que el usuario tenga conocimientos técnicos avanzados.
- **Objetivos:**
 - Encontrar y comprar productos de forma rápida y sencilla.
 - Ver información clara sobre productos, precios y detalles de envío.
 - Realizar un pago de manera segura.
 - Rastrear el estado de su pedido.

2. Usuario Administrador

- **Nivel educacional y experiencia:** Se requiere un conocimiento intermedio de computación a nivel de usuario. Se asume que el usuario tiene experiencia en la gestión de inventarios y procesos de negocio.

- **Experiencia técnica:** Debe estar familiarizado con el uso de sistemas operativos de escritorio, navegadores web y, idealmente, hojas de cálculo (como Excel) para el manejo de datos.
- **Objetivos:**
 - Gestionar el catálogo de productos (añadir, editar, eliminar).
 - Supervisar y procesar los pedidos de los clientes.
 - Controlar y actualizar el inventario.
 - Acceder a reportes de ventas para tomar decisiones de negocio.

2.4. Restricciones

1. Políticas de la empresa

- **Identidad de Marca:** El diseño del sitio web debe adherirse estrictamente a las guías de estilo de la empresa (colores, logotipos y tipografías).
- **Proceso de Negocio:** El flujo de compra y la gestión de pedidos deben seguir los procesos definidos por la gerencia de The Hub.

2. Interfaces con otras aplicaciones

- **Pasarela de Pago:** El sistema debe integrar la pasarela de pago X (ej. Mercado Pago, PayPal) para todas las transacciones.
- **Servicio de Envío:** La plataforma debe conectarse con la API de un proveedor de servicios de logística (ej. Chilexpress, Starken) para gestionar y seguir los envíos.

3. Consideraciones técnicas

- **Tecnologías:** El sistema debe ser desarrollado utilizando un stack tecnológico específico (por ejemplo: springboot para el backend, JavaScript con bootstrap y React para el frontend).
- **Protocolos de comunicación:** Todas las comunicaciones entre el cliente y el servidor deben ser seguras, utilizando el protocolo HTTPS.
- **Rendimiento:** El tiempo de carga de las páginas no debe exceder los 3 segundos. El sistema debe ser capaz de soportar un número X de usuarios concurrentes.

4. Requisitos de seguridad y control

- **Seguridad:** El sistema debe cumplir con los estándares de seguridad para proteger los datos personales de los clientes y la información de sus transacciones.

- **Auditoría:** Todas las acciones realizadas por los administradores (ej. cambios de precio, eliminación de productos) deben ser registradas en un log de auditoría.
- **Operaciones paralelas:** El sistema debe ser capaz de procesar múltiples pedidos y transacciones de forma simultánea sin generar errores ni inconsistencias en el inventario.

2.5. Suposiciones y Dependencias

Suposiciones

- Se asume que la **pasarela de pago** elegida (por ejemplo, Mercado Pago, PayPal) tendrá una documentación completa y estable para su integración.
- Se asume que **The Hub** tiene un contrato de logística y envíos vigente con un proveedor de servicios (ej. Correos de Chile, Chilexpress) para el envío de productos a todo el país.
- Se asume que la **base de datos** de regiones y comunas necesaria para el formulario de registro estará disponible y actualizada.
- Se asume que los **diseños de interfaz** de usuario (UI/UX) se proporcionarán antes del inicio del desarrollo.

Dependencias

- **Servicio de Pasarela de Pago:** El sistema depende del correcto funcionamiento y la disponibilidad de la API de la pasarela de pago para procesar transacciones.
- **Servicio de Envío:** La plataforma depende del servicio del proveedor logístico para calcular los costos de envío y para el seguimiento de los paquetes.
- **Base de Datos de Ubicación:** El formulario de registro de usuarios depende de una base de datos externa de regiones y comunas para asegurar que la información de dirección sea precisa y esté validada.

2.6. Requisitos Futuros

Funcionalidades para Clientes

- **Perfiles de Usuario:** Los usuarios podrán editar su información personal y gestionar sus direcciones guardadas para agilizar futuras compras.
- **Historial de Compras:** Los clientes podrán ver un listado completo de sus pedidos anteriores y su estado.

- **Sistema de Reseñas y Calificaciones:** Los usuarios podrán dejar comentarios y calificar los productos que han comprado, lo que ayudará a otros clientes.
- **Lista de Deseos (Wishlist):** Los usuarios podrán guardar productos que les interesen para comprarlos más tarde.
- **Comparador de Productos:** Se permitirá a los usuarios comparar las especificaciones de dos o más productos para tomar una decisión de compra informada.

Funcionalidades para el Administrador

- **Gestión de Promociones y Cupones:** El administrador podrá crear y gestionar códigos de descuento para campañas de marketing.
- **Notificaciones de Inventario:** El sistema enviará alertas automáticas al administrador cuando el stock de un producto esté bajo.

Mejoras de la Plataforma

- **Notificaciones Automáticas:** El sistema enviará notificaciones por correo electrónico a los usuarios sobre el estado de sus pedidos (confirmación, envío, entrega).
- **Integración con Redes Sociales:** Se permitirá a los usuarios registrarse e iniciar sesión utilizando sus cuentas de redes sociales (ej. Google, Facebook).

3. Requisitos Específicos

1. Módulo de Autenticación y Cuentas

- **REQ-001 - Registro de Usuario**
 - **Descripción:** El sistema debe permitir que un usuario se registre con un nombre de usuario, correo electrónico, contraseña, región y comuna.
 - **Prioridad:** Esencial
 - **Fuente:** Administrador
- **REQ-002 - Verificación de Correo**
 - **Descripción:** El sistema debe enviar un correo electrónico de verificación al usuario después del registro para validar su dirección de correo electrónico.
 - **Prioridad:** Deseable
 - **Fuente:** Administrador
- **REQ-003 - Restablecimiento de Contraseña**
 - **Descripción:** El sistema debe ofrecer una funcionalidad para que los usuarios restablezcan su contraseña mediante un enlace enviado a su correo electrónico.
 - **Prioridad:** Esencial
 - **Fuente:** Cliente

2. Módulo de Catálogo y Productos

- **REQ-010 - Búsqueda de Productos**
 - **Descripción:** El sistema debe permitir a los usuarios buscar productos por su nombre o categoría. La búsqueda debe mostrar resultados en menos de 3 segundos.
 - **Prioridad:** Esencial
 - **Fuente:** Cliente
- **REQ-011 - Vistas de Producto**
 - **Descripción:** El sistema debe mostrar el nombre, precio, imágenes y una descripción detallada de cada producto.
 - **Prioridad:** Esencial

- **Fuente:** Cliente
- **REQ-012 - Control de Inventario**
 - **Descripción:** El sistema debe reducir automáticamente la cantidad en el inventario cuando un producto es comprado.
 - **Prioridad:** Esencial
 - **Fuente:** Administrador

3.1 Requisitos comunes de las interfaces

3.1.1 Requisitos de la Interfaz de Usuario

- **Entradas de datos:** El sistema debe aceptar datos en los siguientes campos de entrada:
 - **Nombre de usuario:** Debe ser una cadena de texto alfanumérica, sin espacios, con una longitud mínima de 6 caracteres y máxima de 20.
 - **Correo electrónico:** Debe seguir un formato de correo electrónico válido y no debe estar registrado previamente en el sistema.
 - **Contraseña:** Debe tener una longitud mínima de 8 caracteres e incluir al menos una letra mayúscula, una letra minúscula y un número.
 - **Región y Comuna:** Ambos campos deben ser seleccionados de una lista desplegable. El campo "Comuna" dependerá de la región seleccionada.
- **Salidas del sistema:** El sistema debe proporcionar la siguiente retroalimentación al usuario:
 - **Mensajes de error:** Si un usuario no cumple con los criterios de validación (por ejemplo, una contraseña débil o un correo ya registrado), se debe mostrar un mensaje de error claro y específico debajo del campo correspondiente.
 - **Confirmación de registro:** Una vez que el registro sea exitoso, el sistema debe redirigir al usuario a la página principal (index.html) y mostrar un mensaje de confirmación.

3.1.2 Requisitos de la Interfaz de Hardware y Software

- **Compatibilidad de hardware:** La plataforma debe ser accesible desde computadoras de escritorio, laptops y dispositivos móviles (smartphones y tablets).
- **Compatibilidad de software:** El sitio web debe funcionar correctamente en los navegadores web más recientes (Chrome, Firefox, Safari, Edge y Opera).

3.1.3 Interfaces de software

1. Pasarela de Pago

- **Descripción del producto software utilizado:** La plataforma de pago que elijas (por ejemplo, PayPal, Mercado Pago, o Stripe).
- **Propósito de la interfaz:** Permite que tu plataforma envíe datos de transacciones de forma segura y reciba confirmación del pago.
- **Definición de la interfaz:**
 - **Contenido:** El sistema de The Hub enviará el total de la compra, el ID de la transacción y, posiblemente, información del cliente a la pasarela de pago. La pasarela devolverá un estado de pago (exitoso, fallido) y un token de transacción.
 - **Formato:** La comunicación se hará a través de una API (Application Programming Interface), probablemente usando el protocolo HTTPS y formatos de datos como JSON o XML.

2. Servicio de Envío

- **Descripción del producto software utilizado:** La API de un proveedor de servicios de logística (por ejemplo, un servicio de mensajería como FedEx o la empresa de correos local).
- **Propósito de la interfaz:** Permite a The Hub calcular los costos de envío y generar guías de seguimiento de paquetes.
- **Definición de la interfaz:**
 - **Contenido:** La plataforma enviará la dirección de origen y destino, las dimensiones y el peso del paquete. La API del proveedor de envíos devolverá el costo de envío y un número de seguimiento.
 - **Formato:** La integración se realizará a través de su API, con la información estructurada en un formato JSON.

3. Base de Datos de Regiones y Comunas

- **Descripción del producto software utilizado:** Una base de datos o servicio externo que contiene la lista de todas las regiones y comunas de tu país.
- **Propósito de la interfaz:** Proporciona los datos necesarios para que los campos de "Región" y "Comuna" en el formulario de registro se autocompleten y validen correctamente.
- **Definición de la interfaz:**

- **Contenido:** La plataforma solicitará la lista de regiones o las comunas de una región específica. La API devolverá un listado de nombres y sus IDs correspondientes.
- **Formato:** La comunicación se hará mediante llamadas a una API REST que devuelve datos en formato JSON.

3.1.4 Interfaces de comunicación

- **Protocolo HTTP/HTTPS:**
 - **Propósito:** Es el protocolo fundamental para la comunicación entre el navegador del usuario y tu servidor.
 - **Requisito:** Todas las comunicaciones de la plataforma, especialmente las páginas de inicio de sesión, registro, carrito de compras y pago, deben usar el protocolo **HTTPS** para garantizar que los datos (como contraseñas e información de tarjetas de crédito) viajen encriptados y de forma segura. El uso de **HTTP** sin encriptación está prohibido en áreas sensibles.
- **Protocolo de API REST:**
 - **Propósito:** Este es el protocolo que se utiliza para que tu sistema se comunique con servicios externos, como las pasarelas de pago y los proveedores de servicios de envío.
 - **Requisito:** La comunicación con la API de terceros (por ejemplo, para procesar pagos o calcular costos de envío) debe realizarse a través de llamadas RESTFUL utilizando el formato de datos JSON.

3.2 Requisitos funcionales

3.2.1 Requisito Funcional 1: Búsqueda de Productos

- **Requerimiento funcional:** El sistema debe permitir a los usuarios buscar productos.
- **Actor:** Usuario Cliente.
- **Descripción:** El usuario debe poder ingresar un término de búsqueda en la barra de búsqueda. El sistema debe mostrar una lista de productos que coincidan con el término, ordenados por relevancia. Si no hay resultados, debe mostrar un mensaje claro informando al usuario.

3.2.2 Requisito Funcional 2: Agregar Producto al Carrito

- **Requerimiento funcional:** El sistema debe permitir a los usuarios agregar productos al carrito de compras.
- **Actor:** Usuario Cliente.
- **Descripción:** En la página de cada producto, debe haber un botón "Agregar al carrito". Al hacer clic, el sistema debe añadir el producto al carrito de compras y mostrar una notificación de éxito. El carrito debe actualizarse para reflejar el nuevo total de productos.

3.2.3 Requisito Funcional 3: Gestión de Inventario

- **Requerimiento funcional:** El sistema debe permitir al administrador gestionar el inventario.
- **Actor:** Usuario Administrador.
- **Descripción:** En el panel de administración, el usuario debe poder ver la cantidad actual de cada producto en inventario. Al realizar una venta, el inventario de los productos comprados debe disminuir automáticamente. El sistema debe alertar al administrador cuando el stock de un producto llegue a un umbral bajo (ej. menos de 5 unidades).

3.3 Requisitos no funcionales

- Usabilidad

- El sitio web debe ser intuitivo y fácil de usar, permitiendo a los clientes completar una compra en un máximo de 5 pasos desde la página del producto.
- Los formularios deben proporcionar mensajes de error claros y específicos para guiar al usuario a corregir sus entradas.

- Rendimiento

- El tiempo de carga de las páginas (página de inicio, catálogo de productos) no debe exceder los 3 segundos, incluso con un alto volumen de usuarios.
- El sistema debe ser capaz de procesar al menos 100 transacciones simultáneas sin degradación del rendimiento.

- Seguridad

- Toda la comunicación entre el navegador del usuario y el servidor debe estar encriptada con HTTPS.
- Las contraseñas de los usuarios deben ser almacenadas de forma segura (encriptadas o con hash) y no como texto plano.

- El sistema debe protegerse de vulnerabilidades comunes como inyección SQL y cross-site scripting (XSS).
- El acceso al panel de administración debe requerir una autenticación robusta y ser exclusivo para el personal autorizado.

- **Disponibilidad**

- El sitio web debe estar disponible el 99.9% del tiempo, excluyendo periodos de mantenimiento programado.
- En caso de fallo, el sistema debe tener un plan de recuperación que permita restaurar el servicio en un máximo de 2 horas.

- **Mantenibilidad**

- El código fuente del sistema debe ser modular y estar bien documentado, facilitando las futuras actualizaciones y correcciones.
- La base de datos debe estar diseñada de manera que permita añadir nuevas categorías de productos sin necesidad de reestructurar todo el sistema.

3.3.1 Requisitos de rendimiento

1. Tiempos de Respuesta

- El 95% de las páginas de productos y categorías deben cargarse en menos de 2 segundos.
- El proceso de pago completo, desde que el usuario hace clic en "Pagar" hasta la confirmación de la transacción, debe completarse en menos de 5 segundos.
- La funcionalidad de búsqueda de productos debe mostrar resultados en menos de **1.5 segundos**.

2. Capacidad y Carga

- El sistema debe ser capaz de soportar hasta 100 usuarios simultáneos navegando y realizando compras sin experimentar una degradación notable del rendimiento.
- La plataforma debe poder procesar al menos 30 transacciones de pago por minuto durante los periodos de mayor tráfico (ej. campañas de ofertas).

3. Escalabilidad

- El sistema debe ser escalable para manejar un aumento del 50% en el tráfico de usuarios sin requerir cambios significativos en la arquitectura del software.

4. Disponibilidad

- El sitio web debe tener una disponibilidad del 99.9% del tiempo, excluyendo los periodos de mantenimiento previamente programados.

3.3.2 Seguridad

1. Protección de Datos y Cuentas

- **Cifrado de Contraseñas:** Las contraseñas de los usuarios deben ser almacenadas utilizando técnicas de hash criptográfico. Bajo ninguna circunstancia deben almacenarse como texto plano.
- **Seguridad de Transmisión:** Toda la comunicación entre el navegador del usuario y el servidor (incluyendo el proceso de registro, inicio de sesión y pago) debe ser encriptada mediante el protocolo HTTPS.
- **Protección contra Ataques Comunes:** El sistema debe implementar protecciones para prevenir ataques de inyección de código (SQL Injection), cross-site scripting (XSS) y falsificación de peticiones de sitios cruzados (CSRF).

2. Acceso y Auditoría

- **Autenticación de Administradores:** El acceso al panel de administración debe requerir credenciales válidas y ser restringido únicamente a usuarios autorizados. Se debe implementar un sistema de bloqueo de cuentas tras varios intentos fallidos de inicio de sesión.
- **Registro de Actividad (Logs):** El sistema debe registrar un historial detallado de todas las acciones críticas realizadas por los administradores (ej. creación, modificación o eliminación de productos, cambios en el inventario o precios).
- **Control de Permisos:** Las funcionalidades del panel de administración deben estar asignadas a diferentes roles de usuario para asegurar que cada empleado solo pueda acceder a las tareas necesarias para su trabajo.

3. Integridad del Sistema

- **Validación de Datos:** El sistema debe validar todas las entradas de datos (inputs) tanto en el cliente (lado del navegador) como en el servidor. Esto es crucial para asegurar la integridad de la información.

- **Manejo de Errores:** En caso de errores, el sistema no debe mostrar información sensible de la base de datos o de la configuración del servidor al usuario. Los mensajes de error deben ser genéricos y fáciles de entender.

3.3.3 Fiabilidad

1. Disponibilidad (Uptime)

- **Tiempo de actividad:** El sistema debe tener una disponibilidad del **99.9%**. Esto equivale a un máximo de 8.76 horas de inactividad por año.
- **Intervalo de fallos:** El tiempo medio entre fallas del sistema no debe ser inferior a 120 días.

2. Recuperación

- **Tiempo de recuperación:** En caso de un fallo inesperado, el sistema debe ser capaz de ser restaurado y estar completamente operativo en un máximo de **2 horas**.
- **Integridad de los datos:** En caso de una falla del sistema, no se debe perder la información de las transacciones ya completadas. El sistema debe mantener la consistencia de los datos en todo momento.

3. Tolerancia a fallos

- **Manejo de errores externos:** Si una dependencia externa (como la API de la pasarela de pago o el servicio de envíos) falla, el sistema debe manejar el error de forma controlada, informando al usuario de la situación y evitando que el fallo afecte a otras partes del sistema. Por ejemplo, si el servicio de envío no responde, se debe mostrar un mensaje al usuario, pero permitirle seguir con la compra.

3.3.4 Disponibilidad

- El sistema debe tener una disponibilidad operativa del **99.9%** a lo largo del año. Este porcentaje excluye los tiempos de inactividad planificados para mantenimiento y actualizaciones.
- El tiempo máximo de inactividad del sistema (downtime) no debe exceder de 8.76 horas al año. Esto asegura que la plataforma esté casi siempre accesible para los clientes de The Hub.
- El sistema debe ser tolerante a fallos, lo que significa que un error en una parte del sistema (por ejemplo, en la conexión con el servicio de envíos) no debe causar una falla total de la plataforma.

- El sistema debe poder recuperar los servicios esenciales en un tiempo máximo de **2 horas** en caso de una interrupción inesperada. Esto incluye la capacidad de restaurar bases de datos y configuraciones a su estado más reciente.

3.3.5 Mantenibilidad

1. Tipo de Mantenimiento

- **Mantenimiento Correctivo:** Para corregir defectos o errores que se descubran en el sistema.
- **Mantenimiento Adaptativo:** Para adaptar el sistema a los cambios en el entorno de software (por ejemplo, nuevas versiones de navegadores web o APIs de pasarelas de pago).
- **Mantenimiento Perfectivo:** Para mejorar la funcionalidad, el rendimiento o la usabilidad del sistema.
- **Mantenimiento Preventivo:** Para realizar tareas de rutina (como actualizaciones de seguridad o limpieza de la base de datos) que prevengan futuros problemas.

2. Roles y Responsabilidades

- **¿Quién realiza el mantenimiento?** Las tareas de mantenimiento serán realizadas por un equipo de desarrolladores designado y no por los usuarios finales. Este equipo será responsable de la implementación y las pruebas de todas las actualizaciones.
- **Notificación al usuario:** Las tareas de mantenimiento que requieran un tiempo de inactividad deben ser programadas con anticipación y los usuarios deben ser notificados a través de un aviso en la página principal del sitio web.

3. Programación de Tareas de Mantenimiento

- **Frecuencia:** Las tareas de mantenimiento preventivo, como la generación de estadísticas de acceso, se realizarán de manera semanal y mensual.
- **Revisiones:** Se realizará una revisión completa del código y la arquitectura del sistema cada seis meses para identificar áreas de mejora y deuda técnica.
- **Actualizaciones:** Las actualizaciones menores se implementarán de forma regular, mientras que las versiones más grandes (con nuevas funcionalidades) se planificarán con base en el roadmap de Requisitos Futuros

3.3.6 Portabilidad

- **Lenguaje de Programación:** Para maximizar la portabilidad, el sistema debe ser desarrollado en un lenguaje de programación de uso común y multiplataforma, como Python, PHP o JavaScript.
- **Dependencia del Servidor:** El porcentaje de código dependiente del servidor debe ser mínimo para permitir una fácil migración a diferentes proveedores de hosting o servicios de nube. La arquitectura del sistema debe ser lo más independiente posible del servidor subyacente.
- **Contenedores:** Se recomienda que la aplicación sea empaquetada en un contenedor (como Docker) para garantizar que el entorno de ejecución sea idéntico en desarrollo, pruebas y producción, facilitando el despliegue en cualquier plataforma que soporte contenedores.
- **Sistema Operativo:** El sistema no debe depender de un sistema operativo específico (como Windows o Linux) para su funcionamiento, permitiendo su despliegue en la mayoría de los entornos de servidor.