

TRABAJO DE GRADO  
Opción Seminario-Diplomado.

## PLAN DE PRUEBAS TIENDA ONLINE NEPTUNE



Corporación Universitaria Remington.

Facultad de Ingenierías

Especialización de seguridad de la información- Pregrado ingeniería de sistemas

Autores:

Carlos Stiwar Asprilla Mosquera

Camilo Ramírez Angarita

Janier Colorado Londoño

Tutor: Fabio León Grisales Morales

Opción de Trabajo de grado Seminario-Diplomado.

2025

## **Dedicatoria**

Este trabajo está dedicado a todas las personas que valoran la calidad del software como un pilar esencial en la creación de sistemas confiables, seguros y centrados en el usuario. En particular, a aquellos que comprenden que detrás de cada aplicación que funciona hay un proceso riguroso de análisis, validación y mejora continua.

### **Agradecimientos**

Este trabajo fue posible gracias al entorno académico del seminario, que permitió consolidar los conocimientos teóricos y prácticos relacionados con el proceso de pruebas de software. También quiero reconocer el valioso aporte del trabajo en equipo y el intercambio de ideas con los participantes del seminario, elementos que facilitaron la aplicación de lo aprendido en el análisis y ejecución de un plan de pruebas para una tienda online, fortaleciendo así una visión crítica y estructurada del aseguramiento de la calidad del software.

## Tabla de Contenidos

Resumen.....	5
1. Marco contextual .....	7
1.1 Contexto del seminario .....	7
1.2 Contexto del proyecto de software .....	7
1.3 Descripción de la tienda online objeto de pruebas.....	8
2. Marco conceptual.....	9
3. Plan de pruebas del proyecto .....	12
3.1 Análisis de riesgo .....	12
3.2 Alcance de las pruebas .....	19
3.3 Exclusiones del plan de pruebas .....	22
3.4 Estrategias de pruebas .....	24
3.5 Estimación del proyecto.....	27
3.6 Flujo de pruebas .....	32
3.7 Tipos de pruebas a ejecutar .....	32
3.8 Precondiciones .....	33
3.9 Casos de prueba .....	34
3.10 Evidencias de ejecución .....	40
3.11 Registro y gestión de defectos (Bug Tracker).....	47
4. Desarrollo e implementación del aprendizaje.....	51
4.1 Aplicación de los conocimientos adquiridos .....	51
4.2 Resultados obtenidos del proceso de pruebas .....	52
4.3 Análisis de resultados .....	53
4.4 Discusión técnica de los hallazgos.....	54
5 Informe de cierre.....	55
5.1 Alcance estimado vs alcance real .....	55
5.2 Riesgos.....	56
5.3 Métricas del proceso de pruebas .....	58
5.4 Tiempos del proyecto.....	59
5.5 Lecciones Aprendidas .....	60
5.6 Oportunidades de Mejora.....	61
5.7 Cierre orientado al cliente .....	62
6. Conclusiones .....	63
Referencias.....	64

## Resumen

El presente documento describe el proceso de elaboración y aplicación de un plan de pruebas de software para NepTune, una tienda online desarrollada bajo un modelo de tipo marketplace. El trabajo se fundamenta en la identificación y análisis de riesgos asociados a los componentes críticos del sistema, con el propósito de priorizar las actividades de prueba y orientar el esfuerzo de aseguramiento de la calidad hacia los escenarios con mayor impacto sobre la experiencia del usuario y la estabilidad operativa de la aplicación.

Dentro de los riesgos más relevantes se identificaron aquellos relacionados con la integración de pasarelas de pago, la protección de datos sensibles de los usuarios y los posibles retrasos en los procesos administrativos de validación de productos. A partir de este análisis se definieron las estrategias de prueba más adecuadas, combinando enfoques analíticos y basados en modelos, junto con una estrategia de regresión controlada, lo que permitió estructurar un proceso de pruebas coherente y alineado con las necesidades del negocio y la arquitectura del sistema.

Posteriormente, se estableció el alcance del plan de pruebas, delimitando de manera precisa los módulos y funcionalidades a evaluar. Este alcance incluyó la validación del flujo de compra y pago, la revisión de la seguridad de la información, el proceso de aprobación de productos por parte del administrador y la ejecución de pruebas sobre las historias de usuario HU-6 y HU-10, así como una regresión focalizada en los componentes impactados. Los elementos excluidos fueron definidos explícitamente con el fin de optimizar recursos y mantener un enfoque claro durante la ejecución.

Con base en lo anterior, se diseñaron y documentaron los casos de prueba, estructurados a partir de objetivos, precondiciones, pasos y resultados esperados, orientados a cubrir los riesgos identificados y validar los criterios funcionales establecidos. Finalmente, el informe consolida los resultados obtenidos, las evidencias recolectadas y las recomendaciones derivadas del proceso, proporcionando una visión técnica del estado de calidad del sistema y aportando insumos para su mejora continua.

**Palabras clave:** marketplace, plan de pruebas, historias de usuario, casos de prueba, análisis de riesgos, seguridad de la información.

## **1. Marco contextual**

### **1.1 Contexto del seminario**

El presente informe técnico se desarrolla en el marco del seminario orientado al aseguramiento de la calidad del software, cuyo propósito es fortalecer la comprensión y aplicación práctica de los conceptos fundamentales relacionados con el análisis, diseño y ejecución de pruebas en sistemas informáticos. Durante el seminario se abordaron contenidos teóricos y metodológicos que permiten comprender el rol del testing como un proceso clave dentro del ciclo de vida del software, haciendo énfasis en la gestión de riesgos, la definición de estrategias de prueba y la adaptación del proceso a distintos contextos de desarrollo.

Como parte del proceso formativo, el seminario propuso la aplicación de los conocimientos adquiridos a un caso práctico, con el fin de simular un entorno de proyecto real y favorecer el desarrollo de habilidades analíticas y técnicas propias del rol de aseguramiento de la calidad. Este enfoque permitió integrar los conceptos vistos en clase dentro de un ejercicio estructurado y orientado a resultados.

### **1.2 Contexto del proyecto de software**

El proyecto de software seleccionado como caso de estudio corresponde a NepTune, una tienda online diseñada bajo un modelo de tipo marketplace, en el cual múltiples proveedores pueden publicar y gestionar sus productos dentro de una plataforma común, sin necesidad de contar con una infraestructura propia de comercio electrónico. Aunque se trata de una organización digital ficticia, el proyecto se encuentra estructurado conforme a las dinámicas reales del comercio electrónico actual.

NepTune integra funcionalidades esenciales para la operación de una tienda online, tales como la gestión del catálogo de productos, el flujo de compra y pago, la comunicación con pasarelas de pago externas, la administración de información transaccional y la generación de comprobantes para los usuarios. Estas características reflejan los desafíos técnicos y operativos habituales de los sistemas de comercio digital, donde la estabilidad, la seguridad y la confiabilidad del software son factores críticos para el negocio.

### **1.3 Descripción de la tienda online objeto de pruebas**

La tienda online objeto de pruebas se caracteriza por la interacción entre diferentes actores del sistema, incluyendo usuarios finales, proveedores y administradores de la plataforma. Desde el punto de vista funcional, el sistema permite a los usuarios navegar por el catálogo, seleccionar productos, realizar compras y efectuar pagos a través de pasarelas externas, mientras que los proveedores gestionan su oferta y el administrador valida los productos antes de su publicación.

Adicionalmente, la plataforma maneja información sensible asociada a datos personales y transacciones, y depende de servicios externos para la ejecución de procesos críticos, como el procesamiento de pagos y el envío de notificaciones. Estas condiciones incrementan la exposición a riesgos técnicos y operativos, lo que hace indispensable la aplicación de un proceso de pruebas de software orientado a identificar fallos, prevenir errores críticos y garantizar la integridad de los flujos de negocio. En este contexto, el desarrollo de un plan de pruebas estructurado resulta fundamental para evaluar el estado de calidad del sistema y apoyar la toma de decisiones orientadas a su mejora continua.

## **2. Marco conceptual**

Las bases teóricas del proceso de aseguramiento de la calidad aplicado en el proyecto parten de los conceptos fundamentales que orientan el testing en el desarrollo de software. En primer lugar, las pruebas de software se definen como el proceso de evaluar y verificar que un producto o aplicación cumpla con los requisitos establecidos y funcione de acuerdo con lo esperado. Este proceso resulta esencial para la prevención de errores, la mejora del rendimiento y la garantía de la calidad del software antes de su puesta en producción (IBM, s. f.).

Dentro del proceso de pruebas, es importante reconocer la existencia de diferentes tipos de pruebas, cada una con objetivos específicos. Las pruebas unitarias permiten validar componentes individuales del sistema, mientras que las pruebas de integración verifican que los distintos módulos interactúen correctamente. Por su parte, las pruebas funcionales se enfocan en confirmar que las funcionalidades cumplan con los requisitos definidos, las pruebas de rendimiento evalúan el comportamiento del software bajo distintas cargas, las pruebas de seguridad identifican posibles vulnerabilidades y las pruebas de usabilidad miden la experiencia del usuario final (IBM, s. f.).

A partir de este fundamento, surge la necesidad de identificar los elementos que podrían afectar negativamente el sistema, lo que conduce al concepto de análisis de riesgos. Este se entiende como el proceso orientado a identificar eventos o situaciones que puedan impactar de manera adversa el funcionamiento de una organización, el desarrollo de una actividad o las condiciones de un entorno determinado, con el fin de

establecer medidas preventivas y planes de contingencia frente a posibles impactos negativos (Porto, 2023). Con base en este análisis, se define la estrategia de pruebas, concebida como un conjunto de definiciones de alto nivel que orientan la forma en que se llevarán a cabo los procesos de prueba y que apoyan la toma de decisiones relacionadas con la asignación de recursos y esfuerzos de testing (Abstracta, 2024).

Para aplicar estas estrategias dentro del desarrollo del proyecto, se emplean las historias de usuario como un insumo clave. Estas se describen como una representación general e informal de una funcionalidad del software escrita desde la perspectiva del usuario final, cuyo propósito es expresar el valor que una determinada característica aporta al cliente o al negocio (Atlassian, s. f.).

Una vez definidos los requisitos funcionales a través de las historias de usuario, se procede al diseño de los casos de prueba. Un caso de prueba corresponde a una secuencia estructurada de acciones o condiciones que se plantean con el fin de verificar la eficacia de un programa o aplicación de software. Su objetivo es confirmar que el sistema funcione conforme a lo planificado y que sus funcionalidades cumplan con los criterios establecidos (web.dev, 2023).

Adicionalmente, cuando se realizan cambios en el sistema, como correcciones de errores, mejoras o incorporación de nuevas funcionalidades, se aplican las pruebas de regresión. Este tipo de pruebas se orienta a verificar que dichas modificaciones no afecten negativamente las funcionalidades existentes y que el sistema continúe respondiendo de acuerdo con lo esperado después de una actualización o nueva versión (QAlified, s. f.).

En el contexto del comercio electrónico, resulta necesario incorporar el concepto de pasarela de pago, entendida como un servicio que permite autorizar y procesar pagos mediante tarjetas de débito o crédito, tanto en entornos online como offline. En las tiendas virtuales, las pasarelas de pago garantizan un flujo de pago seguro mediante la encriptación de la información financiera del cliente antes de su transferencia a la cuenta del comerciante (Boada & Boada, 2024).

Finalmente, dado que el proyecto implica el tratamiento de información personal asociada a transacciones comerciales, es indispensable considerar el concepto de datos sensibles. Estos corresponden a aquellos datos que afectan la intimidad de su titular o cuyo uso indebido puede generar discriminación o vulnerar derechos fundamentales, por lo cual requieren un tratamiento especial y medidas de protección reforzadas (Comisión de la Verdad, s. f.).

### 3. Plan de pruebas del proyecto

#### 3.1 Análisis de riesgo

Con el fin de orientar adecuadamente el proceso de pruebas y priorizar los componentes más sensibles del sistema, se llevó a cabo un análisis de riesgos que permitió identificar los eventos, condiciones o fallos potenciales que podrían afectar negativamente el funcionamiento de la plataforma NepTune. Este análisis constituye un insumo fundamental para la definición del alcance, la selección de las estrategias de prueba y la distribución de los esfuerzos dentro del proceso de aseguramiento de calidad.

La matriz presentada a continuación resume los riesgos detectados, su nivel de impacto, la probabilidad de ocurrencia y las acciones de tratamiento consideradas. Este instrumento facilita una comprensión clara de los puntos críticos del sistema y respalda la toma de decisiones frente a los elementos que deben evaluarse con mayor rigurosidad durante la ejecución del plan de pruebas.

*Tabla 1*

*Matriz de riesgo marketplace NepTune*

MATRIZ DE RIESGOS MARKETPLACE NEPTUNE					
RIESGO	PROBABILIDAD	IMPACTO	VALOR DEL RIESGO	NIVEL DEL RIESGO	PLAN CONTINGENCIA
Fallo en la integración de pasarelas de pago	4	5	20	Muy grave	Definir un plan de rollback y tener una pasarela secundaria lista. Asegurar pruebas de volumen y estrés específicas en el módulo de pago.

Incumplimiento de requisitos de responsive design	3	3	9	Importante	Crear prototipos de alta fidelidad para dispositivos móviles y ejecutar pruebas de usabilidad en una amplia variedad de dispositivos desde el inicio de cada sprint.
Fuga o exposición de datos sensibles (clientes/pagos)	2	5	10	Importante	Realizar auditorías de seguridad (penetration testing) en el entorno de producción. Encriptar todos los datos sensibles en la Base de Datos.
Retraso en la validación de productos por el Administrador	3	4	12	Importante	Establecer un Acuerdo de Nivel de Servicio (SLA) para el tiempo de validación. Si se excede, el sistema debe escalar la alerta al supervisor del Administrador.
Disminución de rendimiento o lentitud con alto tráfico	3	4	12	Importante	Implementar pruebas de carga y estrés. Usar caching agresivo para el catálogo de productos (UH-U-01), optimizar consultas a la Base de Datos e Implementar balanceadores de cargas y autoescalado de servicios en el hosting según el tráfico.

Error de conciliación entre la Base de Datos y el Banco	3	4	12	Importante	Implementar un mecanismo de logs detallado de cada transacción y un proceso de revisión automatizado diario que compare los saldos.
El envío de correo de confirmación a no-Gmail falla	4	3	12	Importante	Añadir un mensaje visible al usuario que indique la limitación del servicio de correo. Ofrecer una alternativa (ej. WhatsApp/SMS) o un PDF de descarga directa.
Falta de claridad en la descripción de los productos por el Proveedor	4	2	8	Apreciable	Establecer directrices de calidad y longitud mínima/máxima en el formulario de carga del Proveedor. Implementar revisión semi-automatizada de formato.
UH-U-01 - La aplicación no muestra exactamente 10 productos por página o el filtro por sexo/tipo de artículo falla, causando una visualización incompleta/incorrecta.	3	3	9	Importante	Implementar pruebas unitarias y de integración que verifiquen rigurosamente las consultas a la base de datos para paginación y filtrado.

<p>UH-U-02</p> <p>- La información esencial del producto (nombre, descripción, precio, talla y stock) no se carga correctamente o el botón de compra aparece habilitado sin que el cliente haya iniciado sesión.</p>	3	3	9	Importante	<p>Realiza una validación de estado de sesión (check de login) antes de renderizar el botón de "Suscribir" o "Comprar". Usar mocks de datos en las pruebas para asegurar que todos los campos (precio, talla, stock) se cargan y tienen formato correcto.</p>
<p>UH-U-03</p> <p>- Los datos del cliente no se almacenan correctamente en la base de datos después del registro, o el sistema permite el registro con datos inválidos/incompletos.</p>	2	4	8	Apreciable	<p>Implementar validaciones front-end y back-end robustas (ej. formato de correo, longitud de contraseña).</p>
<p>UH-U-04</p> <p>- Al iniciar sesión, la redirección a la página principal falla o al cerrar sesión, los datos de la sesión anterior persisten por caché o error de eliminación.</p>	3	3	9	Importante	<p>Reforzar el proceso de logout asegurando la invalidación inmediata de tokens y la limpieza de la caché de sesión (cookies).</p>
<p>UH-U-05</p> <p>- El sistema permite la suscripción sin que el cliente haya iniciado sesión previamente, o el cliente no es redirigido correctamente a la página de suscripciones.</p>	3	2	6	Apreciable	<p>La lógica de suscripción debe tener una dependencia estricta del token de sesión activo.</p>

<p>UH-U-06</p> <p>- El cálculo de la cantidad total a pagar es incorrecto o se permite añadir productos al carrito sin iniciar sesión.</p>	3	4	12	Importante	<p>El cálculo total debe realizarse siempre en el back-end para evitar manipulación client-side. Bloquear la función "Añadir al Carrito" si la API de autenticación no confirma una sesión activa.</p>
<p>UH-U-07</p> <p>- El sistema no permite aplicar vales de descuento correctamente o los datos personales/de envío verificados no se guardan para el siguiente paso.</p>	3	4	12	Importante	<p>Crear un servicio independiente de validación de cupones con pruebas exhaustivas sobre expiración, uso único y aplicación de descuentos. Almacenar los datos de envío en una variable temporal de sesión inmediatamente antes de avanzar al pago.</p>
<p>UH-U-08</p> <p>- La confirmación mostrada al cliente no refleja con exactitud los datos y productos introducidos en los pasos anteriores del flujo de compra.</p>	2	3	6	Apreciable	<p>El endpoint de confirmación debe volver a consultar la orden final almacenada en la base de datos (DB) y no depender de variables de sesión o front-end para mostrar los datos definitivos.</p>

UH-U-09 - El sistema falla en generar los vales promocionales, incluso si la compra supera la cantidad mínima requerida, o no se muestra el código de compra.	3	4	12	Importante	La lógica de generación de vales debe ser atómica y ejecutarse inmediatamente después del éxito del pago.
UH-U-10 - El correo de confirmación de compra no se envía a una cuenta Gmail o, de enviarse, el PDF adjunto omite información clave como el vale descuento o el código de compra/expedición.	4	4	16	Muy grave	Implementar un servicio de queue (cola de mensajes) para el envío de correos, asegurando el reintento si el primer intento falla.
UH-U-11 - Se permite visualizar la contraseña del usuario o se omite algún otro dato importante almacenado.	2	3	6	Apreciable	Excluir explícitamente el campo contraseña (password) de cualquier consulta a la DB para la visualización del perfil.
UH-U-12 - El sistema no actualiza los datos del cliente en la Base de Datos o la edición de datos falla con mensajes de error confusos.	3	2	6	Apreciable	Implementar pruebas que editen y verifiquen inmediatamente la persistencia del cambio.
UH-U-13 - El cálculo del dinero total gastado en la aplicación es incorrecto o no se pueden visualizar los productos y	3	3	9	Importante	El cálculo del total gastado debe basarse en la suma de los precios finales registrados en los históricos de órdenes

precios de compras anteriores.					completadas, no en los precios actuales del catálogo.
UH-U-14 - El proceso de recuperación de contraseña falla en enviar una nueva contraseña al cliente, impidiendo su acceso al sistema.	3	4	12	Importante	Utilizar un servicio de terceros de confianza para el envío de correos de recuperación que garantice la alta disponibilidad.
Probabilidad que surja la caída del servidor web	3	4	12	Importante	Realizar la activación del failover y restauración de servicios.

En la tabla 2 se representa una matriz de evaluación de riesgos que combina dos dimensiones fundamentales: la probabilidad de ocurrencia y el impacto o gravedad que tendría cada riesgo en caso de materializarse. Su objetivo es facilitar la clasificación de los riesgos y permitir al evaluador determinar cuáles requieren mayor atención durante el proceso de pruebas. A partir de la relación entre estas dos variables, es posible establecer niveles de prioridad que orientan la planificación y ejecución de las actividades de testing. Esta matriz contribuye a una asignación más eficiente de los recursos disponibles, enfocando los esfuerzos en aquellos riesgos con mayor probabilidad de afectar la operación del sistema o la experiencia del usuario. Asimismo, sirve como herramienta de apoyo para la toma de decisiones, al proporcionar una visión estructurada y comparativa del perfil de riesgo del proyecto.

Tabla 2

Matriz de evaluación de riesgos

			IMPACTO (GRAVEDAD)				
			MUY BAJO 1	BAJO 2	MEDIO 3	ALTO 4	MUY ALTO 5
PROBABILIDAD (OCURRENCIA)	MUY ALTA	5	5	10	15	20	25
	ALTA	4	4	8	12	16	20
	MEDIA	3	3	6	9	12	15
	BAJA	2	2	4	6	8	12
	MUY BAJA	1	1	2	3	4	5

### 3.2 Alcance de las pruebas

Se definió a partir del análisis de riesgos, las necesidades específicas de la plataforma NepTune y las funcionalidades críticas identificadas en las historias de usuario. Este alcance establece los componentes, flujos y elementos del sistema que serán verificados durante el proceso de testing, priorizando aquellos cuya falla podría afectar la estabilidad del sistema, la seguridad de la información o la experiencia del usuario final. Asimismo, permite delimitar claramente los límites del proceso de pruebas, evitando la dispersión de esfuerzos y facilitando una planificación más eficiente de las actividades. La definición del alcance también contribuye a la correcta asignación de recursos y a la selección de estrategias de prueba acordes con el nivel de criticidad de cada

funcionalidad. De esta manera, se busca obtener una evaluación objetiva del estado de calidad del sistema dentro de los tiempos y condiciones establecidos para el proyecto.

En este proyecto, las pruebas se concentrarán exclusivamente en los siguientes aspectos:

1. Integración con pasarelas de pago

Se validará el funcionamiento completo del flujo de pago, evaluando la comunicación entre la plataforma y la pasarela externa. Esto incluye la correcta respuesta ante pagos aprobados, rechazados, errores, tiempos de espera y respuestas inesperadas, asegurando que el estado del pedido y el inventario no se vean afectados ante fallos de la pasarela.

2. Protección y manejo de datos sensibles

Se verificará que la plataforma no exponga información sensible del usuario o del método de pago, tanto en el front-end como en las respuestas de la API. La validación incluye la revisión de cifrado, enmascaramiento y ausencia de datos en texto plano durante el proceso transaccional.

3. Validación de productos por el administrador

Se revisará el proceso mediante el cual el administrador aprueba, rechaza o gestiona el estado de los productos enviados por los proveedores. El enfoque estará en garantizar la correcta actualización del estado, la ausencia de bloqueos y la consistencia entre la interfaz y la base de datos.

4. Historias de usuario HU-6 y HU-10

Se implementarán pruebas específicas correspondientes a estas dos historias de usuario, verificando sus criterios de aceptación y asegurando la correcta funcionalidad en los flujos asociados al carrito de compras (HU-6) y al envío de correo de confirmación con PDF (HU-10).

UH-U-06			
<b>IDENTIFICADOR:</b>	UH-U-06	<b>TÍTULO:</b>	Carrito
<b>VALOR:</b>	100	<b>ESTIMACIÓN:</b>	5
<b>DEPENDENCIAS:</b>	UH-U-02, UH-U-03 y UH-U-04	<b>DESCRIPCIÓN:</b>	Como cliente, quiero ver mi carrito para poder ver los productos que voy a comprar.
<b>CONFIRMACIONES:</b>			
<ul style="list-style-type: none"> <li>~ Para añadir productos al carrito se debe iniciar sesión.</li> <li>~ Se puede visualizar información acerca de los productos (Nombre, Descripción y Precio).</li> <li>~ Se puede cancelar el producto si no se desea comprar.</li> <li>~ Se puede ver la cantidad total a pagar.</li> </ul>			

*Figura 1 – Historia de usuario UH-U-06*

UH-U-10			
<b>IDENTIFICADOR:</b>	UH-U-10	<b>TÍTULO:</b>	Mail
<b>VALOR:</b>	100	<b>ESTIMACIÓN:</b>	8
<b>DEPENDENCIAS:</b>	UH-U-03, UH-U-04 y UH-U-09	<b>DESCRIPCIÓN:</b>	Como cliente, quiero que me llegue un mensaje con la compra que acabo de realizar al correo para poder verificarla.
<b>CONFIRMACIONES:</b>			
<ul style="list-style-type: none"> <li>~ Solo funciona con cuentas Gmail.</li> <li>~ En el PDF, se puede visualizar los productos comprados con su información respectiva. También se puede visualizar el código de compra, el vale descuento y el de expedición.</li> </ul>			

*Figura 2 – Historia de usuario UH-U-10*

## 5. Pruebas de regresión acotada

La regresión se enfocará únicamente en los componentes directamente afectados por los cambios asociados a los puntos anteriores. Esto incluye el flujo de pago, la validación de productos, la consulta y actualización del estado del pedido y todas aquellas funcionalidades derivadas de HU-6 y HU-10. Esta regresión será

limitada para evitar la ejecución de pruebas que se encuentran fuera del alcance del proyecto, manteniendo el enfoque en los módulos con mayor nivel de riesgo e impacto.

### **3.3 Exclusiones del plan de pruebas**

Con el fin de mantener un enfoque preciso y garantizar la revisión de los componentes más críticos de la plataforma NepTune, se definieron las funcionalidades, módulos y actividades que quedan explícitamente fuera del alcance del presente plan de pruebas. Estas exclusiones permiten evitar dispersión, optimizar los recursos y asegurar que los esfuerzos se dirijan hacia las áreas con mayor impacto sobre la operación del sistema.

Las siguientes actividades y funcionalidades no serán evaluadas durante este proceso:

1. Pruebas completas del catálogo de productos

La recolección, edición, filtrado avanzado, búsqueda por categorías o visualización general del catálogo no se incluirán en este ciclo, salvo los elementos estrictamente necesarios para validar el flujo del administrador.

2. Funcionalidades no asociadas al flujo de compra

Se excluyen comprobaciones sobre la gestión de usuarios, la creación o recuperación de cuentas, la actualización de perfiles, y los módulos informativos o estáticos de la plataforma.

3. Pruebas profundas de rendimiento y estrés

Evaluaciones relacionadas con comportamiento bajo carga, concurrencia, pruebas volumétricas o simulación de múltiples transacciones simultáneas no se evaluarán.

4. Evaluaciones completas de interfaz y diseño

El análisis detallado de usabilidad, accesibilidad o consistencia visual del sistema no se abordará, exceptuando los puntos mínimos necesarios para ejecutar los casos de prueba funcionales.

5. Pruebas de integraciones externas no relacionadas con pagos

Servicios secundarios, APIs complementarias o módulos externos distintos de la pasarela de pago no serán evaluados en esta fase.

6. Validaciones de seguridad avanzadas

No se realizarán pruebas de penetración, análisis exhaustivo de vulnerabilidades, escaneo automatizado ni explotación de fallos. Solo se validará la no exposición de datos sensibles durante el flujo transaccional.

7. Pruebas generales del flujo de carritos, órdenes o notificaciones

El testeo del carrito, de las órdenes y del sistema de notificaciones se limitará exclusivamente a lo cubierto por HU-6, HU-10 y los módulos afectados por la regresión acotada. Todo lo demás será omitido.

8. Revisión completa de procesos administrativos

Se excluyen módulos adicionales del panel de administración, tales como gestión de usuarios internos, reportes, métricas o administración global del marketplace.

#### 9. Configuraciones del entorno o infraestructura

Instalación de servidores, ajustes de hosting, monitoreo de infraestructura o configuraciones de base de datos no forman parte del alcance de las pruebas.

#### 10. Funcionalidades futuras o no entregadas

Cualquier componente en desarrollo, incompleto o no entregado por el equipo de construcción quedará fuera de esta fase de pruebas.

### **3.4 Estrategias de pruebas**

Para garantizar que el proceso de aseguramiento de calidad se ejecute de manera ordenada, eficiente y orientada a los riesgos identificados, es necesario definir un conjunto de estrategias que guíen la planificación y ejecución de las actividades de prueba. Estas estrategias permiten estructurar el trabajo, determinar el enfoque más adecuado según las características del proyecto y asegurar que los esfuerzos se concentren en los aspectos que representan mayor impacto para la operación del sistema. Además, facilitan la definición de criterios claros para la selección de técnicas de prueba y la priorización de escenarios a validar. En el caso de la tienda online NepTune, la definición de estrategias cobra especial relevancia debido a la naturaleza crítica de sus funcionalidades, entre ellas el flujo de pagos, la gestión de productos y la interacción con servicios externos. Por ello, se seleccionaron y adaptaron estrategias que responden tanto al análisis de riesgos como al alcance delimitado del proyecto, priorizando la detección oportuna de fallos en áreas sensibles y la validación de comportamientos clave para la experiencia del usuario final. Asimismo, estas estrategias permiten anticipar posibles

puntos de fallo derivados de la integración con componentes externos y del manejo de información sensible.

Estas estrategias no solo orientan el qué y el cómo se probará, sino que también permiten establecer un marco metodológico coherente con los objetivos del proyecto, optimizando los recursos disponibles y asegurando un proceso de verificación sólido y trazable. De esta manera, el plan de pruebas se convierte en una herramienta de apoyo para la toma de decisiones técnicas y para la evaluación continua del estado de calidad del sistema a lo largo del proyecto.

*Tabla 3 – Estrategias de pruebas NepTune*

Estrategia	Descripción	Justificación	Como se aplicará
<b>Analítica (Basada en Riesgos)</b>	Consiste en diseñar y priorizar las pruebas de acuerdo con los riesgos del sistema, considerando factores como impacto, probabilidad de fallo y criticidad del flujo funcional.	El sistema integra componentes de alta relevancia como la pasarela de pagos, conciliación de ventas, manejo de roles, validación de stock y envío de correos. Debido a su naturaleza crítica, cualquier fallo en estos módulos puede afectar directamente la operación del negocio y la experiencia del usuario final.	<ul style="list-style-type: none"> <li>• Identificación y categorización de riesgos por módulo.</li> <li>• Priorización de casos de prueba en niveles (alto, medio, bajo).</li> <li>• Fortalecimiento de pruebas en flujos clave: pagos, login, carrito y conciliación.</li> <li>• Validaciones específicas sobre las integraciones externas (pasarelas de pago y servicio de correo).</li> </ul>
<b>Basada en Modelo</b>	Las pruebas se diseñan a partir de modelos de flujos, estados o procesos que representan el comportamiento esperado del sistema.	Los procesos de la tienda online son complejos e involucran múltiples transiciones: flujo de compra, aprobación de ventas, gestión de	<ul style="list-style-type: none"> <li>• Elaboración de diagramas de flujo para compra, carga de productos y aprobación de ventas.</li> <li>• Derivación de casos de prueba según estados, decisiones y transiciones</li> </ul>

		productos y permisos según el rol. Modelar estos procesos permite asegurar una cobertura completa y coherente.	<ul style="list-style-type: none"> <li>• Verificación de la cobertura completa de todos los caminos posibles del proceso</li> </ul>
<b>Metódica</b>	Se basa en checklists, buenas prácticas, validaciones comunes, estándares de calidad y taxonomías de fallos conocidos.	La plataforma debe ser usable, funcional tanto en dispositivos móviles como en equipos de escritorio, y mantener coherencia visual. Además, los formularios, catálogos y elementos UI suelen presentar fallos frecuentes que deben prevenirse.	<ul style="list-style-type: none"> <li>• Uso de listas de verificación para UI, formularios, usabilidad y accesibilidad.</li> <li>• Validación minuciosa de todos los campos, reglas y restricciones.</li> <li>• Revisión de estándares de responsividad, coherencia visual y navegación.</li> </ul>
<b>Adversa a la Regresión</b>	Busca asegurar que los cambios recientes no afecten funcionalidades previamente estables, mediante la ejecución repetida y controlada de pruebas ya validadas.	En una tienda online, cualquier cambio puede impactar módulos sensibles como pagos, catálogo, permisos o mensajería. La regresión es clave para evitar efectos secundarios no deseados.	<ul style="list-style-type: none"> <li>• Reutilización de los casos de prueba validados en sprints anteriores.</li> <li>• Ejecución de regresión tras cada cambio funcional o corrección relevante.</li> <li>• Evaluación de automatización futura para flujos de checkout (opcional).</li> </ul>
<b>Reactiva</b>	Permite descubrir defectos inesperados mediante sesiones exploratorias en las que no se siguen pasos predefinidos, sino que se investiga el sistema de forma dinámica.	Dado que la plataforma sigue en desarrollo y algunos flujos pueden presentar comportamientos no contemplados, esta estrategia permite detectar fallos que podrían pasar desapercibidos con pruebas tradicionales.	<ul style="list-style-type: none"> <li>• Ejecución de pruebas exploratorias por módulos.</li> <li>• Diseño de rutas de prueba espontáneas según hallazgos en ejecución.</li> <li>• Uso temprano en versiones inestables o prototipos funcionales.</li> </ul>

### 3.5 Estimación del proyecto

La estimación del esfuerzo se elaboró tomando como punto de partida el análisis de riesgos y el alcance definido para el proyecto. Este enfoque permitió proyectar el tiempo y los recursos necesarios de manera realista, priorizando los componentes críticos de la plataforma NepTune. Para ello, se empleó una combinación de técnicas: evaluación por complejidad, priorización por riesgo y análisis de desviación, lo cual permitió obtener un valor ajustado a las actividades incluidas dentro del alcance. Adicionalmente, este proceso facilitó la identificación de posibles cuellos de botella y dependencias que podrían impactar el cronograma de pruebas. En primer lugar, se identificaron los módulos y funcionalidades que serían evaluados: integración con pasarelas de pago, manejo de datos sensibles, validación de productos por parte del administrador, historias de usuario HU-6 y HU-10, y la regresión acotada sobre los módulos afectados. Cada uno de estos elementos fue clasificado según su nivel de complejidad, considerando factores como cantidad de validaciones, dependencias técnicas, interacción con servicios externos y sensibilidad del proceso. Asimismo, se tuvo en cuenta el grado de criticidad de cada módulo en relación con la operación del negocio. Posteriormente, a cada componente se le asignó un valor de esfuerzo estimado en horas, tomando como referencia la experiencia previa, la naturaleza del riesgo asociado y el número de casos de prueba requeridos. Estos valores iniciales fueron evaluados mediante una desviación estándar simple, con el propósito de ajustar la estimación frente a posibles variaciones en la ejecución. De esta forma, se redujo la incertidumbre asociada a la planificación y se mejoró la confiabilidad

de los resultados proyectados. Con ello se obtuvo un rango más estable y coherente con el comportamiento esperado del proyecto.

La estimación consolidada permitió definir un esfuerzo total que se ajusta a las necesidades del sistema y a la profundidad requerida para validar los flujos críticos. Este esfuerzo contempla actividades como la revisión del entorno, diseño y ejecución de casos de prueba, documentación de resultados, análisis de incidencias y pruebas de regresión controlada. Asimismo, considera tiempos de ajuste y validación posterior a la corrección de defectos críticos. Como resultado, la estimación final refleja un equilibrio entre el nivel de riesgo identificado y la complejidad de los módulos seleccionados, asegurando que la validación del sistema se enfoque en las áreas de mayor impacto operativo, sin destinar recursos a elementos que se encuentran fuera del alcance del proyecto.

*Tabla 4 – Porcentaje de desviación NepTune*

PORCENTAJE DE DESVIACIÓN MARKETPLACE NEPTUNE		
Impedimento	%	Justificación %
Ambiente NO disponible	3%	A veces el ambiente simplemente no está arriba cuando lo necesitamos. Esto nos retrasa porque debemos esperar a que infraestructura lo levante o lo estabilice para seguir con las pruebas.
App externas Off-line	2%	Dependemos de servicios externos y, cuando alguno se cae, no podemos avanzar con validaciones que requieren esa conexión. Muchas veces toca esperar o coordinar nuevamente.
Medios de Pago No disponible	2%	Cuando el servicio de pagos falla, las pruebas se detienen por completo porque el flujo queda incompleto. Nos toca reprogramar y volver a intentar.
Data no disponible	2%	En varias ocasiones la data que necesitamos no está cargada o llega incompleta. Esto nos obliga a esperar o solicitar soporte para continuar.

Bugs sin resolución oportuna	3%	Algunos defectos tardan en corregirse y eso hace que tengamos que pausar ciertos casos. También nos obliga a retomar las pruebas en momentos en que ya teníamos planeado otra cosa.
Inestabilidad de datos	2%	A veces los datos cambian sin previo aviso o se sobrescriben. Esto genera confusión y obliga a repetir pruebas porque los resultados no son consistentes.
Borrados de desarrollos de integración	1%	Ocurre cuando hay errores en los merges y se pierden avances. Toca esperar a que el equipo de desarrollo lo restaure para poder continuar.
Disponibilidad de las personas	2%	Entre vacaciones, incapacidades o reuniones imprevistas, hay momentos en los que el equipo no está completo y esto se refleja en los tiempos de entrega.
Falta de capacidad y disponibilidad por parte de QA (Profesionales de Calidad)	2%	A veces no contamos con el equipo suficiente para atender todas las pruebas que llegan al mismo tiempo. Esto hace que debamos priorizar y dejar algunos temas para después.
Dependencia de personal experto o con conocimiento específico	2%	Hay módulos tan específicos que solo una persona los domina. Cuando no está disponible, las validaciones quedan en pausa hasta que pueda apoyarnos.
Alto tiempo en atención de bugs	3%	La investigación de ciertos defectos toma más tiempo del esperado. A veces el análisis es complejo y nos retrasa otras tareas en cola.
Entregas tardías de desarrollo a calidad	2%	Cuando los desarrollos llegan más tarde de lo planeado, directamente reduce el tiempo que tenemos para probar. Esto nos obliga a correr o a reacomodar todo el plan.
Inconsistencias en los despliegues de los nuevos desarrollos en ambiente QA	2%	Ocurre que a veces el ambiente no queda con la versión correcta o falta algún ajuste. Esto nos hace detenernos y solicitar un nuevo despliegue.
Mala estimación y planeación	1%	Hay casos donde lo estimado inicialmente no se ajusta a la realidad. Esto genera presión y hace que las actividades se acumulen más de lo previsto.
Rotación de personal y/o roles en el equipo	2%	Cada que alguien sale del equipo o llega una persona nueva, hay una curva de adaptación. Esto naturalmente impacta los tiempos.

Actividades del proyecto no planeadas	2%	De vez en cuando surgen tareas de último minuto, incidentes o solicitudes internas. Estas actividades no estaban contempladas y obligan a reorganizar el trabajo.
Cambios constantes en el alcance	3%	Si los requerimientos cambian varias veces durante el desarrollo, debemos volver a ajustar casos de prueba, escenarios e incluso automatizaciones.
Desconocimiento del negocio	1%	Hay situaciones donde necesitamos más contexto funcional. Cuando no lo tenemos, debemos preguntar o revisar documentación, lo cual toma tiempo adicional.
Ejecución en ambientes compartidos	2%	En los ambientes compartidos hay interferencias con otros equipos. A veces alguien modifica datos o deja sesiones abiertas, afectando nuestras pruebas.
Novedades en el equipo de trabajo	1%	Reuniones inesperadas, cambios de prioridad o ajustes internos también mueven el cronograma y reducen el tiempo disponible para ejecutar.
Total % Desviación (Es el total que estimamos se puede desviar el proyecto, según experiencias o históricos), este porcentaje debe disminuir de proyecto a proyecto, para cumplir la promesa de disminuir el desperdicio y aumentar la disponibilidad.	40%	

Tabla 5 - Estimación marketplace Neptune

ESTIMACIÓN MARKETPLACE NEPTUNE				
Número de Horas a trabajar por día:	24.00			
% Desviación	40%			
Actividad	Total Horas Estimadas	Horas con % Desviación	Total Días Estimados	Total Días con % Desviación
<b>Contextualización</b>	8.00	11.20	<b>0.333333333</b>	<b>0.466666667</b>
Entendimiento de la solución de software	2.00	2.80	0.08	0.12
Reunión de contextualización	1.50	2.10	0.06	0.09
Estimación a alto nivel (plan del plan)	1.50	2.10	0.06	0.09

Sustentación de la estimación a alto nivel	1.00	1.40	0.04	0.06
Crear proyecto en herramientas de apoyo	2.00	2.80	0.08	0.12
<b>Planeación</b>	<b>5.50</b>	<b>7.70</b>	<b>0.229166667</b>	<b>0.320833333</b>
Elaboración del plan de calidad	3.00	4.20	0.13	0.18
Socializar y aprobar plan de calidad	1.50	2.10	0.06	0.09
Realizar calendario de actividades	1.00	1.40	0.04	0.06
<b>Diseño/Ejecución</b>	<b>10.00</b>	<b>14.00</b>	<b>0.42</b>	<b>0.583333333</b>
Fallo en la integración de pasarelas de pago	2.00	2.80	0.08	0.12
Fuga o exposición de datos sensibles (clientes/pagos)	2.00	2.80	0.08	0.12
Retraso en la validación de productos por el Administrador	1.00	1.40	0.04	0.06
Historia de Usuario 6	1.00	1.40	0.04	0.06
Historia de Usuario 10	3.00	4.20	0.13	0.18
Regresión	1.00	1.40	0.04	0.06
<b>Documentación, cierre y retrospectiva</b>	<b>9.00</b>	<b>12.60</b>	<b>0.375</b>	<b>0.525</b>
Actualización documentación diaria	1.00	1.40	0.04	0.06
Creación informe de cierre	3.00	4.20	0.13	0.18
Socialización del informe de cierre	1.00	1.40	0.04	0.06
Creación SUITE (Base de conocimiento)	2.00	2.80	0.08	0.12
Retrospectiva del Proyecto	2.00	2.80	0.08	0.12
<b>Total Tiempo Estimado de Pruebas</b>	<b>32.50</b>	<b>45.50</b>	<b>1.35</b>	<b>1.90</b>

### 3.6 Flujo de pruebas

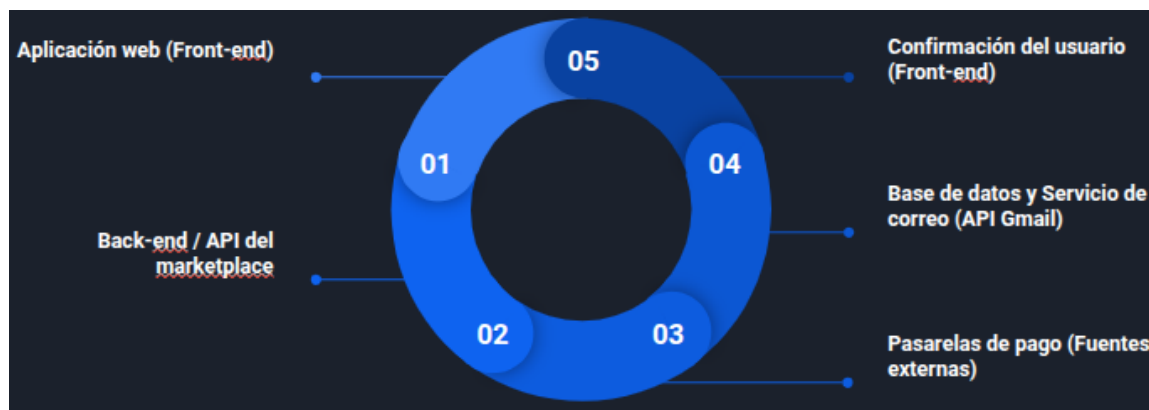


Figura 3 – Flujo de pruebas NepTune

### 3.7 Tipos de pruebas a ejecutar

Para garantizar la calidad del producto final y mitigar los riesgos identificados durante el análisis previo, se estableció un conjunto específico de tipos de pruebas orientadas a validar los componentes críticos del proyecto NepTune. Estas pruebas se seleccionaron en función del impacto potencial sobre la experiencia del usuario, la seguridad de la información y la estabilidad operativa del sistema, priorizando aquellos escenarios donde un fallo podría comprometer transacciones, datos sensibles o procesos esenciales del negocio.

En este sentido, la planificación se centró en validar los flujos asociados al procesamiento de pagos, la manipulación de información personal, la correcta gestión de productos por parte del administrador y la evaluación de funcionalidades clave definidas en las historias de usuario seleccionadas. Adicionalmente, se incluyó una regresión acotada para asegurar que las modificaciones implementadas no afecten módulos previamente estables.

El conjunto de pruebas definido constituye una línea base que permite verificar la confiabilidad del sistema, evaluar su respuesta ante escenarios críticos y confirmar que cumple con los requisitos funcionales y no funcionales asociados al alcance establecido.

01	Pruebas funcionales manuales	<ul style="list-style-type: none"> <li>Flujo de compra, pasarela de pagos, descuentos y totales, login, registro, perfil, catálogo y carga de productos, validación del administrador, envío de correo de confirmación.</li> </ul>
02	Pruebas de regresión	<ul style="list-style-type: none"> <li>Módulos críticos: Pago, carrito, login, catálogo y perfil.</li> </ul>
03	Pruebas End-To-End	<ul style="list-style-type: none"> <li>Cliente: Registro &gt; Selección de producto &gt; Pago &gt; Correo de confirmación</li> <li>Proveedor: Carga de producto &gt; Validación del admin &gt; Compra del cliente</li> </ul>
04	Pruebas de usabilidad y Responsive Design	<ul style="list-style-type: none"> <li>Vista móvil</li> <li>Navegación</li> <li>Visualización de productos</li> </ul>
05	Pruebas de confirmación	<ul style="list-style-type: none"> <li>Verificación de qué los bugs reportados ya no se presenten</li> </ul>

*Figura 4 - Tipos de pruebas a ejecutar NepTune*

### 3.8 Precondiciones

Antes de ejecutar los casos de prueba definidos para el proyecto NepTune, es necesario asegurar una serie de precondiciones que garanticen que el entorno, los datos y los componentes del sistema están en un estado adecuado para su validación. Estas condiciones previas permiten que las pruebas se desarrollen de manera consistente, reducen la aparición de errores ajenos a la funcionalidad evaluada y aseguran que los resultados obtenidos reflejen el comportamiento real del sistema. A continuación, se describen las precondiciones establecidas para el proceso de pruebas.

1. Ambiente QA disponible y estable, con la versión del desarrollo correctamente desplegada.

2. Accesos habilitados para todos los roles: Cliente, Proveedor, Administrador.
3. Conexión operativa a las integraciones: Pasarela de pago, API Gmail para envío de correos.
4. Base de datos cargada con datos mínimos necesarios: Productos de prueba, Usuarios de prueba, Vales o descuentos configurados.
5. Criterios de aceptación aprobados y claros para cada historia dentro del alcance.
6. Casos de prueba diseñados y revisados, priorizados según el análisis de riesgos.
7. Logs habilitados en Back-End para validar: Transacciones, Respuestas de la pasarela, Errores.
8. Acceso a herramientas de monitoreo y BD para verificar estados de pago y conciliación.
9. Correo de prueba (Gmail) configurado para recibir la confirmación de compra.
10. Datos de tarjeta de prueba proporcionados por la pasarela.
11. Entregables previos completos: Build desplegado, Historias cerradas por desarrollo, Evidencia de pruebas unitarias.

### 3.9 Casos de prueba

CP-PAGO-001	
Nombre o descripción	Pago exitoso
Objetivo	Validar que el pago se procese correctamente cuando la pasarela responde "Aprobado".

<b>Precondiciones</b>	- Usuario logueado - Producto en el carrito - Pasarela operativa - Tarjeta válida disponible
<b>Pasos</b>	1. Ir al carrito. 2. Seleccionar “Comprar”. 3. Ingresar los datos de tarjeta válida. 4. Confirmar el pago.
<b>Resultado esperado</b>	El pedido queda en estado “Pagado”.
<b>Información de seguimiento</b>	
<b>Resultado obtenido</b>	
<b>Estado</b>	
<b>Observaciones</b>	

<b>CP-PAGO-002</b>	
<b>Nombre o descripción</b>	Pago rechazado
<b>Objetivo</b>	Validar que el sistema maneje correctamente un pago con tarjeta inválida.
<b>Precondiciones</b>	- Usuario logueado - Producto en el carrito - Tarjeta inválida definida
<b>Pasos</b>	1. Ir al carrito. 2. Proceder al pago. 3. Ingresar datos de tarjeta inválida. 4. Confirmar pago.
<b>Resultado esperado</b>	El pedido queda en estado “Rechazado”.
<b>Información de seguimiento</b>	
<b>Resultado obtenido</b>	
<b>Estado</b>	
<b>Observaciones</b>	

<b>CP-PAGO-003</b>	
<b>Nombre o descripción</b>	Time-out de pasarela
<b>Objetivo</b>	Validar el comportamiento ante demora o falta de respuesta de la pasarela.
<b>Precondiciones</b>	- Producto en carrito - Modo de pasarela configurado en time-out
<b>Pasos</b>	1. Proceder al pago. 2. Ingresar datos de tarjeta. 3. Confirmar pago.
<b>Resultado esperado</b>	El usuario recibe un mensaje de error por tiempo de espera.
<b>Información de seguimiento</b>	
<b>Resultado obtenido</b>	
<b>Estado</b>	

Observaciones	
---------------	--

CP-PAGO-004	
Nombre o descripción	Error inesperado
Objetivo	Validar que el sistema maneje errores 500 o respuestas desconocidas.
Precondiciones	- Pasarela simulada con error 500
Pasos	1. Proceder al pago. 2. Ingresar datos de tarjeta. 3. Confirmar pago.
Resultado esperado	El sistema muestra un error genérico de procesamiento.
Información de seguimiento	
Resultado obtenido	
Estado	
Observaciones	

CP-SEG-001	
Nombre o descripción	Datos enmascarados en pantalla
Objetivo	Verificar que los datos sensibles no se muestren visibles en el frontend.
Precondiciones	Estar en pantalla de checkout con datos ingresados.
Pasos	1. Revisar el campo de número de tarjeta. 2. Revisar el campo de CVV y expiración.
Resultado esperado	El número de tarjeta se muestra en formato enmascarado.
Información de seguimiento	
Resultado obtenido	
Estado	
Observaciones	

CP-SEG-002	
Nombre o descripción	Datos no viajan en texto plano
Objetivo	Validar que no se envíen datos sensibles en las peticiones de red.
Precondiciones	Herramienta de inspección (Network) abierta.
Pasos	1. Realizar el proceso de pago. 2. Revisar la petición de pago en la pestaña "Network".
Resultado esperado	No se detecta información crítica en texto plano.
Información de seguimiento	

Resultado obtenido	
Estado	
Observaciones	
<b>CP-SEG-003</b>	
Nombre o descripción	API no expone información sensible
Objetivo	Verificar que las respuestas de la API no devuelvan datos críticos.
Precondiciones	Flujo de pago ejecutado.
Pasos	1. Revisar la respuesta de la API luego del intento de pago.
Resultado esperado	La API no retorna información sensible no necesarios para la operación.
<b>Información de seguimiento</b>	
Resultado obtenido	
Estado	
Observaciones	

<b>CP-ADM-001</b>	
Nombre o descripción	Aprobar producto
Objetivo	Validar que el administrador pueda aprobar un producto pendiente.
Precondiciones	Producto en estado "Pendiente".
Pasos	1. Ingresar al módulo de productos pendientes. 2. Seleccionar un producto. 3. Presionar "Aprobar".
Resultado esperado	El producto cambia, ha estado "Aprobado".
<b>Información de seguimiento</b>	
Resultado obtenido	
Estado	
Observaciones	

<b>CP-ADM-002</b>	
Nombre o descripción	Rechazar producto
Objetivo	Validar que el administrador pueda rechazar productos.
Precondiciones	Producto en estado "Pendiente".
Pasos	1. Abrir la lista de productos pendientes. 2. Seleccionar un producto. 3. Presionar "Rechazar".
Resultado esperado	El producto cambia, ha estado "Rechazado".
<b>Información de seguimiento</b>	
Resultado obtenido	

Estado	
Observaciones	

CP-ADM-003	
Nombre o descripción	Evitar estados bloqueados
Objetivo	Verificar que no ocurran estados inválidos.
Precondiciones	Producto en estado "Pendiente".
Pasos	1. Cambiar estado del producto (aprobar o rechazar). 2. Revisar que no queden estados vacíos o inconsistentes.
Resultado esperado	El sistema no permite estados nulos
Información de seguimiento	
Resultado obtenido	
Estado	
Observaciones	

CP-CAR-001	
Nombre o descripción	Visualizar carrito
Objetivo	Validar que los productos se muestren correctamente.
Precondiciones	Carrito con productos.
Pasos	Abrir la sección "Carrito".
Resultado esperado	El carrito muestra correctamente los productos agregados, con nombre, descripción y precio.
Información de seguimiento	
Resultado obtenido	
Estado	
Observaciones	

CP-CAR-002	
Nombre o descripción	Agregar producto
Objetivo	Validar que un producto pueda agregarse al carrito.
Precondiciones	Producto disponible.
Pasos	1. Abrir página del producto. 2. Presionar "Agregar al carrito".
Resultado esperado	El producto agregado aparece inmediatamente en el carrito.
Información de seguimiento	
Resultado obtenido	
Estado	
Observaciones	

CP-CAR-003	
Nombre o descripción	Eliminar producto
Objetivo	Validar la eliminación de un producto.
Precondiciones	Producto en carrito.
Pasos	1. Abrir carrito. 2. Presionar “Eliminar” sobre el producto.
Resultado esperado	El producto se elimina del carrito.
Información de seguimiento	
Resultado obtenido	
Estado	
Observaciones	

CP-CAR-004	
Nombre o descripción	Ver total del carrito
Objetivo	Validar el cálculo del total.
Precondiciones	Carrito con varios productos.
Pasos	Abrir carrito
Resultado esperado	El total refleja la suma correcta del valor de todos los productos en el carrito.
Información de seguimiento	
Resultado obtenido	
Estado	
Observaciones	

CP-MAIL-001	
Nombre o descripción	Correo enviado
Objetivo	Validar que el correo llegue después de un pago exitoso.
Precondiciones	Compra exitosa completada.
Pasos	Revisar la bandeja de entrada Gmail asociada.
Resultado esperado	El correo llega a la cuenta Gmail del usuario con toda la información de la compra realizada.
Información de seguimiento	
Resultado obtenido	
Estado	
Observaciones	

CP-MAIL-002	
Nombre o descripción	Validar PDF adjunto
Objetivo	Verificar que el PDF contiene la información correcta.
Precondiciones	Correo recibido con adjunto.
Pasos	Abrir el PDF adjunto.
Resultado esperado	El PDF contiene correctamente los productos, cantidades, código de compra, descuento aplicado y fecha de expedición.
Información de seguimiento	
Resultado obtenido	
Estado	
Observaciones	

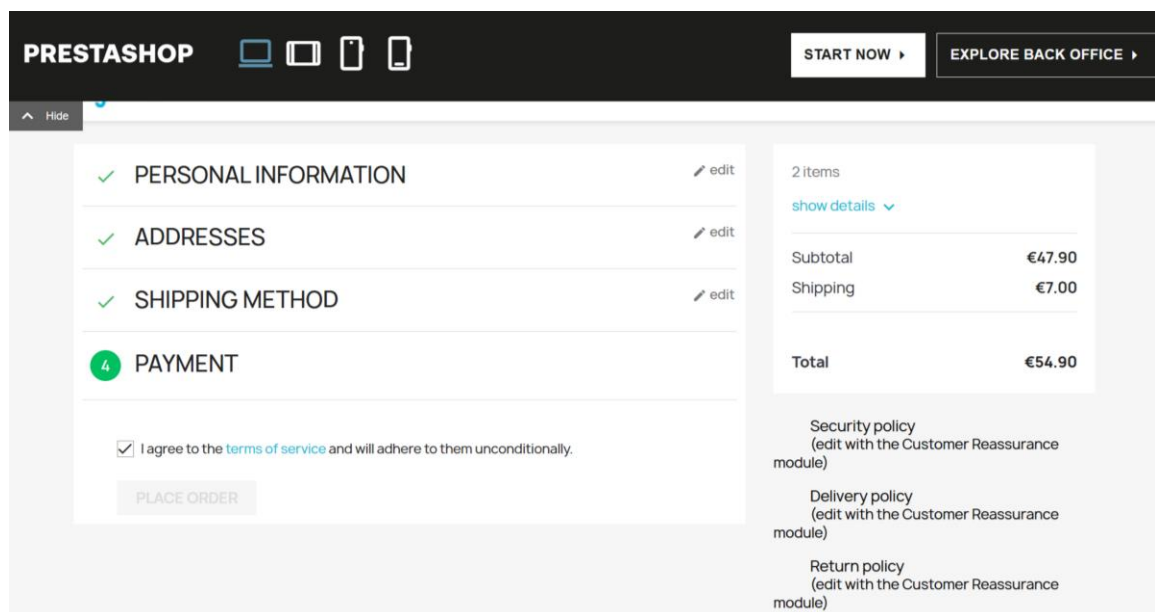
CP-MAIL-003	
Nombre o descripción	Correo no enviado en pago fallido
Objetivo	Validar que no se envíe correo cuando el pago falla.
Precondiciones	Pago rechazado o fallido.
Pasos	Revisar inbox.
Resultado esperado	No se envía ningún correo cuando el pago es rechazado o falla.
Información de seguimiento	
Resultado obtenido	
Estado	
Observaciones	

### 3.10 Evidencias de ejecución

Durante la ejecución del plan de pruebas del proyecto NepTune se recolectaron evidencias que respaldan de manera objetiva y verificable la aplicación de los casos de prueba definidos en el alcance. Estas evidencias corresponden principalmente a capturas de pantalla obtenidas durante la ejecución de los diferentes flujos evaluados, así como a registros visuales asociados a respuestas del sistema en escenarios exitosos y fallidos.

Las evidencias se encuentran vinculadas a los identificadores de los casos de prueba y documentan eventos clave como la respuesta del sistema ante transacciones aprobadas y rechazadas, la gestión de errores provenientes de servicios externos, la

actualización de estados de pedidos y la interacción con funcionalidades administrativas. Su función principal es demostrar que las pruebas fueron ejecutadas efectivamente y que los resultados reportados se fundamentan en observaciones reales del comportamiento del sistema. Adicionalmente, estas evidencias constituyen un insumo fundamental para el proceso de análisis y para la gestión de defectos, ya que permiten reproducir los escenarios evaluados, validar los hallazgos reportados y servir como soporte para la toma de decisiones técnicas posteriores. De esta manera, se garantiza la trazabilidad completa entre planificación, ejecución y resultados.



*Figura 5 - Evidencia de ejecución del caso de prueba CP-PAGO-001: flujo de pago exitoso con pasarela externa*

PRESTASHOP

START NOW ▶ EXPLORE BACK OFFICE ▶

Hide

✓ PERSONAL INFORMATION [edit](#)

✓ ADDRESSES [edit](#)

✓ SHIPPING METHOD [edit](#)

4 PAYMENT

I agree to the [terms of service](#) and will adhere to them unconditionally.

PLACE ORDER

2 items  
[show details](#) ▼

Subtotal	€47.90
Shipping	€7.00
<b>Total</b>	<b>€54.90</b>

Security policy  
(edit with the Customer Reassurance module)

Delivery policy  
(edit with the Customer Reassurance module)

Return policy  
(edit with the Customer Reassurance module)

*Figura 6 - Evidencia de ejecución del caso de prueba CP-PAGO-002: manejo de pago rechazado por la pasarela*

PRESTASHOP

START NOW ▶ EXPLORE BACK OFFICE ▶

Hide

✓ PERSONAL INFORMATION [edit](#)

✓ ADDRESSES [edit](#)

✓ SHIPPING METHOD [edit](#)

4 PAYMENT

I agree to the [terms of service](#) and will adhere to them unconditionally.

PLACE ORDER

2 items  
[show details](#) ▼

Subtotal	€47.90
Shipping	€7.00
<b>Total</b>	<b>€54.90</b>

Security policy  
(edit with the Customer Reassurance module)

Delivery policy  
(edit with the Customer Reassurance module)

Return policy  
(edit with the Customer Reassurance module)

*Figura 7 - Evidencia de ejecución del caso de prueba CP-PAGO-003: validación del estado del pedido ante error de transacción*

PRESTASHOP

START NOW EXPLORE BACK OFFICE

Hide

✓ PERSONAL INFORMATION edit

✓ ADDRESSES edit

✓ SHIPPING METHOD edit

4 PAYMENT

I agree to the [terms of service](#) and will adhere to them unconditionally.

PLACE ORDER

2 items

[show details](#)

Subtotal	€47.90
Shipping	€7.00
<b>Total</b>	<b>€54.90</b>

Security policy  
(edit with the Customer Reassurance module)

Delivery policy  
(edit with the Customer Reassurance module)

Return policy  
(edit with the Customer Reassurance module)

*Figura 8 - Evidencia de ejecución del caso de prueba CP-PAGO-004: respuesta del sistema ante fallo en la comunicación con la pasarela de pago*

PRESTASHOP

START NOW EXPLORE BACK OFFICE

Hide

✓ PERSONAL INFORMATION edit

✓ ADDRESSES edit

✓ SHIPPING METHOD edit

4 PAYMENT

I agree to the [terms of service](#) and will adhere to them unconditionally.

PLACE ORDER

2 items

[show details](#)

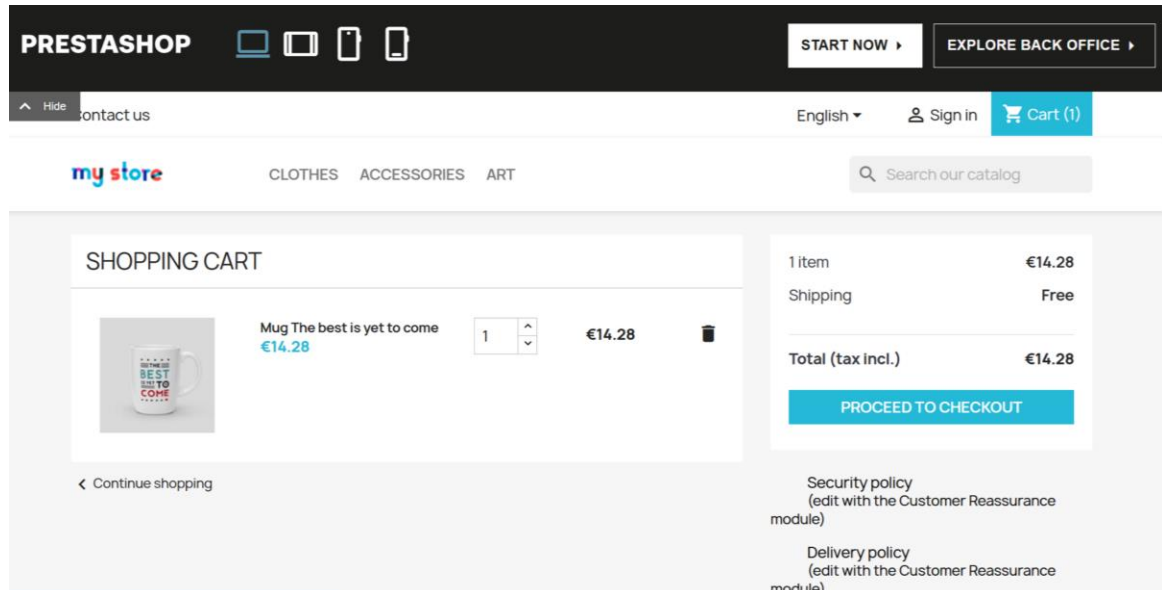
Subtotal	€47.90
Shipping	€7.00
<b>Total</b>	<b>€54.90</b>

Security policy  
(edit with the Customer Reassurance module)

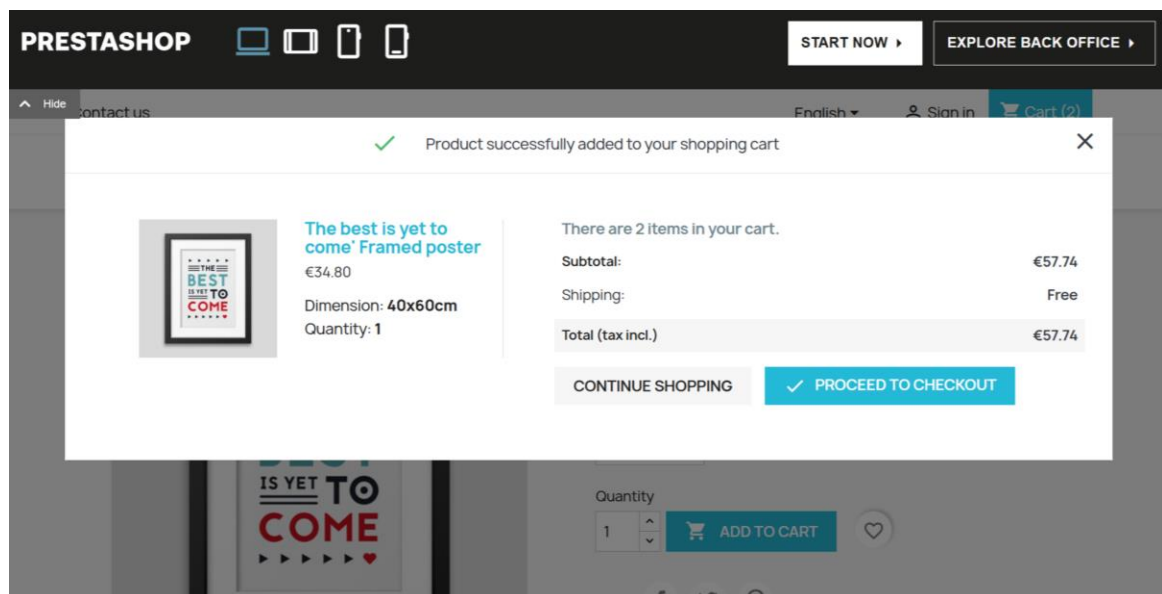
Delivery policy  
(edit with the Customer Reassurance module)

Return policy  
(edit with the Customer Reassurance module)

*Figura 9 - Evidencia de ejecución del caso de prueba CP-SEG-001: verificación de no exposición de datos sensibles en el front-end. Evidencia de ejecución del caso de prueba CP-SEG-002: validación de manejo seguro de información durante el proceso transaccional. Evidencia de ejecución del caso de prueba CP-SEG-003: control de mensajes de error relacionados con información sensible*



*Figura 10 - Evidencia de ejecución del caso de prueba CP-CAR-001: adición de productos al carrito de compras*





*Figura 11 - Evidencia de ejecución del caso de prueba CP-CAR-002: actualización de cantidades y valores en el carrito de compras*

**PRESTASHOP** [Desktop Icon] [Tablet Icon] [Mobile Icon]

[START NOW](#) [EXPLORE BACK OFFICE](#)

Hide

### SHOPPING CART

	The best is yet to come' Framed poster €34.80 Dimension: 40x60cm	1	€34.80	
---	---	---	--------	---

[Continue shopping](#)

1 item	€34.80
Shipping	Free
<b>Total (tax incl.)</b>	<b>€34.80</b>

[PROCEED TO CHECKOUT](#)

Security policy (edit with the Customer Reassurance module)

Delivery policy (edit with the Customer Reassurance module)

Return policy (edit with the Customer Reassurance module)





*Figura 12 - Evidencia de ejecución del caso de prueba CP-CAR-003: validación del flujo del carrito previo al proceso de pago*

**PRESTASHOP** [Desktop Icon] [Tablet Icon] [Mobile Icon]

[START NOW](#) [EXPLORE BACK OFFICE](#)

Hide **my store** CLOTHES ACCESSORIES ART

### SHOPPING CART

	The best is yet to come' Framed poster €34.80 Dimension: 40x60cm	1	€34.80	
	Mountain fox cushion €22.68 Color: White	1	€22.68	

[Continue shopping](#)

2 items	€57.48
Shipping	Free
<b>Total (tax incl.)</b>	<b>€57.48</b>

[PROCEED TO CHECKOUT](#)

Security policy (edit with the Customer Reassurance module)

Delivery policy (edit with the Customer Reassurance module)

Return policy (edit with the Customer Reassurance module)

*Figura 13 - Evidencia de ejecución del caso de prueba CP-CAR-004: comportamiento del carrito ante eliminación de productos*

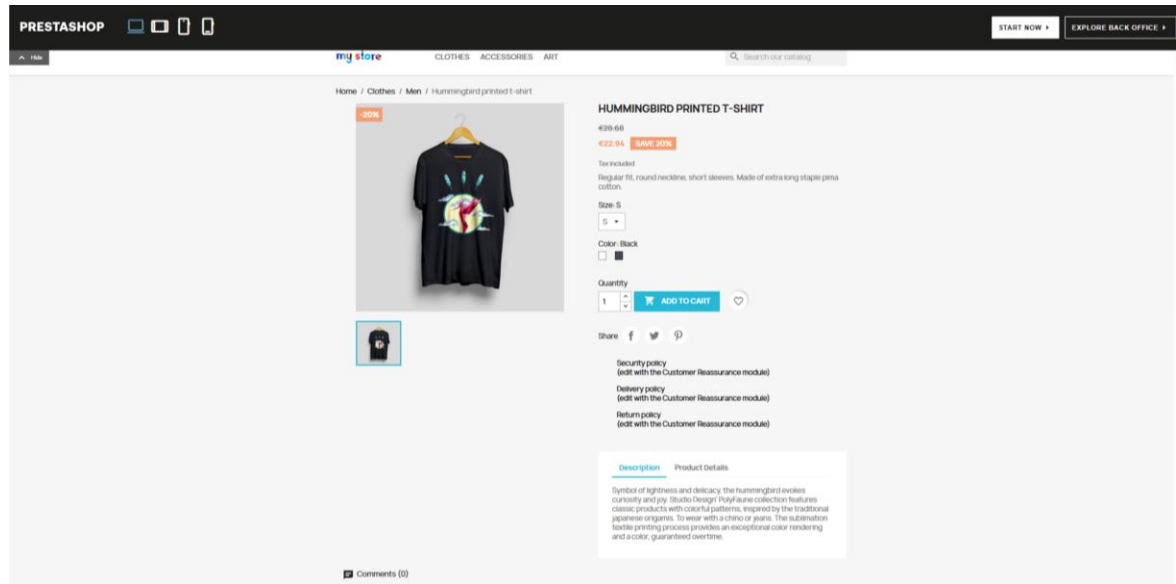


Figura 14 - Evidencia de ejecución del caso de prueba CP-UH-U02-001

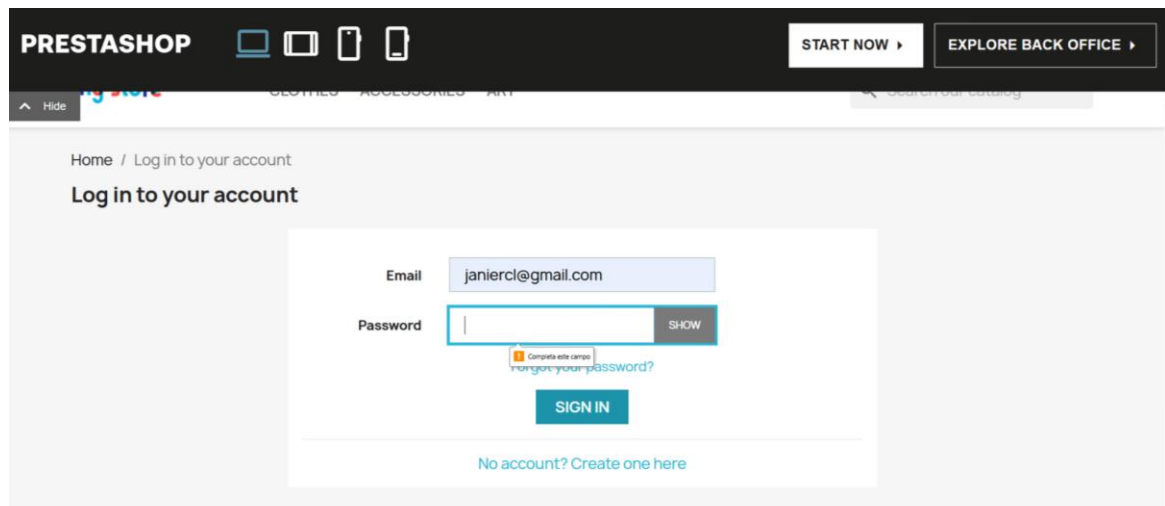
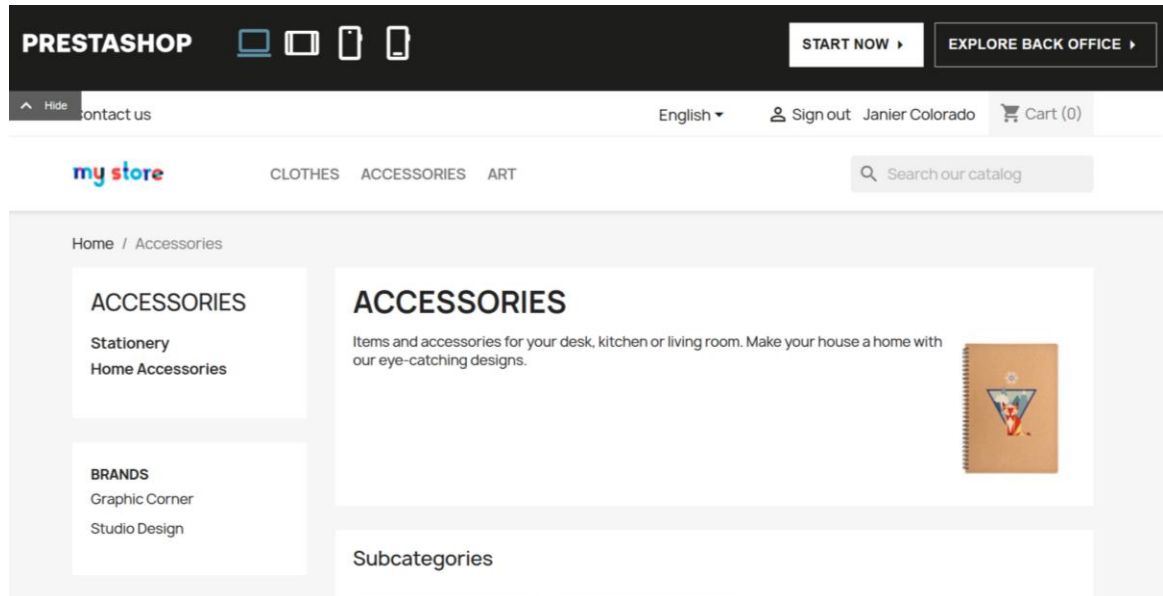


Figura 15 - CP-LOGIN-003 Evidencia de ejecución del caso validación de campos vacíos



*Figura 16 - CP-LOGIN-005: Evidencia de ejecución del caso de prueba sesión inválida al expirar*

### 3.11 Registro y gestión de defectos (Bug Tracker)

Los defectos identificados durante la ejecución de las pruebas fueron registrados y gestionados mediante un bug tracker, el cual permitió centralizar, clasificar y dar seguimiento a cada incidencia detectada en la plataforma NepTune. Para cada defecto se documentó información clave como su descripción detallada, el módulo afectado, la severidad, los pasos para reproducirlo, el resultado esperado, el resultado obtenido y su estado dentro del ciclo de vida del defecto.

El uso del bug tracker facilitó no solo el control de los defectos individuales, sino también el análisis global del comportamiento del sistema, permitiendo identificar concentraciones de incidencias por módulo y severidad. Este enfoque estructurado favoreció la priorización de defectos críticos y el entendimiento de cuáles fallos representan un mayor riesgo para la operación del negocio.

Asimismo, el registro sistemático de defectos permitió evidenciar patrones recurrentes de fallo y sirvió como base para el análisis posterior de resultados y la discusión técnica de los hallazgos. En este sentido, el bug tracker se consolidó como una herramienta clave para garantizar la transparencia, la trazabilidad y la consistencia del proceso de pruebas.

*Tabla 6 - Archivo de bugs en Bugtracker*

Impacto: Efecto hacia el cliente	Prioridad: Rapidez con que se necesita resuelto un reporte	Tipo	Naturaleza	Reporte	Estado	Creado Por	Fecha Creación	Asignado a	Fecha de Asignación
Alto	Alta	Hallazgo	Funcional	La plataforma no cuenta con una pasarela de pago asociada, impidiendo el flujo de cobros o transacciones dentro del sistema. Pasos: 1. Ingresar al módulo de pagos o carrito de compras. 2. Intentar proceder al pago.	Hallazgo	SA	2025-12-10	JC	9/12/2025
Medio	Alta	Hallazgo	Funcional	Los usuarios no cuentan con acceso al panel administrativo, lo que limita la gestión del sistema. Pasos: 1. Iniciar sesión con un rol administrativo. 2. Intentar acceder al panel administrativo desde el menú principal.	Hallazgo	SA	2025-12-10	JC	9/12/2025
Alto	Alta	Hallazgo	Funcional	La plataforma no cuenta con una pasarela de pago asociada, impidiendo el flujo de cobros o transacciones dentro del sistema. Pasos: 1. Ingresar al	Hallazgo	SA	2025-12-10	SA	10/12/2025

				módulo de pagos o carrito de compras. 2. Intentar proceder al pago.					
--	--	--	--	--	--	--	--	--	--

*Tabla 7 – Reporte de hallazgos de ejecución e información para el seguimiento.*

	Información para el Seguimiento		
Id	Resultado Obtenido	Estado	Observaciones
CP-PAGO-001	No es posible realizar la compra, en la opción payment no carga ningún formulario para asociar un medio de pago	Pendiente, no se cumplen condiciones	No hay pasarela de pagos asociada
CP-PAGO-002	No es posible realizar la compra, en la opción payment no carga ningún formulario para asociar un medio de pago	Pendiente, no se cumplen condiciones	No hay pasarela de pagos asociada
CP-PAGO-003	No hay pasarela de pagos asociada	Pendiente, no se cumplen condiciones	No hay pasarela de pagos asociada
CP-PAGO-004	No hay pasarela de pagos asociada	Pendiente, no se cumplen condiciones	No hay pasarela de pagos asociada
CP-SEG-001	No hay pasarela de pagos asociada	Pendiente, no se cumplen condiciones	No hay pasarela de pagos asociada
CP-SEG-002	No hay pasarela de pagos asociada	Pendiente, no se permite realizar el proceso de pago en el sitio	El sistema no habilita el botón de pagar y no se puede validar la data que viaja en la red
CP-SEG-003	No hay pasarela de pagos asociada	Pendiente, no se permite realizar el proceso de pago en el sitio	El sistema no habilita el botón de pagar y no se puede validar la data que viaja en la red
CP-ADM-001	Pendiente por realizar	Pendiente, no se cuenta con acceso al perfil de administrador	Sin observaciones
CP-ADM-002	Pendiente por realizar	Pendiente, no se cuenta con acceso al perfil de administrador	Sin observaciones
CP-ADM-003	Pendiente por realizar	Pendiente, no se cuenta con acceso al perfil de administrador	Sin observaciones
CP-CAR-001	El carrito muestra correctamente los productos agregados, con nombre, descripción y precio.	Se realizó prueba con éxito	Sin observaciones
CP-CAR-002	El producto agregado aparece inmediatamente en el carrito.	Se realizó prueba con éxito	Sin observaciones
CP-CAR-003	El producto se elimina del carrito.	Se realizó prueba con éxito	Se elimina el producto de inmediato, pero no se muestra un mensaje que lo confirme
CP-CAR-004	El total refleja la suma correcta del valor de todos los productos en el carrito.	Se realizó prueba con éxito	Sin observaciones

CP-MAIL-001	Pendiente por realizar	Pendiente, no se cumplen condiciones	Sin observaciones
CP-MAIL-002	Pendiente por realizar	Pendiente, no se cumplen condiciones	Sin observaciones
CP-MAIL-003	Pendiente por realizar	Pendiente, no se cumplen condiciones	Sin observaciones
CP-UH-U02-001	Se visualiza: imagen, nombre, descripción, precio, talla pero no se visualiza stock.	Se realiza prueba pero no cumple	No se visualiza stock
CP-LOGIN-003	Se muestran mensajes de validación: "Completa este campo".	Se realiza prueba con éxito	No dice "campo obligatorio" en su lugar dice "Completa este campo"
CP-LOGIN-005	No detecta token de expirado	Se realiza prueba pero no cumple	No detecta token de expirado

## **4. Desarrollo e implementación del aprendizaje**

### **4.1 Aplicación de los conocimientos adquiridos**

Los conocimientos adquiridos durante el seminario fueron aplicados de manera integral, progresiva y consciente a lo largo del desarrollo del plan de pruebas del proyecto NepTune, permitiendo trasladar los conceptos teóricos a un contexto práctico que simula condiciones reales de un sistema de comercio electrónico. Esta aplicación práctica favoreció la comprensión del aseguramiento de la calidad como un proceso continuo y transversal al ciclo de vida del software, más allá de la ejecución puntual de pruebas.

El análisis de riesgos se empleó como eje central para la planificación del proceso de pruebas, permitiendo identificar los componentes con mayor probabilidad de fallo y mayor impacto operativo. A partir de este análisis se definió un alcance de pruebas alineado con las limitaciones de tiempo y recursos del proyecto, replicando decisiones habituales en entornos profesionales donde no es posible probar todo con la misma profundidad.

La selección de estrategias de prueba se realizó considerando la naturaleza del sistema y su dependencia de servicios externos, combinando enfoques analíticos y pruebas basadas en riesgos. Esta decisión permitió comprender cómo la estrategia adoptada influye directamente en la eficiencia del proceso de testing y en la calidad de los resultados obtenidos. El diseño de los casos de prueba a partir de historias de usuario fortaleció la trazabilidad entre requisitos, riesgos y pruebas, mientras que la ejecución controlada de dichos casos permitió validar el comportamiento real del sistema.

Finalmente, el uso del bug tracker como herramienta de gestión de defectos fortaleció habilidades relacionadas con la documentación técnica, el análisis de

incidencias y la comunicación de hallazgos. En conjunto, la aplicación de estos conocimientos permitió consolidar un proceso de aseguramiento de calidad metodológico, trazable y orientado a resultados, generando aprendizajes transferibles a futuros proyectos de software.

#### **4.2 Resultados obtenidos del proceso de pruebas**

La ejecución del proceso de pruebas sobre la plataforma NepTune permitió obtener resultados concretos y verificables a partir de la ejecución de los casos de prueba definidos dentro del alcance del proyecto. Estos resultados se fundamentan en las evidencias recolectadas durante la ejecución, en los defectos registrados y en el seguimiento de métricas asociadas a la cobertura y severidad de las incidencias.

Los resultados muestran que el sistema presenta un comportamiento estable en escenarios funcionales estándar, particularmente en los flujos principales del carrito de compras y la navegación general de la plataforma. No obstante, se identificaron fallos relevantes en funcionalidades críticas, principalmente en el módulo de integración con la pasarela de pago y en los procesos administrativos de validación de productos.

En el caso del flujo de pago, los resultados evidencian inconsistencias ante escenarios de error, respuestas inesperadas y tiempos de espera prolongados, los cuales impactan directamente la actualización del estado del pedido y del inventario. Estos comportamientos fueron registrados como defectos con severidad media y alta debido a su impacto operativo. Respecto al manejo de datos sensibles, los resultados indican que no se presentó exposición directa de información crítica en las interfaces visibles al usuario. Sin embargo, se identificaron comportamientos que requieren atención, como

mensajes de error poco controlados y respuestas técnicas que podrían optimizarse para reducir riesgos potenciales.

En relación con las historias de usuario HU-6 y HU-10, los resultados evidencian que los criterios de aceptación se cumplen en la mayoría de los escenarios principales, aunque se identificaron fallos en condiciones específicas o escenarios alternos. Estos resultados permiten establecer un panorama claro del estado del sistema frente a los escenarios evaluados y constituyen la base objetiva para el análisis posterior.

### **4.3 Análisis de resultados**

El análisis de los resultados obtenidos permite identificar patrones consistentes en la distribución de defectos y en el comportamiento del sistema frente a los escenarios evaluados. Los datos evidencian que los módulos clasificados como de mayor riesgo concentraron la mayor cantidad de incidencias, lo que confirma la efectividad del enfoque de pruebas basado en riesgos aplicado durante la planificación.

El flujo de pago se consolidó como el componente más sensible del sistema, seguido por los procesos administrativos de validación de productos. Esta distribución valida la coherencia entre el análisis de riesgos inicial, la definición del alcance y los resultados obtenidos durante la ejecución de las pruebas.

Asimismo, el análisis evidencia que los módulos de menor criticidad presentaron un número reducido de defectos, lo que refuerza la decisión de enfocar los esfuerzos de testing en los componentes con mayor impacto operativo. La comparación entre el alcance estimado y el alcance real indica que, aunque no se ejecutaron pruebas sobre la

totalidad del sistema, la cobertura alcanzada fue suficiente para evaluar de manera confiable la calidad de los componentes críticos.

#### **4.4 Discusión técnica de los hallazgos**

La discusión técnica de los hallazgos permite profundizar en las causas y las implicaciones de los defectos identificados durante el proceso de pruebas. Los resultados evidencian debilidades estructurales en la gestión de integraciones externas y en el control de estados del sistema, especialmente en el flujo de pago y en los procesos administrativos.

Las inconsistencias detectadas ante respuestas inesperadas de la pasarela de pago sugieren limitaciones en el manejo de excepciones, validación de estados y sincronización entre componentes internos y servicios externos. Desde una perspectiva técnica, estos hallazgos indican la necesidad de fortalecer los mecanismos de control de errores y de mejorar la resiliencia del sistema ante fallos externos.

De igual forma, las inconsistencias entre la interfaz de usuario y la persistencia de datos en los procesos administrativos evidencian oportunidades de mejora en la comunicación entre capas del sistema. Estos hallazgos tienen implicaciones directas en la experiencia del usuario y en la operación del marketplace, especialmente en contextos de crecimiento y escalabilidad.

En conjunto, los hallazgos demuestran que el enfoque de pruebas basado en riesgos fue adecuado para identificar defectos con alto impacto operativo y que los problemas detectados no corresponden a fallos aislados, sino a áreas que requieren atención prioritaria desde una perspectiva técnica y arquitectónica.

## 5 Informe de cierre

### 5.1 Alcance estimado vs alcance real

Alcance estimado (Plan de pruebas)

Durante la planificación se estimó cubrir los siguientes elementos:

Flujo completo de pago.

Inicio de sesión y manejo de sesión.

Visualización de productos.

Validación y aprobación de productos (HU6 y HU10).

Pruebas de regresión sobre módulos afectados.

Alcance real

El alcance real se ajustó basado en la matriz de riesgos, priorizando:

Pago (crítico para el negocio).

Seguridad de datos sensibles.

Validación y aprobación de productos por el Administrador.

Regresión acotada sobre: pago, validación de productos, consulta de estado del pedido, flujos afectados por HU6 y HU10.

*Tabla 8 - Diferencias y sustentación en variaciones del alcance*

<b>Tipo</b>	<b>Descripción</b>	<b>Justificación</b>
Alcance Mayor	Se agregaron pruebas adicionales sobre manejo de errores en la pasarela de pago	Riesgo crítico: Fallo en integración con pasarela
Alcance Menor	Se redujo la regresión general	Alcance acotado por tiempo y priorización basada en la matriz de riesgo

## Análisis del alcance definido VS el alcance real

El alcance definido inicialmente para el proceso de pruebas se estableció a partir del análisis de riesgos y contempló la validación de los componentes críticos de la plataforma NepTune. El alcance estimado incluyó la ejecución de pruebas sobre la integración con pasarelas de pago, el manejo y protección de datos sensibles, la validación administrativa de productos, las historias de usuario HU-6 y HU-10, y una regresión acotada sobre los módulos impactados por las pruebas ejecutadas. Durante la ejecución del proyecto, el alcance real se mantuvo alineado con los componentes críticos definidos inicialmente; sin embargo, se presentaron ajustes en la profundidad de algunas pruebas de regresión. Estas variaciones se debieron principalmente a restricciones de tiempo y a la priorización consciente de escenarios de mayor riesgo operativo. En lugar de ampliar el alcance a módulos de bajo impacto, se decidió concentrar los esfuerzos en aquellos flujos cuya falla podría comprometer directamente la operación del negocio o la experiencia del usuario final. La diferencia entre el alcance estimado y el alcance real no representa una desviación metodológica, sino una decisión técnica fundamentada en el enfoque de pruebas basado en riesgos. Esta decisión permitió maximizar el valor del proceso de pruebas dentro de las limitaciones del proyecto y obtener resultados relevantes para la toma de decisiones.

### **5.2 Riesgos**

#### Riesgos identificados

Fallo en integración con pasarelas de pago.

No acceso al panel administrador para validación de productos.

*Tabla 9 - Riesgos que se materializaron*

Riesgo	¿Qué ocurrió?	Cómo se abordó
Integración de pago	La pasarela devolvió errores no controlados en ciertos flujos.	Se crearon casos para validar timeouts, rechazos y estados inconsistentes controlados en ciertos flujos.
Validación de productos	Sin acceso al panel para poder validar la publicación de los mismos	Se reportó en el bugtracker el hallazgo

La priorización del alcance final se basó en impacto alto + probabilidad alta según la matriz de riesgos. La matriz de riesgos definida en la etapa de planeación identificó como riesgos principales aquellos asociados a la integración con servicios externos, el manejo de datos sensibles y la correcta sincronización de estados dentro del sistema. Durante la ejecución del proceso de pruebas, varios de estos riesgos se materializaron, especialmente los relacionados con la pasarela de pago y los procesos administrativos de validación de productos.

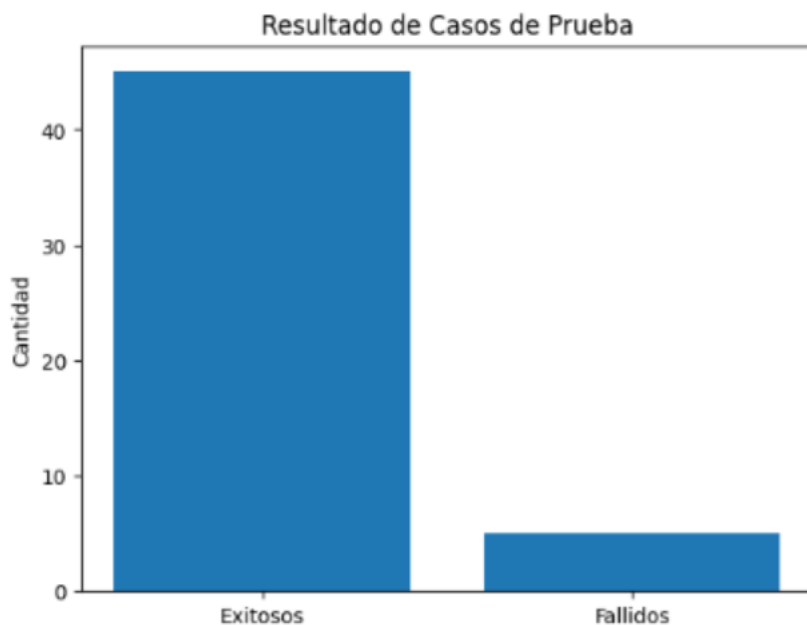
En el caso de la integración con la pasarela de pago, se materializaron riesgos asociados a respuestas inesperadas, tiempos de espera prolongados y manejo inadecuado de errores, lo que derivó en inconsistencias en el estado de los pedidos y del inventario. La causa raíz de estos riesgos se relaciona con una gestión limitada de excepciones y una dependencia directa del comportamiento del servicio externo.

Respecto al manejo de datos sensibles, si bien no se evidenció exposición directa de información crítica, se identificaron comportamientos que podrían derivar en riesgos futuros, como mensajes de error poco controlados. Estos riesgos no se materializaron completamente, pero fueron documentados como alertas preventivas.

Como contingencia, se registraron los defectos en el bug tracker, se documentaron las evidencias correspondientes y se formularon recomendaciones técnicas orientadas a fortalecer el control de errores, la validación de estados y la resiliencia del sistema. De manera transparente, se reconoce que no todos los riesgos pudieron ser mitigados completamente dentro del alcance del proyecto, pero sí fueron identificados y documentados para su gestión posterior.

### **5.3 Métricas del proceso de pruebas**

El proceso de pruebas fue evaluado mediante métricas orientadas a medir el desempeño, la cobertura y la efectividad del testing ejecutado. Entre las métricas consideradas se incluyen el número de casos de prueba ejecutados, la distribución de defectos por severidad y módulo, el nivel de cobertura del alcance definido y la relación entre riesgos identificados y riesgos materializados. Las métricas evidencian una mayor concentración de defectos en los módulos clasificados como de alto riesgo, particularmente en la integración con la pasarela de pago y en los procesos administrativos. Esta distribución valida la efectividad del enfoque de priorización aplicado y demuestra que los esfuerzos de testing se concentraron en las áreas de mayor impacto operativo.



*Figura 17 - Resultado de Casos de Prueba, el alto porcentaje de casos exitosos demuestra la estabilidad del sistema dentro del alcance definido para el plan de pruebas.*

Asimismo, las métricas permiten evidenciar que los módulos de menor criticidad presentaron un menor número de incidencias, lo que refuerza la decisión de no extender el alcance a estos componentes dentro del tiempo disponible. La información recopilada a través de estas métricas se presenta de manera visual mediante tablas y gráficas, facilitando su interpretación por parte del cliente.

#### **5.4 Tiempos del proyecto**

La estimación inicial del esfuerzo se realizó a partir del análisis de riesgos, la complejidad de los módulos y la cantidad de casos de prueba requeridos. Durante la ejecución, los tiempos reales presentaron variaciones moderadas frente a los tiempos estimados, principalmente debido a la atención de defectos críticos y a la ejecución de escenarios no previstos inicialmente, finalmente en la etapa de ejecución se dedicaron 2 días, menos del tiempo programado inicialmente.

Estas variaciones fueron gestionadas mediante ajustes en la planificación y priorización de actividades, evitando afectar el cumplimiento de los objetivos principales del proyecto. La diferencia entre tiempos estimados y reales se considera controlada y coherente con la naturaleza del sistema y con la aparición de incidencias críticas durante la ejecución.

### **5.5 Lecciones Aprendidas**

#### Individual

Mejora en creación de casos de prueba basados en criterios de aceptación.

Mayor dominio en estimación y planificación.

Aprendizaje en validaciones de seguridad (datos sensibles).

#### Equipo

Mejoró la comunicación al definir riesgos tempranos.

La revisión cruzada de casos aumentó la calidad.

El proyecto permitió identificar lecciones aprendidas tanto a nivel de equipo como a nivel individual. Una de las principales lecciones es la importancia de realizar un análisis de riesgos detallado desde etapas tempranas, ya que este insumo resultó determinante para orientar el alcance y la priorización del proceso de pruebas.

Asimismo, se evidenció la relevancia de mantener una documentación clara y estructurada, especialmente en lo relacionado con el registro de defectos y la recolección de evidencias. A nivel individual, el proyecto fortaleció competencias relacionadas con el análisis crítico, la toma de decisiones técnicas y la comunicación efectiva de hallazgos.

## 5.6 Oportunidades de Mejora

### Proceso

Incluir análisis de riesgo desde el inicio.

Formalizar una checklist para validación de seguridad.

### Equipo

Mejorar tiempos de respuesta en revisiones del Administrador.

Documentación más clara sobre cambios de HU.

### Individual

Profundizar en pruebas API y seguridad.

Mejorar precisión en estimaciones (reduce desviación).

A partir de los resultados obtenidos, se identifican oportunidades de mejora tanto a nivel de proceso como a nivel técnico. Entre ellas se destacan la necesidad de fortalecer las pruebas de integración tempranas, mejorar el manejo de excepciones en flujos críticos y ampliar la cobertura de pruebas en escenarios alternos y de error. Desde una perspectiva de proceso, se recomienda incorporar prácticas de automatización parcial para escenarios críticos y fortalecer la planificación de pruebas en iteraciones tempranas del desarrollo. Estas mejoras permitirían optimizar el uso de recursos y reducir la probabilidad de defectos en entornos productivos.

### **5.7 Cierre orientado al cliente**

El presente informe de cierre se entrega como un insumo técnico para apoyar la toma de decisiones del cliente respecto al estado actual de la plataforma NepTune. Los hallazgos identificados permiten priorizar acciones correctivas y definir estrategias de mejora orientadas a fortalecer la estabilidad, seguridad y confiabilidad del sistema.

Si bien la plataforma presenta un comportamiento aceptable en escenarios estándar, los resultados del proceso de pruebas indican la necesidad de atender los defectos críticos identificados antes de considerar su despliegue en un entorno productivo. La información contenida en este informe proporciona una base objetiva y transparente para la planificación de acciones futuras y para la mejora continua del sistema.

## 6. Conclusiones

El desarrollo y ejecución del plan de pruebas para la plataforma NepTune permitió aplicar de manera rigurosa los conocimientos adquiridos durante el seminario, evidenciando la relevancia del aseguramiento de la calidad como un proceso estratégico dentro del desarrollo de software. La adopción de un enfoque basado en riesgos permitió priorizar de forma efectiva las funcionalidades críticas del sistema y optimizar el uso de los recursos disponibles.

Los resultados obtenidos confirman que los componentes identificados como críticos durante la fase de planeación fueron los que presentaron mayor número de incidencias durante la ejecución de las pruebas, lo que valida la coherencia metodológica del proceso adoptado. Asimismo, la gestión estructurada de defectos y la documentación de evidencias garantizaron la trazabilidad y la transparencia del proceso de testing.

A pesar de las limitaciones propias del contexto académico y del alcance definido, el proyecto permitió obtener una visión clara del estado de calidad de los componentes críticos del sistema y generar insumos técnicos relevantes para la toma de decisiones. Los hallazgos identificados ponen de manifiesto la necesidad de fortalecer aspectos como la gestión de integraciones externas, el control de estados y el manejo de escenarios de error antes de considerar el sistema para un entorno productivo.

Finalmente, este trabajo evidencia que un proceso de pruebas bien planificado, ejecutado y analizado constituye un elemento clave para la mejora continua del software. Los aprendizajes obtenidos y los resultados alcanzados aportan valor tanto al proyecto NepTune como a la formación profesional, consolidando competencias transferibles a futuros proyectos de aseguramiento de calidad y desarrollo de software.

## Referencias

1. Abstracta. (2024, 4 de junio). Estrategia de pruebas de software: ¿Cómo crear la adecuada para tu proyecto? <https://es.abstracta.us/blog/guia-crear-estrategia-pruebas-software-adecuada/>
2. Atlassian. (s. f.). Historias de usuario con ejemplos y plantilla. <https://www.atlassian.com/es/agile/project-management/user-stories>
3. Boada, D., & Boada, D. (2024, 28 de agosto). Qué es una pasarela de pago: beneficios y las 7 mejores del mercado. Hostinger. <https://www.hostinger.com/mx/tutoriales/pasarela-de-pago>
4. Comisión de la Verdad Colombia. (s. f.). Datos sensibles. <https://web.comisiondelaverdad.co/transparencia/informacion-de-interes/glosario/datos-sensibles>
5. IBM. (s. f.). ¿Qué son las pruebas de software? <https://www.ibm.com/mx-es/think/topics/software-testing>
6. Porto, J. P. (2023, 24 de noviembre). Análisis de riesgos: Qué son, herramientas, definición y concepto. <https://definicion.de/analisis-de-riesgos/>
7. QAlified. (s. f.). ¿Qué son las pruebas de regresión? <https://qalified.com/es/blog/pruebas-regresion/>
8. web.dev. (2023, 9 de agosto). Definición de casos de prueba y prioridades. <https://web.dev/articles/ta-test-cases>
9. International Software Testing Qualifications Board. (s. f.). ISTQB® glossary of testing terms. <https://glossary.istqb.org/>

10. International Software Testing Qualifications Board. (2018). Certified Tester Foundation Level Syllabus (CTFL). <https://www.istqb.org/certifications/certified-tester-foundation-level>

11. International Software Testing Qualifications Board. (2023). Certified Tester Foundation Level Syllabus (version 4.0). <https://www.istqb.org/certifications/certified-tester-foundation-level>

12. International Software Testing Qualifications Board. (s. f.). Why is software testing important? <https://www.istqb.org/istqb/why-is-testing-important.html>

13. ISO/IEC. (2018). ISO 31000: Risk management — Guidelines. <https://www.iso.org/iso-31000-risk-management.html>