

# Especificación de Requisitos de Software

---

*Proyecto: Dulce Vida*

Revisión: [99.99]

01/09/2025

## 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.....	5
1.4. REFERENCIAS .....	ERROR! BOOKMARK NOT DEFINED.
1.5. VISIÓN GENERAL DEL DOCUMENTO .....	6
<b>2. DESCRIPCIÓN GENERAL.....</b>	<b>7</b>
2.1. PERSPECTIVA DEL PRODUCTO.....	7
2.2. FUNCIONES DEL PRODUCTO.....	7
2.3. CARACTERÍSTICAS DE LOS USUARIOS .....	8
2.4. RESTRICCIONES.....	ERROR! BOOKMARK NOT DEFINED.
2.5. SUPOSICIONES Y DEPENDENCIAS.....	ERROR! BOOKMARK NOT DEFINED.
2.6. REQUISITOS FUTUROS.....	ERROR! BOOKMARK NOT DEFINED.
<b>3. REQUISITOS ESPECÍFICOS .....</b>	<b>10</b>
3.1 REQUISITOS COMUNES DE LAS INTERFACES.....	10
3.1.1 <i>Interfaces de usuario</i> .....	10
3.1.2 <i>Interfaces de hardware</i> .....	10
3.1.3 <i>Interfaces de software</i> .....	11
3.1.4 <i>Interfaces de comunicación</i> .....	11
3.2 REQUISITOS FUNCIONALES.....	11
3.3 REQUISITOS NO FUNCIONALES .....	11
3.3.1 <i>Requisitos de rendimiento</i> .....	13
3.3.2 <i>Seguridad</i> .....	13
3.3.3 <i>Fiabilidad</i> .....	14
3.3.4 <i>Disponibilidad</i> .....	14
3.3.5 <i>Mantenibilidad</i> .....	14
3.3.6 <i>Portabilidad</i> .....	15
3.4 OTROS REQUISITOS.....	ERROR! BOOKMARK NOT DEFINED.

## Ficha del documento

Fecha	Revisión	Autor	Modificación
07.09.2025	0.0	<i>Ignacio Silva</i>	<i>Proyecto Terminado.</i>
08.09.2025	0.1	<i>Ignacio Silva</i>	<i>Eliminación de todos los comentarios del código.</i>

Documento validado por las partes en fecha: [Fecha]

Por el cliente

[Firma]

Por la empresa suministradora

[Firma]

---

Sr./Sra. Angela Andrea Carvajal

---

Sr./Sra. Ignacio Silva

## 1. Introducción

### 1.1. Propósito

El propósito de este documento es definir de manera clara y detallada todos los requisitos del sistema de comercio electrónico "DulceVida". Fue creado para servir como una guía fundamental y un acuerdo mutuo entre el desarrollador del proyecto y la dueña de la pastelería.

**Este documento tiene dos objetivos principales:**

Para la dueña de la pastelería (la cliente): Servir como una descripción precisa de lo que el sistema hará, permitiéndole validar que la plataforma web cumple con todas sus necesidades y expectativas para digitalizar su negocio, facilitar la toma de pedidos y expandir sus ventas.

Para el desarrollador: Funcionar como el plano de construcción del software, asegurando que todas las funcionalidades, características y restricciones estén documentadas. Esto garantiza que el producto final sea exactamente lo que la clienta solicitó y necesita para operar su tienda en línea de manera eficiente.

### 1.2. Ámbito del Sistema

**El software "DulceVida" será una aplicación web completa que permitirá las siguientes acciones:**

**Para el Cliente:**

- Crear una cuenta de usuario personal con sus datos de contacto y dirección.
- Iniciar sesión para gestionar sus datos y realizar compras.
- Navegar por un catálogo de productos de pastelería, organizados visualmente en una grilla.
- Filtrar los productos por categorías para facilitar la búsqueda.
- Ver los detalles de un producto, incluyendo imagen, descripción e ingredientes.
- Añadir productos a un carrito de compras, especificando la cantidad y notas personalizadas para el pedido.
- Finalizar la compra a través de un proceso de checkout, donde podrá elegir entre delivery a domicilio o retiro en tienda.
- Aplicar códigos de descuento para obtener beneficios en el total de la compra.

**Para la Administradora:**

- Acceder a un panel de administración protegido para gestionar la tienda.
- Visualizar un dashboard con indicadores clave, como el número total de productos y usuarios registrados.

- Gestionar el catálogo de productos (añadir, editar y eliminar) de forma simulada desde el panel.
- Administrar la base de datos de usuarios, pudiendo crear nuevos perfiles o eliminarlos.

**El sistema NO contemplará:**

- Las siguientes funcionalidades quedan explícitamente fuera del alcance de esta versión del proyecto:**
- Gestión de inventario en tiempo real de productos o materias primas.
  - Gestión de pagos en efectivo o contra entrega.
  - Un historial de pedidos consultable por el cliente desde su cuenta.
  - Sistema de seguimiento del estado del delivery en tiempo real para el cliente.

**Beneficios esperados:**

- Facilitar el proceso de compra: Permitir a los clientes realizar pedidos desde cualquier lugar y en cualquier momento.
- Automatizar la gestión de pedidos: Centralizar la recepción de órdenes, reduciendo el tiempo dedicado a la atención manual por teléfono o mensajería.
- Incrementar las ventas: Abrir un nuevo canal de venta digital para alcanzar a más clientes.
- Mejorar el control: Tener un registro digital de las órdenes (simulado) y de los clientes, lo que facilita la gestión del negocio.

### 1.3. Definiciones, Acrónimos y Abreviaturas

**CLP (Peso Chileno):** Abreviatura para el Peso Chileno, la moneda utilizada para todos los precios en el sistema.

**CRUD:** Acrónimo de Create, Read, Update, Delete (Crear, Leer, Actualizar, Borrar). Se refiere a las operaciones básicas de gestión de datos en una base de datos o sistema.

**CSS (Cascading Style Sheets):** Hojas de Estilo en Cascada. Es el lenguaje utilizado para describir la presentación y el diseño visual de las páginas web.

**HTML (HyperText Markup Language):** Lenguaje de Marcado de Hipertexto. Es el lenguaje estándar utilizado para crear y estructurar el contenido de las páginas web.

**JS (JavaScript):** Lenguaje de programación que permite implementar la lógica, interactividad y funcionalidades dinámicas en las páginas web del proyecto.

**JSON (JavaScript Object Notation):** Formato de texto ligero para el intercambio de datos. Es el formato en que la API entrega la información de productos, usuarios y órdenes.

**RUN (Rol Único Nacional):** Número de identificación único para los residentes en Chile, solicitado y validado en el formulario de registro.

**UI (User Interface):** Interfaz de Usuario. Es el medio visual a través del cual el usuario interactúa con la aplicación web.

## 1.5. Visión General del Documento

Este documento está estructurado para guiar tanto a la cliente como al desarrollador a través de todos los aspectos del sistema "DulceVida", desde su concepto general hasta los detalles más específicos de su funcionamiento.

- **Sección 1 - Introducción:** Presenta el propósito, el alcance, las definiciones clave y la organización de este documento, estableciendo el contexto inicial del proyecto.
- **Sección 2 - Descripción General:** Ofrece una vista panorámica del sistema, describiendo su relación con otros productos, sus funciones principales, las características de los usuarios a los que se dirige (cliente y administradora) y las restricciones generales que lo afectan. Esta sección ayuda a entender el "porqué" del sistema antes de entrar en el "qué".
- **Sección 3 - Requisitos Específicos:** Es la sección más detallada y fundamental del documento. Aquí se desglosan todos los requisitos que el software debe cumplir, divididos en **requisitos funcionales** (lo que el sistema *hace*, como registrar un usuario o procesar un carrito) y **requisitos no funcionales** (cómo el sistema *es*, en términos de rendimiento, seguridad y usabilidad).

## 2. Descripción General

### 2.1. Perspectiva del Producto

El sistema "DulceVida" es una plataforma de comercio electrónico totalmente independiente y autocontenido. Ha sido desarrollado como una solución completa y no como un módulo que se integra a un sistema empresarial más grande.

**Independencia Operativa:** El software no se conecta ni depende de sistemas externos de la pastelería, como un sistema de inventario (ERP) o un sistema de contabilidad. Toda la gestión de productos, precios y usuarios se realiza dentro de la propia aplicación, a través del panel de administración.

**Flujo de Pago Interno:** El proceso de compra culmina con el registro de la orden directamente en la base de datos del sistema. En su versión actual, el sistema gestiona el ciclo completo del pedido sin interactuar con pasarelas de pago de terceros, registrando la transacción localmente para el control administrativo de la dueña.

### 2.2. Funciones del Producto

En esta subsección de la ERS se mostrará un resumen, a grandes rasgos, de las funciones del futuro sistema. Las funciones deberán mostrarse de forma organizada, y pueden utilizarse gráficos, siempre y cuando dichos gráficos reflejen las relaciones entre funciones y no el diseño del sistema. (Se recomienda algún tipo de Diagrama de los componentes del sistema)

El sistema "DulceVida" está diseñado para ofrecer una experiencia de compra completa para el cliente y herramientas de gestión esenciales para la administradora. Sus funciones principales se pueden agrupar en dos grandes módulos: el Módulo de Cliente (Tienda Virtual) y el Módulo de Administración (Panel de Control).

**A continuación, se presenta un resumen de las funciones clave:**

Módulo de Cliente (Tienda Virtual):

-Navegación y Descubrimiento: Permite a los usuarios explorar el catálogo completo de productos, usar filtros por categoría y ver los detalles de cada artículo, incluyendo sus ingredientes.

-Gestión de Cuentas: Ofrece la funcionalidad para que los clientes se registren en la plataforma y puedan iniciar sesión para una experiencia de compra personalizada.

-Carrito de Compras Interactivo: Los usuarios pueden añadir productos a su carrito, modificar las cantidades, eliminar artículos y aplicar cupones de descuento válidos.

-Proceso de Finalización de Compra (Checkout): Guía al usuario a través de los pasos finales para confirmar su pedido, incluyendo la selección del método de entrega (delivery o retiro) y la introducción de datos de contacto y dirección.

-Comunicación: Incluye un formulario de contacto para que los clientes puedan enviar consultas directamente a la administradora.

#### Módulo de Administración (Panel de Control):

-Acceso Restringido: Protege el panel de gestión para que solo usuarios con el rol "admin" puedan acceder.

-Dashboard Informativo: Proporciona una vista rápida del estado de la tienda con indicadores clave (KPIs) sobre el total de productos y usuarios registrados.

-Gestión de Catálogo: Permite a la administradora realizar operaciones CRUD (Crear, Leer, Actualizar, Borrar) sobre los productos de la tienda, gestionando así la oferta disponible para los clientes.

-Gestión de Usuarios: Ofrece herramientas para administrar las cuentas de los usuarios registrados en la plataforma.

### 2.3. Características de los Usuarios

Esta subsección describe los perfiles de los usuarios que interactuarán con el sistema "DulceVida". Para este proyecto, se han definido dos roles principales con características y objetivos distintos:

#### 1. Cliente Final (Rol: usuario)

- **Descripción General:** Es el consumidor que desea comprar productos de la pastelería. Representa el público general y es el usuario principal de la tienda virtual.
- **Experiencia Técnica:** Se asume un **nivel de usuario básico**. El cliente debe estar familiarizado con la navegación web estándar, el uso de aplicaciones móviles y los procesos de compra en línea (añadir al carrito, llenar formularios, etc.).
- **Objetivos en el Sistema:**
  - Explorar el catálogo de productos de manera sencilla.
  - Realizar pedidos de forma rápida y sin complicaciones.
  - Crear una cuenta para guardar sus datos y facilitar futuras compras.
  - Contactar a la pastelería en caso de dudas.

#### 2. Administradora (Rol: admin)

- **Descripción General:** Es la dueña y operadora de la pastelería, responsable de gestionar todo el contenido y la operativa de la tienda en línea.
- **Experiencia Técnica:** Se requiere un **nivel de usuario básico-intermedio de PC**. Debe sentirse cómoda utilizando un navegador web, llenando formularios en línea y comprendiendo datos presentados en

tablas y paneles (similar a manejar una hoja de cálculo como Excel a nivel medio). No se requieren conocimientos de programación.

- **Objetivos en el Sistema:**

- Mantener el catálogo de productos actualizado (añadir, editar o eliminar pasteles, tortas, etc.).
- Gestionar precios y la información visible de los productos.
- Supervisar la actividad de la tienda a través de los indicadores del dashboard.
- Administrar las cuentas de los clientes registrados.

### 3. Requisitos Específicos

#### 3.1 Requisitos comunes de las interfaces

esta subsección detalla todos los puntos de interacción del sistema "DulceVida" con los usuarios y otros componentes de software.

##### 3.1.1 Interfaces de usuario

La interfaz principal del sistema será una aplicación web accesible a través de un navegador.

El diseño de la interfaz será responsive, adaptándose a diferentes tamaños de pantalla como computadores de escritorio, tabletas y teléfonos inteligentes. Esto se logra mediante el uso del framework Bootstrap 5.3.

La paleta de colores principal se basa en tonos cálidos y hogareños como el cacao, crema y dorado, para reflejar la identidad de la marca.

La navegación principal será consistente en todas las páginas, proporcionada por una barra de navegación superior fija que incluye enlaces a "Productos", "Nosotros", "Contacto", "Admin" y el carrito de compras.

El sistema utilizará componentes visuales estándar como formularios, botones, tablas y modales para asegurar una experiencia de usuario intuitiva y familiar.

##### 3.1.2 Interfaces de hardware

El sistema no requiere ninguna interfaz con hardware especializado. Funcionará en cualquier dispositivo de computación estándar (PC, Mac, tableta, smartphone) que cuente con un navegador web moderno y conexión a internet.

### 3.1.3 Interfaces de software

Navegador Web: El sistema interactuará con el usuario final a través de un navegador web.

Debe ser compatible con las últimas versiones de Google Chrome, Mozilla Firefox y Safari.

API del Servidor de Datos: La aplicación se comunicará con un servicio de backend (API) para todas

las operaciones de datos. Este servicio gestiona las solicitudes para obtener, crear y modificar productos, usuarios y órdenes. El endpoint base de esta API es <http://localhost:3000>.

LocalStorage del Navegador: El sistema utilizará la API localStorage del navegador para mantener el estado del carrito de compras del usuario entre sesiones y páginas.

### 3.1.4 Interfaces de comunicación

Protocolo HTTP: La comunicación entre el cliente (navegador web) y el servidor de datos se realizará a través del protocolo HTTP.

Formato de Datos JSON: Todos los datos intercambiados entre el cliente y el servidor de la API se estructurarán en formato JSON (JavaScript Object Notation).

Notificaciones en Pantalla: El sistema comunicará estados de éxito o error al usuario mediante alertas visuales directamente en la interfaz, como mensajes de confirmación tras enviar un formulario o errores de validación.

## 3.2 Requisitos funcionales

A continuación, se definen las acciones fundamentales que debe realizar el software al recibir información, procesarla y producir resultados.

**RF-01:** Registro de nuevo cliente

**Actores:** Cliente.

**Descripción:** El sistema debe permitir que un visitante se registre para crear una cuenta de cliente. El usuario deberá acceder al formulario de registro y completar los campos requeridos (RUN, Nombres, Apellidos, Correo, Región, Comuna, Dirección). El sistema debe realizar validaciones en tiempo real para cada campo, por ejemplo, verificar que el RUN sea válido mediante el algoritmo Módulo 11 y que el correo electrónico pertenezca a un dominio permitido (@duoc.cl, @profesor.duoc.cl, @gmail.com). Al finalizar, el sistema mostrará una confirmación de que el registro fue exitoso.

**RF-02:** Autenticación de usuario

**Actores:** Cliente registrado, Administradora.

**Descripción:** El sistema debe permitir a un usuario registrado iniciar sesión. El usuario ingresará su correo electrónico y una contraseña de al menos 8 caracteres. El sistema verificará las credenciales contra la base de datos. Si son correctas, se le dará acceso a la plataforma y sus datos de sesión se guardarán localmente para mantenerlo autenticado. En caso de credenciales incorrectas o problemas de comunicación, se mostrará un mensaje de error claro.

**RF-03:** Control de acceso por rol

**Actores:** Usuario (Cliente), Administradora.

**Descripción:** El sistema debe restringir el acceso a ciertas áreas según el rol del usuario. Específicamente, el panel de administración (/admin) solo será accesible para usuarios con el rol de "admin". Si un usuario con rol de "cliente" intenta acceder a esta URL, el sistema deberá redirigirlo automáticamente a la página de inicio de sesión.

**RF-04:** Visualización del catálogo de productos

**Actores:** Cliente, Administradora.

**Descripción:** El sistema debe mostrar todos los productos disponibles en una grilla visualmente atractiva en la página "Productos". Para cada producto, se debe presentar su imagen, título, una breve descripción (atributos) y su precio en CLP. La información de los productos se obtendrá consultando la API del sistema.

**RF-05:** Filtrado de productos por categoría

**Actores:** Cliente.

**Descripción:** Para facilitar la navegación, el sistema debe generar y mostrar una serie de botones de filtro basados en las categorías de los productos existentes (ej. Pasteles, Galletas, Dulces). Al hacer clic en un botón de categoría, la grilla de productos se actualizará dinámicamente para mostrar únicamente los artículos que pertenecen a la categoría seleccionada.

**RF-06:** Añadir productos al carrito de compras

**Actores:** Cliente.

**Descripción:** El sistema debe permitir al cliente añadir productos a su carrito de compras. Esta acción se podrá realizar desde la vista principal del catálogo o desde el modal de detalle del producto. El sistema guardará los productos seleccionados, junto con su cantidad, en el localStorage del navegador para que el carrito persista entre diferentes páginas y visitas. El ícono del carrito en la barra de navegación deberá actualizarse para reflejar el número total de artículos.

**RF-07:** Aplicar cupón de descuento

**Actores:** Cliente.

**Descripción:** En la página del carrito, el sistema debe proporcionar un campo para que el cliente ingrese un código de descuento. Al aplicarlo, el sistema consultará la API para validar si el código es activo y válido. Si lo es, calculará el descuento correspondiente (porcentual o monto fijo) y actualizará el total de la compra en el resumen. Si el cupón es inválido, mostrará un mensaje de error.

**RF-08:** Confirmación de pedido (Checkout)

**Actores:** Cliente.

**Descripción:** El sistema debe guiar al cliente para finalizar su compra. El cliente deberá seleccionar un tipo de entrega (delivery o retiro) y completar un formulario con sus datos de contacto y dirección. El sistema calculará automáticamente los costos de envío y mostrará un resumen final del pedido. Al confirmar, el sistema validará que todos los campos obligatorios estén completos y registrará la orden en la base de datos, vaciando el carrito del cliente y mostrando un mensaje de agradecimiento.

**RF-09:** Gestión de productos del catálogo

**Actores:** Administradora.

**Descripción:** Desde el panel de administración, la administradora debe poder realizar operaciones CRUD sobre los productos. Deberá poder crear un nuevo producto a través de un formulario modal, así como editar la información (nombre, precio, stock, etc.) o eliminar un producto existente. Todos los cambios se reflejarán en el estado de la aplicación.

**RF-10:** Gestión de usuarios del sistema

**Actores:** Administradora.

**Descripción:** El sistema debe proveer a la administradora una vista donde pueda ver una tabla con todos los usuarios registrados. Desde esta interfaz, deberá tener la funcionalidad para eliminar usuarios del sistema y también para crear nuevas cuentas de usuario, asignándoles un rol específico.

### 3.3 Requisitos no funcionales

#### 3.3.1 Requisitos de rendimiento

**Tiempo de carga de la página:** El 95% de las veces, las páginas principales del sitio (Inicio, Productos, Carrito) deberán cargarse y ser interactivas en menos de 3 segundos en una conexión a internet de banda ancha estándar.

**Respuesta de la API:** Las interacciones que dependen de la API de datos (como cargar productos, aplicar un cupón o registrar una orden) deberán completarse en menos de 1 segundo desde la acción del usuario hasta la actualización visual en la interfaz.

**Carga de usuarios simultáneos:** El sistema está diseñado para un negocio de pequeña escala. La arquitectura actual debe poder soportar hasta 50 usuarios simultáneos navegando por el sitio sin una degradación perceptible del rendimiento.

### 3.3.2 Seguridad

**Control de Acceso:** El acceso al panel de administración estará estrictamente restringido a los usuarios autenticados que posean el rol de "admin". El sistema verificará el rol del usuario en el momento de intentar acceder a las rutas de administración.

**Política de Contraseñas:** Durante el proceso de inicio de sesión y registro, el sistema exigirá que las contraseñas tengan una longitud mínima de 8 caracteres para fomentar la creación de credenciales más seguras.

**Validación de Entradas:** Todos los datos ingresados por el usuario en los formularios (registro, contacto, checkout) serán validados tanto en el lado del cliente para prevenir la entrada de datos malformados. Por ejemplo, se restringen los dominios de correo electrónico en el registro para controlar quién puede crear una cuenta.

### 3.3.3 Fiabilidad

**Manejo de Errores:** El sistema gestionará los errores de comunicación con la API de forma controlada. Si no se pueden cargar los productos, se mostrará un conjunto de datos de respaldo para que la página no quede vacía, informando al mismo tiempo del fallo en la consola del desarrollador. De manera similar, los fallos en el envío de formularios mostrarán un mensaje de alerta al usuario sin interrumpir el funcionamiento del sitio.

**Consistencia del Carrito:** El estado del carrito de compras se almacena en el localStorage del navegador, lo que asegura que los productos añadidos por el usuario persistan incluso si cierra la pestaña o el navegador, minimizando la pérdida de datos por cierres accidentales.

### 3.3.4 Disponibilidad

**Disponibilidad del Frontend:** La aplicación web (frontend) estará disponible el 100% del tiempo, ya que se compone de archivos estáticos (HTML, CSS, JS) que pueden ser servidos por cualquier servidor web.

**Dependencia del Backend:** La disponibilidad de las funciones dinámicas (cargar productos, registrarse, etc.) depende directamente de que el servicio de la API (json-server) esté en

ejecución. Se espera que este servicio tenga una disponibilidad del 99.9% en un entorno de producción.

### 3.3.5 Mantenibilidad

**Modularidad del Código:** El código JavaScript está organizado en módulos con responsabilidades únicas (cart.js para el carrito, products.js para el catálogo, admin.js para el panel, etc.), lo que facilita la localización de errores y la implementación de futuras mejoras.

**Mantenimiento de Contenido:** La administradora de la pastelería podrá realizar el mantenimiento del contenido principal (productos, precios, usuarios) de forma autónoma a través del panel de administración, sin necesidad de la intervención de un desarrollador.

**Mantenimiento del Software:** Las actualizaciones del código fuente, corrección de errores de programación o cambios estructurales deberán ser realizadas por un desarrollador con conocimientos de HTML, CSS y JavaScript.

### 3.3.6 Portabilidad

**Tecnologías Estándar:** El sistema está construido utilizando tecnologías web universales y no propietarias (HTML5, CSS3, JavaScript ES6), lo que garantiza su funcionamiento en una amplia gama de sistemas operativos (Windows, macOS, Linux, Android, iOS).

**Compatibilidad de Navegadores:** Se garantiza el correcto funcionamiento en las últimas dos versiones de los principales navegadores web: Google Chrome, Mozilla Firefox y Safari.

**Independencia del Servidor:** Al ser una aplicación frontend que se comunica con una API, puede ser desplegada en cualquier servidor web moderno (Apache, Nginx, etc.) sin necesidad de configuraciones complejas dependientes del servidor. El backend (json-server) a su vez es altamente portable y solo requiere Node.js para funcionar.

## ERS - Dulce Vida Web

### 1. Introducción

Sistema e-commerce para pastelería que permite a clientes explorar catálogo, gestionar un carrito, emitir compras (boletas) y a administradores administrar productos, usuarios y monitorear ventas. Arquitectura basada en \*\*React 19\*\* y \*\*Spring Boot 3\*\* con autenticación \*\*JWT\*\*.

### 2. Alcance

Incluye: autenticación, autorización por roles (CLIENTE/ADMINISTRADOR), gestión de productos, flujo de compra y emisión de boleta, historial paginado, panel administrador, contactos básicos y cambio de contraseña. No incluye: pasarela de pago real, facturación electrónica oficial, multi-tenancy, internacionalización.

### 3. Actores

- Cliente: Navega catálogo, administra carrito, realiza checkout, consulta historial.
- Administrador: CRUD productos/usuarios, visualiza compras globales.
- Sistema Externo (opcional futuro): Pasarela de pago (placeholder).

### 4. Requisitos Funcionales

1. Autenticación JWT con refresh y logout seguro.
2. Hashing de contraseñas (BCrypt cost configurable, por defecto 11).
3. Registro y login usuarios (validación email único).
4. CRUD Productos (nombre, descripción, precio, stock, imagen, categoría).
5. CRUD Categorías (nombre único, relación 1:N con productos).
6. Carrito: agregar/quitar productos, ajustar cantidades, persistencia por usuario.
7. Checkout: genera Pedido, descuenta stock, crea Boleta correlativa y Detalles.
8. Historial de compras paginado para cliente (consulta boletas propias).
9. Dashboard administrador: listado paginado de boletas globales.
10. Cambio de contraseña con verificación hash actual.
11. Gestión básica de contactos (almacenamiento mensajes).
12. Paginación en listados (productos, boletas).

### 5. Requisitos No Funcionales

- Seguridad: JWT HS256, secreto >=256 bits Base64, rotación refresh, rate limiting login.
- Rendimiento: Tiempo medio de respuesta API < 1s en operaciones CRUD estándar.
- Escalabilidad: Stateless en capa web (tokens, no sesión servidor para lógica principal).
- Mantenibilidad: Capas separadas (controlador, servicio, repositorio, modelo, dto).
- Calidad: Pruebas unitarias y de integración backend; 6 pruebas frontend con RTL/Vitest.
- Confiabilidad: Manejo de excepciones centralizado (GlobalExceptionHandler).
- Portabilidad: Configuración mediante `application.properties` y variables de entorno.

### 6. Modelo de Datos (Resumen Entidades)

- Usuario(id, nombre, email, passwordHash, rol, activo, creadoEn)
- Producto(id, nombre, descripción, precio, stock, imageUrl, categoria\_id)
- Categoria(id, nombre)
- Pedido(id, usuario\_id, estado, creadoEn)
- Detalle\_Pedido(id, pedido\_id, producto\_id, cantidad, precioUnitario)
- Boleta(idBoleta, numero, fechaEmisión, subtotal, iva, total, pedido\_id, usuario\_id)
- Detalle\_Boleta(idDetalleBoleta, boleta\_id, producto\_id, cantidad, precioUnitario, totalLinea)

- 
- Refresh\_Token(id, usuario\_id, token, revocado, creadoEn, expiracion)
  - Contacto(id, nombre, email, mensaje, creadoEn)

Relaciones Clave:

- Usuario 1:N Pedido
- Pedido 1:N Detalle\_Pedido
- Pedido 1:1 Boleta
- Boleta 1:N Detalle\_Boleta
- Categoria 1:N Producto
- Usuario 1:N Refresh\_Token

## 7. Reglas de Negocio

- IVA fijo 19%: `iva = round(subtotal \* 0.19)`.
- Total boleta = subtotal + iva.
- Numeración boleta correlativa incremental (sin huecos salvo fallos transaccionales rollbacks).
- Stock nunca negativo; checkout falla si algún producto no tiene cantidad suficiente.
- Refresh token rotado en cada `/auth/refresh`; anterior marcado como revocado.
- Contraseña mínima 8 caracteres (validación en registro/cambio password).
- Intentos de login: máximo 5 cada 5 minutos por IP/email (rate limiting simple).

## 8. Paginación

- Parámetros estándar `page` (0-based) y `size`.
- Boletas cliente: `GET /api/boletas/mias?page=0&size=5`.
- Boletas admin: `GET /api/boletas/admin?page=0&size=10`.
- Respuesta conforme a `Page<>` de Spring (content, totalPages, totalElements).

## 9. Suposiciones

- Un pedido se asocia a exactamente una boleta.
- No se manejan devoluciones / notas de crédito en esta etapa.
- El carrito se reconstruye a partir de estado en servidor al autenticarse (persistencia ligada a usuario, no anónimo prolongado).

## 10. Riesgos

- Exposición de secreto JWT si no se configura variable entorno segura.
- Falta de índice en columnas de búsqueda podría degradar rendimiento con crecimiento de datos.
- Uso de correlativo simple podría requerir separación futura para entornos multi- nodo.

## 11. Validación

Pruebas JUnit cubren generación de boleta, ajuste stock, hash de contraseña y autenticación. Pruebas frontend verifican flujos críticos de UI y protección de rutas.

## 12. Trazabilidad (Mapa rápido RF -> Artefactos)

RF	Artefactos
-----	-----
1	SecurityConfig, JwtAuthenticationFilter, AuthControlador
4-7	CarritoServicio, Pedido repositorio, BoletaServicio/Controlador
8-9	BoletaControlador endpoints `/mias`, `/admin`
10	UsuarioServicio (cambio contraseña)
12	Controladores con parámetros `page`, `size`  ---

