

# Especificación de Requisitos de Software

---

*Proyecto: Rincón Crochetito*

Revisión: [02.00]

[24-11-2025]

## Contenido

<b>FICHA DEL DOCUMENTO</b>	<b>3</b>
<b>1. INTRODUCCIÓN</b>	<b>4</b>
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
<b>2. DESCRIPCIÓN GENERAL</b>	<b>5</b>
2.1. PERSPECTIVA DEL PRODUCTO	5
2.2. FUNCIONES DEL PRODUCTO	5
2.3. CARACTERÍSTICAS DE LOS USUARIOS	5
2.4. RESTRICCIONES	5
2.5. SUPOSICIONES Y DEPENDENCIAS	6
2.6. REQUISITOS FUTUROS	6
<b>3. REQUISITOS ESPECÍFICOS</b>	<b>7</b>
3.1. REQUISITOS COMUNES DE LAS INTERFACES	8
3.1.1 <i>Interfaces de usuario</i>	8
3.1.2 <b>Interfaces de hardware</b>	8
3.1.3 <i>Interfaces de software</i>	8
3.1.4 <i>Interfaces de comunicación</i>	8
3.2. REQUISITOS FUNCIONALES	9
3.3. REQUISITOS NO FUNCIONALES	9
3.3.1 <i>Requisitos de rendimiento</i>	9
3.3.2 <b>Seguridad</b>	10
3.3.3 <i>Fiabilidad</i>	10
3.3.4 <i>Disponibilidad</i>	10
3.3.5 <b>Mantenibilidad</b>	10
3.3.6 <b>Portabilidad</b>	10
3.4. OTROS REQUISITOS	10

**Ficha del documento**

Fecha	Revisión	Autor	Modificación
08-09-2025	01.00	<i>Alejandra Elizabeth Reyes Duque</i>	<i>Versión Inicial ERS Semestral</i>
24-11-2025	02.00	<i>Alejandra Elizabeth Reyes Duque</i>	

Documento validado por las partes en fecha:

Por el cliente

[Firma]

---

Sr./Sra.

Por la empresa suministradora

[Firma]

---

Sr./Sra.

## 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

El propósito de este documento es detallar la Especificación de Requisitos Software (ERS) para el proyecto "Rincón Crochetito", una plataforma de comercio electrónico especializada en productos y recursos para el tejido. Este documento está dirigido principalmente al equipo de desarrollo, a los stakeholders del proyecto, analistas de negocios, futuros usuarios y administradores de la tienda online.

Su objetivo es establecer una comprensión clara y unificada de los requisitos funcionales y no funcionales del software, sirviendo como base para el diseño, desarrollo, pruebas y validación del sistema. Se busca minimizar ambigüedades y asegurar que la tienda online cumpla con las expectativas y necesidades de los clientes, proporcionando una experiencia de compra intuitiva, segura y eficiente para productos relacionados con el crochet, palillos y otras técnicas de tejido.

### 1.2. Ámbito del Sistema

El futuro sistema, denominado "Rincón Crochetito", será una tienda online especializada en la venta de productos y recursos relacionados con el tejido, incluyendo lanas, hilos, agujas de crochet, palillos, patrones, kits de tejido y accesorios.

El sistema permitirá a los usuarios navegar por un catálogo de productos, ver descripciones detalladas, imágenes, precios y añadir artículos al carrito de compras. Se incluirán funcionalidades para el registro de usuarios, gestión de perfiles, historial de pedidos y seguimiento de envíos. Además de la venta de productos físicos, la tienda online podrá ofrecer patrones digitales descargables y acceso a tutoriales premium como valor añadido a la compra de ciertos productos.

"Rincón Crochetito" hará posible que los entusiastas del tejido adquieran fácilmente los materiales y recursos que necesitan para sus proyectos, desde la comodidad de su hogar. Se enfocará en la facilidad de uso, la seguridad en las transacciones y la eficiencia en la gestión de pedidos. Sin embargo, en esta fase inicial, el sistema no incluirá funcionalidades de personalización avanzada de productos (ej. diseño de patrones a medida), ni foros de discusión interactivos o funcionalidades de comunidad que no estén directamente ligadas al proceso de compra. Tampoco se contempla la integración con sistemas de inventario de terceros más allá de la gestión interna de stock.

Los beneficios de "Rincón Crochetito" incluyen la conveniencia para los clientes al acceder a una amplia gama de productos especializados en un solo lugar, la expansión del alcance de la marca a nivel nacional, y la optimización de los procesos de venta y distribución. Los objetivos principales son: convertirse en la tienda online de referencia para productos de tejido en el mercado, aumentar las ventas y la base de clientes, y mejorar la satisfacción del cliente a través de una experiencia de compra

fluida y un servicio de calidad. La meta es que "Rincón Crochetito" sea reconocido como el destino preferido para tejedores que buscan productos de alta calidad y recursos inspiradores.

### 1.3. Definiciones, Acrónimos y Abreviaturas

- **ERS:** Especificación de Requerimientos de Software. Documento que describe los requisitos funcionales y no funcionales de un sistema de software.
- **Rincón Crochetito:** Nombre del proyecto de software, una plataforma de comercio electrónico especializada en productos y recursos para el tejido.
- **Cliente:** Usuario que realiza compras en la tienda online "Rincón Crochetito".
- **Producto:** Artículo ofrecido para la venta en la tienda online, que puede ser físico (lanas, agujas) o digital (patrones descargables).
- **Carrito de Compras:** Funcionalidad que permite a los clientes seleccionar y almacenar temporalmente los productos que desean adquirir antes de finalizar la compra.
- **Pasarela de Pago:** Servicio que autoriza pagos electrónicos para negocios de comercio electrónico, tiendas online, etc.
- **Catálogo de Productos:** Colección organizada de todos los productos disponibles en la tienda online, con sus respectivas descripciones, imágenes y precios.
- **Gestión de Pedidos:** Proceso que abarca desde la recepción de un pedido hasta su entrega al cliente, incluyendo el procesamiento, empaquetado y envío.
- **Inventario:** Cantidad de productos disponibles en stock para la venta.
- **SEO:** Search Engine Optimization (Optimización para Motores de Búsqueda). Conjunto de técnicas para mejorar la visibilidad de un sitio web en los resultados orgánicos de los buscadores.
- **UI/UX:** Interfaz de Usuario / Experiencia de Usuario. Se refiere al diseño de la interacción y la experiencia del usuario con la tienda online.
- **API:** Interfaz de Programación de Aplicaciones. Conjunto de definiciones y protocolos que se utilizan para diseñar y construir software de aplicaciones.
- **Base de Datos (BD):** Colección organizada de información o datos estructurados, típicamente almacenados electrónicamente en un sistema de computadora.

### 1.4. Referencias

- **IEEE Std 830-1998:** Recommended Practice for Software Requirements Specifications. Estándar de la IEEE para la creación de ERS, utilizado como guía para la estructura y contenido de este documento.
- **Documento de Visión del Proyecto "Rincón Crochetito" (Tienda Online):** Documento interno que describe la visión general, los objetivos de alto nivel y el alcance inicial de la tienda online.
- **Actas de Reuniones con Stakeholders:** Registros de discusiones y decisiones clave con interesados del proyecto.
- **Análisis de Competencia y Benchmarking:** Estudios de otras tiendas online de productos de tejido o manualidades.

- Guías de Usabilidad y Experiencia de Usuario para E-commerce.
- Regulaciones de Comercio Electrónico y Protección de Datos (Ley chilena de datos personales y GDPR si aplica).

## 1.5. Visión General del Documento

Este documento está estructurado para facilitar la comprensión de los requisitos de la tienda online "Rincón Crochetito" de manera lógica y progresiva.

Áreas principales:

- **Definición del Negocio:** Contexto, propósito, ámbito del sistema, definiciones y referencias.
- **Especificación de Requisitos:** Requisitos funcionales y no funcionales del sistema, incluyendo catálogo, carrito de compras, gestión de pedidos y atributos de calidad (rendimiento, seguridad, usabilidad).
- **Anexos:** Formularios de caso de uso que detallan las interacciones entre usuarios y el sistema.

## 2. Descripción General

### 2.1. Perspectiva del Producto

El sistema "Rincón Crochetito" se concibe como una tienda online independiente, pero que se integrará con diversos sistemas externos:

- **Pasarelas de Pago:** Transbank, PayPal.
- **Sistemas de Envío/Logística:** Chilexpress, Correos de Chile.
- **Plataformas de Email Marketing/Notificaciones:** Mailchimp, SendGrid.
- **Herramientas de Análisis Web:** Google Analytics.
- **Redes Sociales (Opcional):** Instagram, Facebook.

Esta perspectiva asegura que "Rincón Crochetito" funcione como un ecosistema conectado, aprovechando servicios especializados para ofrecer una experiencia de compra completa y eficiente.

### 2.2. Funciones del Producto

#### Gestión de Catálogo de Productos:

- Añadir, editar y eliminar productos.
- Categorías y subcategorías.
- Control de inventario automático.

#### Experiencia del Cliente:

- Navegación y búsqueda con filtros.
- Detalle de producto (imágenes, descripciones, reseñas).
- Carrito de compras con edición de cantidades.
- Proceso de compra seguro (checkout).

#### Gestión de Usuarios y Cuentas:

- Registro y autenticación.
- Perfil de usuario (datos personales, direcciones, historial).

#### Gestión de Pedidos y Pagos:

- Procesamiento y actualización de estado.
- Integración con pasarelas de pago.

#### Módulo de Administración (Backend):

- Dashboard de control.
- Gestión de contenido (estático).
- Gestión de clientes.

### 2.3. Características de los Usuarios

#### Cliente (Usuario General):

Es el usuario final que navega por la tienda, busca productos, realiza compras y gestiona su propia cuenta. Se espera que cuente con experiencia básica en navegación web y en el uso de plataformas de compras en línea, sin requerir conocimientos técnicos avanzados.

#### Administrador de la Tienda (roles de sistema: SUPERADMIN / ADMIN):

Es el responsable de la gestión completa de la plataforma, incluyendo la administración de productos, pedidos, clientes, contenido y configuración del sistema. Se recomienda que cuente con educación media o superior, experiencia en comercio electrónico y conocimientos intermedios en el uso de PC y software de gestión (por ejemplo, Excel a nivel medio).

#### Gestor de Pedidos (Rol Operativo (rol de sistema: VENDEDOR)):

Es un perfil con permisos limitados, centrado en la gestión operativa de pedidos. Sus tareas incluyen actualizar el estado de los pedidos, generar etiquetas de envío y gestionar devoluciones. Se recomienda que cuente con educación media, experiencia previa en logística o gestión de almacenes, y conocimientos básicos en el uso de PC.

### 2.4. Restricciones

Además de las restricciones ya señaladas en el ámbito y la perspectiva del producto, se consideran las siguientes limitaciones que condicionan el diseño, desarrollo y operación del sistema:

- **Operaciones Paralelas:** el sistema debe ser capaz de manejar múltiples transacciones y usuarios concurrentes sin degradación significativa del rendimiento. Esto incluye la capacidad de procesar pedidos simultáneamente, mantener sesiones activas de distintos usuarios y actualizar el inventario en tiempo real.
- **Funciones de Auditoría:** se debe implementar un registro de actividades clave, tales como transacciones de compra, modificaciones en el inventario y acciones realizadas por los administradores. Estos registros son necesarios para la trazabilidad, la resolución de incidencias y el cumplimiento normativo.
- **Funciones de Control:** el sistema debe contar con controles de acceso basados en roles, de manera que solo los usuarios autorizados puedan ejecutar determinadas operaciones (ejemplo: solo administradores pueden modificar precios o añadir productos). Asimismo, se deben establecer validaciones en los datos de entrada y mecanismos de detección de intentos de fraude.
- **Criticidad de la Aplicación:** la tienda online es una aplicación de alta importancia para los ingresos y la reputación de la empresa. Por ello, se requiere un nivel elevado de fiabilidad y la capacidad de recuperación rápida frente a fallos, de manera de minimizar impactos económicos o de confianza de los clientes.

## 2.5. Suposiciones y Dependencias

- **Disponibilidad y Estabilidad de Pasarelas de Pago y Servicios de Envío:** se asume que Transbank, PayPal, Chilexpress y Correos de Chile mantienen SLA estables.
- **Conectividad a Internet del Usuario:** se asume que los clientes cuentan con conexión estable y suficiente.
- **Actualizaciones de Navegadores:** compatibilidad garantizada con las 2 versiones más recientes de Chrome, Firefox, Edge y Safari.
- **Disponibilidad de Servidores y Hosting:** el proveedor asegura alta disponibilidad y escalabilidad.
- **Disponibilidad de Personal Capacitado:** habrá personal entrenado para la gestión diaria y el soporte.
- **Estrategia de Marketing y Ventas Activa:** se asume tráfico cualificado a través de campañas.
- **Calidad de Datos de Productos:** descripciones, imágenes y stock serán precisos y estarán actualizados.

## 2.6. Requisitos Futuros

- Personalización avanzada de productos.
- Módulo de Reseñas y Calificaciones de Productos mejorado (reseñas con fotos y calificaciones detalladas).
- Integración con redes sociales.
- Programas de fidelización y recompensas.
- Recomendaciones basadas en IA.
- Wishlist.
- Blog o contenido educativo.

- Soporte multilingüe y multidivisa.
- Integración con ERP/Contabilidad (sincronización de ventas, inventario y finanzas).
- Aplicación móvil nativa.
- Chatbot de atención al cliente.

### 3. Requisitos Específicos

Esta sección contiene los requisitos a un nivel de detalle suficiente como para permitir a los diseñadores diseñar un sistema que satisfaga estos requisitos, y que permita al equipo de pruebas planificar y realizar las pruebas que demuestren si el sistema satisface, o no, los requisitos. Todo requisito aquí especificado describirá comportamientos externos del sistema, perceptibles por parte de los usuarios, operadores y otros sistemas. Esta es la sección más larga e importante de la ERS. Deberán aplicarse los siguientes principios:

- El documento debería ser perfectamente legible por personas de muy distintas formaciones e intereses.
- Deberán referenciarse aquellos documentos relevantes que poseen alguna influencia sobre los requisitos.
- Todo requisito deberá ser únicamente identificable mediante algún código o sistema de numeración adecuado.
- Lo ideal, aunque en la práctica no siempre realizable, es que los requisitos posean las siguientes características:
  - **Corrección:** La ERS es correcta si y sólo si todo requisito que figura aquí (y que será implementado en el sistema) refleja alguna necesidad real. La corrección de la ERS implica que el sistema implementado será el sistema deseado.
  - **No ambiguos:** Cada requisito tiene una sola interpretación. Para eliminar la ambigüedad inherente a los requisitos expresados en lenguaje natural, se deberán utilizar gráficos o notaciones formales. En el caso de utilizar términos que, habitualmente, poseen más de una interpretación, se definirán con precisión en el glosario.
  - **Completos:** Todos los requisitos relevantes han sido incluidos en la ERS. Conviene incluir todas las posibles respuestas del sistema a los datos de entrada, tanto válidos como no válidos.
  - **Consistentes:** Los requisitos no pueden ser contradictorios. Un conjunto de requisitos contradictorios no es implementable.
  - **Clasificados:** Normalmente, no todos los requisitos son igual de importantes. Los requisitos pueden clasificarse por importancia (esenciales, condicionales u opcionales) o por estabilidad (cambios que se espera que afecten al requisito). Esto sirve, ante todo, para no emplear excesivos recursos en implementar requisitos no esenciales.
  - **Verificables:** La ERS es verificable si y sólo si todos sus requisitos son verificables. Un requisito es verificable (testable) si existe un proceso finito y no costoso para demostrar que el sistema

cumple con el requisito. Un requisito ambiguo no es, en general, verificable. Expresiones como a veces, bien, adecuado, etc. Introducen ambigüedad en los requisitos. Requisitos como "en caso de accidente la nube tóxica no se extenderá más allá de 25Km" no es verificable por el alto costo que conlleva.

- **Modificables:** La ERS es modificable si y sólo si se encuentra estructurada de forma que los cambios a los requisitos pueden realizarse de forma fácil, completa y consistente. La utilización de herramientas automáticas de gestión de requisitos facilitan enormemente esta tarea.
- **Trazables:** La ERS es trazable si se conoce el origen de cada requisito y se facilita la referencia de cada requisito a los componentes del diseño y de la implementación. La trazabilidad hacia atrás indica el origen (documento, persona, etc.) de cada requisito. La trazabilidad hacia delante de un requisito R indica qué componentes del sistema son los que realizan el requisito R.

### 3.1 Requisitos comunes de las interfaces

Descripción detallada de todas las entradas y salidas del sistema de software.

#### 3.1.1 Interfaces de usuario

- **Web responsive (desktop/tablet/móvil)** con navbar superior, buscador, breadcrumbs, cards de productos y footer.
- **Páginas mínimas:** Home, Catálogo (lista + filtros), Detalle, Carrito, Checkout (datos envío/pago), Registro/Login, Perfil (datos y direcciones), Mis pedidos, Admin (CRUD productos), Políticas y Contacto.
- **Usabilidad:** validaciones en formularios con mensajes claros, estados vacíos, loaders, confirmaciones, manejo de errores en contexto.
- **Accesibilidad (AA básica):** etiquetas semánticas, foco visible, texto alternativo en imágenes, contraste suficiente y navegación por teclado.
- **Idioma y formato local:** español (Chile), moneda CLP, fechas dd-mm-aaaa.

#### 3.1.2 Interfaces de hardware

- Compatible con **PC** (teclado/mouse) y **dispositivos táctiles móviles** (gestos, teclado virtual).
- No requiere periféricos especiales; recomendado pantalla  $\geq 360\text{px}$  de ancho para móviles.

#### 3.1.3 Interfaces de software

**Frontend:** React (SPA).

**Backend:** Spring Boot (API REST).

**BD:** H2 (dev) / MySQL (prod/demo).

**Formato de intercambio:** JSON UTF-8; timezone America/Santiago.

**Autenticación:** JWT en header Authorization: Bearer {token}.

**Errores:** JSON con `timestamp`, `status`, `error`, `message`, `path`.

**Ejemplos de endpoints (ET2–ET3):**

- `GET /api/productos?buscar=&page=&size=` – listado/paginación.
- `GET /api/productos/{id}` – detalle.
- `POST /api/carrito/items` – agregar ítem (auth).
- `POST /api/ordenes` – crear orden (auth).
- `GET /api/ordenes/{id}` – ver estado.
- `GET/POST/PUT/DELETE /api/admin/productos` – CRUD (rol ADMIN).

### 3.1.4 Interfaces de comunicación

- **Protocolo:** HTTP/HTTPS.
- **CORS:** habilitado solo para el dominio del frontend.
- **Seguridad en tránsito:** TLS 1.2+ en ambientes publicados.
- **Timeouts cliente:** 10 s para catálogo/detalle; 20 s para checkout.
- **Reintentos:** el frontend reintenta 1 vez las solicitudes GET ante errores 5xx; POST no se reintenta.

## 3.2 Requisitos funcionales

### RF-01 Catálogo de productos

- **CA1 (ET1):** al abrir “Catálogo” se muestran  $\geq 6$  productos en formato card con CTA “Agregar”.
- **CA2 (ET2–ET3):** `GET /api/productos` responde **200** con JSON paginado (campos: `content`, `totalElements`, `number`, `size`).
- **CA3 (ET2–ET3):** búsqueda por texto retorna solo coincidencias en nombre o categoría.
- **CA4 (ET2–ET3):** p95 del endpoint de listado  $\leq 1.5$  s con 20–100 ítems.

### RF-02 Detalle de producto

- **CA1 (ET1):** al abrir un producto se muestran **nombre**, **precio**, **stock**, **imagen** y **descripción**.
- **CA2 (ET2–ET3):** `GET /api/productos/{id}`
  - 200 con JSON válido si existe;
  - **404** con mensaje **Producto no encontrado** si no existe.
- **CA3 (ET2–ET3):** si `stock = 0`, el botón “Agregar al carrito” aparece deshabilitado.

#### RF-03 Carrito – Agregar ítem

- **CA1 (ET1):** al presionar “Agregar”, el ítem aparece en el carrito y **persiste** tras recargar (LocalStorage).
- **CA2 (ET2–ET3):** al agregar ítem autenticado, el carrito persiste en BD y se recupera tras relogueo.
- **CA3 (ET2–ET3):** no permite agregar cantidades que excedan **stock** (respuesta 400 con mensaje claro).

#### RF-04 Carrito – Editar/Eliminar ítem

- **CA1 (ET1):** cambiar cantidad actualiza subtotal y total **en tiempo real**; eliminar quita el ítem del listado.
- **CA2 (ET2–ET3):** edición/eliminación sincroniza con servidor y refleja cambios al recargar.
- **CA3 (ET2–ET3):** si la cantidad editada supera stock, se restaura último valor válido y se muestra aviso.

#### RF-05 Totales y descuento (10%)

- **CA1 (ET1):** si el usuario está “logueado” (simulado) se aplica **10%** y el total muestra redondeo a 2 decimales.
- **CA2 (ET2–ET3):** cálculo de totales se realiza **server-side**; respuesta incluye **subtotal**, **descuento**, **total**.
- **CA3 (ET2–ET3):** no se aplica descuento a usuario no autenticado; códigos de cupón (si se usan) se validan y devuelven error 400 cuando corresponda.

#### RF-06 Registro/Login

- **CA1 (ET1):** formulario valida campos requeridos y formato de email; muestra mensajes de error en línea.
- **CA2 (ET2–ET3):** login exitoso retorna **JWT**; rutas protegidas exigen **Authorization : Bearer {token}**.
- **CA3 (ET2–ET3):** sesión expirada produce **401** y redirección a login con mensaje de sesión vencida.
- **CA4 (ET2–ET3):** flujo de refresh token renueva token sin reingresar credenciales (si implementado).

#### RF-07 Perfil de usuario y pedidos (autenticado)

- **CA1 (ET2–ET3):** GET /api/usuarios/me retorna 200 con datos del perfil (nombre, email) y direcciones.
- **CA2 (ET2–ET3):** PUT /api/usuarios/me actualiza perfil validando formato (email) y campos obligatorios; errores 400 con detalle.

- **CA3 (ET2–ET3):** GET /api/ordenes?usuario=me lista órdenes del usuario autenticado con paginación.
- **CA4 (ET2–ET3):** POST /api/usuarios/direcciones crea/edita/elimina direcciones; requiere auth y valida comuna/CP.

#### RF-08 Checkout / Órdenes

- **CA1 (ET1):** “Confirmar compra” crea orden simulada y **vacía** el carrito; se muestra número de orden.
- **CA2 (ET2–ET3):** POST /api/ordenes crea orden con estado inicial **CREADA** y devuelve resumen (ítems, totales, dirección).
- **CA3 (ET2–ET3):** GET /api/ordenes/{id} muestra estados válidos (**CREADA, PAGADA, ENVIADA, ENTREGADA**).
- **CA4 (ET2–ET3):** si el carrito está vacío, el checkout **no** permite continuar y muestra aviso.

#### RF-09 Admin – CRUD productos (Crear/Editar/Eliminar)

- **CA1 (ET1):** altas/ediciones mock rechazan campos vacíos obligatorios (nombre, precio, stock).
- **CA2 (ET2–ET3):** endpoints admin requieren rol **ADMIN**; accesos sin rol responden **403**.
- **CA3 (ET2–ET3):** validaciones: **precio ≥ 0, stock ≥ 0**; errores con JSON de validación detallado.
- **CA4 (ET2–ET3):** listado admin admite paginación (**page, size**) y ordenamiento por nombre/precio.

#### RF-10 Admin – Publicación/Visibilidad

- **CA1 (ET2–ET3):** un producto con **activo=false** **no** aparece en el catálogo público.
- **CA2 (ET2–ET3):** cambios de visibilidad se reflejan en el front tras recargar o invalidar caché.
- **CA3 (ET2–ET3):** historial de cambios registra usuario y timestamp de la acción (log/auditoría).

### 3.3 Requisitos no funcionales

#### 3.3.1 Requisitos de rendimiento

- **Listado catálogo:** p95 ≤ **1.5 s** (20–100 ítems).
- **Detalle producto:** p95 ≤ **800 ms**.
- **Búsqueda/filtros locales:** p95 ≤ **500 ms** con 100 ítems.
- **Operaciones de carrito (UI):** actualización ≤ **150 ms**.
- **CRUD admin (API):** p95 ≤ **300 ms** en red local de demo.
- **Imágenes:** tamaño individual ≤ **300 KB** (miniaturas) y ≤ **1 MB** (detalle).
- **Concurrencia de referencia:** **20** usuarios simultáneos en demo sin caída de p95.

### 3.3.2 Seguridad

- **Autenticación/Autorización:** JWT (login/refresh), rutas protegidas por rol (CLIENTE/ADMIN).
- **Datos en tránsito:** HTTPS; cookies httpOnly (si se usa cookie) o header Authorization: Bearer {token}.
- **Validación y sanitización:** Bean Validation en backend; escape de HTML para inputs.
- **CORS y CSRF:** CORS restringido a dominio del front; CSRF mitigado (según estrategia JWT).
- **Gestión de errores:** sin stack traces en respuestas; códigos 4xx/5xx correctos.
- **Logs y auditoría:** altas/bajas/cambios en productos, creación/estado de órdenes y accesos admin.
- **Pagos reales (si se activa):** cumplimiento PCI DSS y nunca almacenar PAN/CVV.

### 3.3.3 Fiabilidad

- **Integridad transaccional:** operaciones críticas (orden, stock) con transacciones atómicas.
- **Reintentos técnicos:** una re-ejecución para llamadas idempotentes GET ante 5xx.
- **MTBF (objetivo académico):** sin fallos críticos durante **8 h** de demo.
- **Backups (ET3 si aplica):** snapshot BD diario; **RPO ≤ 24 h**.

### 3.3.4 Disponibilidad

- **Objetivo académico:**  $\geq 99\%$  durante ventanas de evaluación.
- **Mantenimiento planificado:** fuera de horario hábil (p.ej., 22:00–06:00).
- **RTO ≤ 30 min y RPO ≤ 5 min** en demo final (si se configura respaldo).

### 3.3.5 Mantenibilidad

- **Arquitectura:** backend en capas (controller/service/repository); front modular por componentes.
- **Código:** linters, formateo, y convenciones de nombres; cobertura de pruebas unitarias objetivo  $\geq 60\%$  en módulos clave.
- **Repositorios:** GitHub (feature → PR → main), revisión por pares.
- **Documentación:** README con instrucciones de build/run, variables de entorno y scripts.

### 3.3.6 Portabilidad

- **Backend:** ejecutable JAR; **Dockerfile** opcional para despliegue.
- **Frontend:** build estándar de React (estático).
- **Navegadores soportados:** últimas **2** versiones de Chrome, Edge, Firefox y Safari.
- **BD intercambiable:** perfil H2 (dev) / MySQL (demo/prod) sin cambios de código.

### 3.4 Otros Requisitos

- **Accesibilidad:** cumplir pautas WCAG 2.1 nivel AA básico (foco, contraste, texto alternativo, navegación teclado).
- **Analítica:** integración con Google Analytics respetando consentimiento y privacidad.
- **Localización:** formatos locales (CLP, zona horaria America/Santiago).
- **Políticas:** mostrar Términos y Condiciones, Política de Privacidad y de Devoluciones accesibles desde el footer.
- **Consentimiento de cookies:** banner de consentimiento y preferencia básica para analítica.

Hito	Alcance técnico	Entregables clave
<b>ET1 (30%)</b>	Prototipo con <b>JSON Server + LocalStorage</b> ; flujo catálogo→carrito→checkout simulado; registro/login simulado.	ERS v1 (este), mock API, UI básica, README de arranque.
<b>ET2 (30%)</b>	<b>React + Spring Boot</b> (parcial); <b>JWT</b> básico; BD real; CRUD parcial de admin.	ERS v2 (evolución), endpoints implementados, pruebas unitarias iniciales.
<b>ET3 (40%)</b>	API completa + seguridad y roles; pruebas (front/back); performance y documentación final.	ERS v3, manual de usuario, colección Postman, despliegue local/documentado.