

Especificación de Requisitos de Software

Proyecto: SPAManager

Revisión: [1.0]

[12/09/2025]

Contenido

FICHA DEL DOCUMENTO	3
1. INTRODUCCIÓN	4
1.1. PROPÓSITO	4
1.2. ÁMBITO DEL SISTEMA	4
1.3. DEFINICIONES, ACRÓNIMOS Y ABREVIATURAS	4
1.4. REFERENCIAS	5
1.5. VISIÓN GENERAL DEL DOCUMENTO	5
2. DESCRIPCIÓN GENERAL	6
2.1. PERSPECTIVA DEL PRODUCTO	6
2.2. FUNCIONES DEL PRODUCTO	7
2.3. CARACTERÍSTICAS DE LOS USUARIOS	7
2.4. RESTRICCIONES	8
2.5. SUPOSICIONES Y DEPENDENCIAS	9
2.6. REQUISITOS FUTUROS	10
3. REQUISITOS ESPECÍFICOS	11
3.1 REQUISITOS COMUNES DE LAS INTERFACES	11
3.1.1 <i>Interfaces de usuario</i>	11
3.1.2 <i>Interfaces de hardware</i>	14
3.1.3 <i>Interfaces de software</i>	15
3.1.4 <i>Interfaces de comunicación</i>	16
3.2 REQUISITOS FUNCIONALES	16
3.3 REQUISITOS NO FUNCIONALES	31
3.3.1 <i>Requisitos de rendimiento</i>	31
3.3.2 <i>Seguridad</i>	31
3.3.3 <i>Fiabilidad</i>	32
3.3.4 <i>Disponibilidad</i>	32
3.3.5 <i>Mantenibilidad</i>	33
3.3.6 <i>Portabilidad</i>	33
3.4 OTROS REQUISITOS	33

Ficha del documento

Fecha	Revisión	Autor	Modificación
05/09/2025	1.0	P. Alvarez	Inicio informe
06/09/2025	1.1	P. Alvarez	Avance
11/09/2025	1.2	P. Alvarez	Requerimientos

Documento validado por las partes en fecha:

Por el cliente

Por la empresa suministradora

[Firma]

[Firma]

Sr./Sra.

Sr./Sra.

1. Introducción

1.1. Propósito

El propósito de este documento es especificar de manera clara y estructurada los requisitos funcionales y no funcionales del sistema de reservas y gestión de comisiones para el SPA del Bosque.

Este ERS servirá como contrato entre los clientes (administración del SPA, terapeutas y usuarios finales) y el equipo de desarrollo, estableciendo qué funcionalidades serán implementadas, así como las restricciones y condiciones bajo las cuales debe operar el sistema.

1.2. Ámbito del Sistema

El sistema por desarrollar será una plataforma web que permitirá:

- A los clientes: explorar servicios, registrarse, iniciar sesión, agendar reservas, comprar Gift Cards y realizar pagos.
- A los terapeutas: visualizar sus servicios agendados, confirmar atenciones, registrar información relevante de los clientes y poder llevar un registro de sus comisiones.
- A la jefatura y administración: administrar servicios, gestionar terapeutas, calcular comisiones automáticamente y generar reportes exportables.
- A contabilidad: recibir información consolidada para el cálculo de remuneraciones (incluyendo locomoción y descuentos legales).

El sistema debe ser escalable, de modo que en el futuro pueda integrarse con otros sistemas externos como Novohit (sistema hotelero) y pasarelas de pago.

1.3. Definiciones, Acrónimos y Abreviaturas

- SPA: Centro de terapias y masajes.
- ERS: Especificación de Requisitos de Software.
- UI: Interfaz de usuario.
- CRUD: Create, Read, Update, Delete (operaciones básicas de gestión de datos).
- Giftcard: Tarjeta de prepago para usar en servicios del SPA.
- Novohit: Sistema de gestión hotelera que se considera para una futura integración.

1.4. Referencias

- Planilla de Requerimientos del proyecto SPA del Bosque (versión validada).
- Entrevistas realizadas a terapeutas y administración del SPA (fuente directa de requisitos).

1.5. Visión General del Documento

Este documento de Especificación de Requisitos de Software (ERS) está organizado en tres grandes áreas:

- Introducción (Sección 1): entrega el propósito, alcance, definiciones, referencias y esta visión general.
- Descripción General (Sección 2): describe el contexto en el que se desarrollará el sistema, sus funciones principales, los perfiles de usuario, restricciones, dependencias y posibles extensiones futuras.
- Requisitos Específicos (Sección 3): detalla de manera estructurada los requerimientos del sistema, divididos en:
 - Requisitos comunes de las interfaces (usuario, hardware, software y comunicación).
 - Requisitos funcionales (RF) del sistema: funcionalidades principales que debe cumplir la plataforma.
 - Requisitos no funcionales (RNF): desempeño, seguridad, fiabilidad, disponibilidad, mantenibilidad y portabilidad.
 - Otros requisitos adicionales relevantes.

Finalmente, el documento incluye anexos que contienen la planilla de requerimientos y formularios de caso de uso, los cuales complementan la especificación y sirven como base para el diseño y la validación del sistema.

2. Descripción General

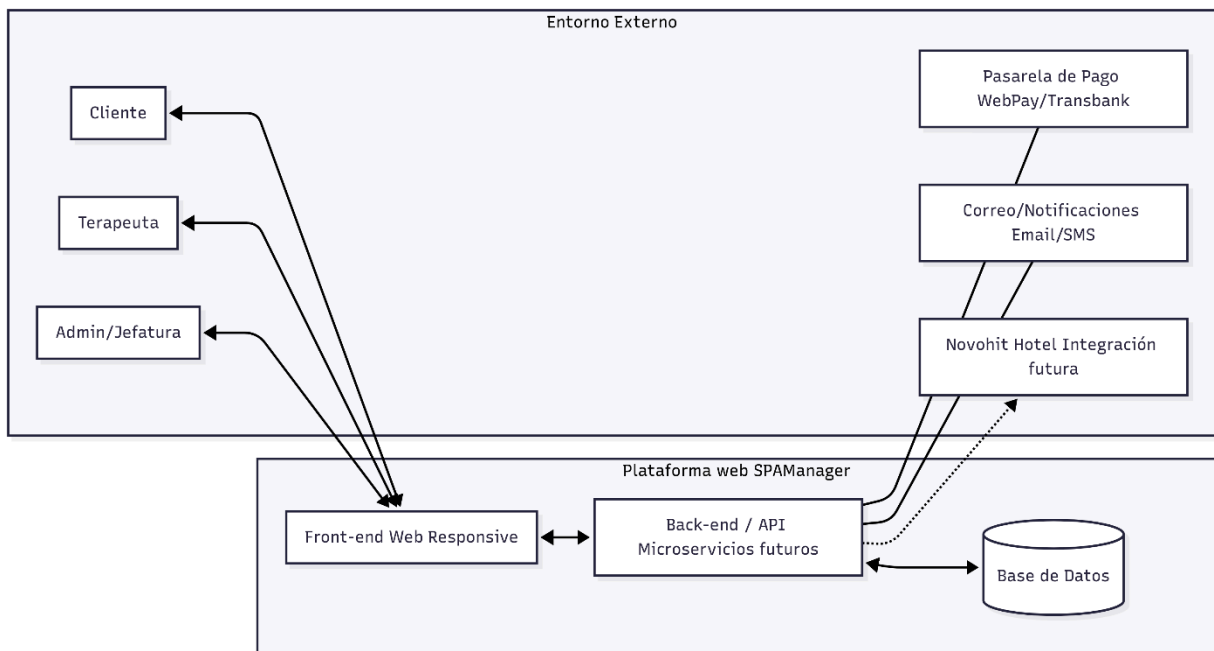
2.1. Perspectiva del Producto

El sistema propuesto corresponde a una plataforma web para la gestión integral de reservas, clientes y servicios del SPA del Bosque, el sistema en adelante llamado “SPAManager”. Actualmente, gran parte de la administración de agendamientos, pagos y control de comisiones de las terapeutas se realiza de manera manual, con registros en hojas de cálculo y comunicación informal con los clientes.

El producto se concibe como un sistema centralizado que digitaliza y unifica estas operaciones, proporcionando:

- Front-end para clientes: visualización de servicios, reservas en línea, compra de Gift Cards y acceso a promociones.
- Front-end para terapeutas: registro de atenciones realizadas y consultas de comisiones acumuladas.
- Panel de administración/jefatura: gestión de usuarios, servicios, control de reservas, autorizaciones de correcciones de comisiones y generación de reportes (PDF/Excel) para el área contable.

El sistema se integrará en fases: en primera instancia como un portal web responsive, y en el futuro podrá conectarse con plataformas externas de pago electrónico y, eventualmente, evolucionar a una aplicación de escritorio o móvil.



2.2. Funciones del Producto

El sistema SPAManager contará con un conjunto de funciones principales, agrupadas según el tipo de usuario:

1. Funciones para clientes

- a. Navegar por catálogos de servicios (masajes, tratamientos, programas).
- b. Realizar reservas en línea según disponibilidad.
- c. Comprar y/o canjear giftcards.
- d. Gestionar cuenta de usuario (registro, inicio de sesión, perfil).
- e. Visualizar historial de reservas y pagos.

2. Funciones para terapeutas

- a. Acceder al perfil de terapeuta (sesión personal).
- b. Registrar servicios realizados.
- c. Consultar comisiones acumuladas por día, semana o mes.
- d. Visualizar agenda de reservas asignadas.

3. Funciones para Administración/Jefatura

- a. Gestión de servicios (crear, modificar, desactivar).
- b. Gestión de usuarios (clientes, terapeutas, administradores).
- c. Asignación de terapeutas a las reservas.
- d. Generación de reportes de reservas y comisiones en PDF/Excel.
- e. Control de pagos pendientes y estados de giftcards.

4. Funciones transversales

- a. Sistema de autenticación y roles (cliente, terapeuta, administración).
- b. Envío de notificaciones (confirmación de reservas, recordatorios, cambios).
- c. Seguridad y respaldo de la información.

2.3. Características de los Usuarios

El sistema está orientado a distintos perfiles de usuario, cada uno con diferentes niveles de conocimiento y necesidades específicas. A continuación, se describen sus características generales:

1. Usuario Cliente

- a. Perfil: Persona que accede a la plataforma para reservar y/o comprar servicios del SPA.
- b. Nivel educacional: Diverso; se asume al menos educación media.
- c. Experiencia técnica: Uso básico de PC o dispositivos móviles, manejo de internet y correo electrónico.
- d. Necesidades: Acceder fácilmente al catálogo, reservar servicios, pagar o gestionar su historial de atenciones.

2. Usuario Terapeuta

- a. Perfil: Profesional del SPA que ejecuta los servicios (masajes, tratamientos, programas).
- b. Nivel educacional: Formación técnica o profesional en terapias de bienestar, corporal y/o facial.

- c. Experiencia técnica: Conocimiento básico en uso de PC y/o dispositivos móviles, especialmente para consultar su agenda y registrar servicios realizados.
- d. Necesidades: Registrar terapias realizadas, visualizar sus reservas asignadas y consultar comisiones acumuladas.

3. Usuario Administrador / Jefatura

- a. Perfil: Encargado de la gestión operativa y administrativa del SPA.
- b. Nivel educacional: Educación técnica/profesional en administración o áreas afines.
- c. Experiencia técnica: Uso de PC nivel intermedio, con manejo de Excel a nivel medio y sistemas de gestión básicos.
- d. Necesidades: Administrar usuarios y servicios, asignar terapeutas a reservas, generar reportes de comisiones y controlar pagos.

2.4. Restricciones

El sistema estará sujeto a las siguientes limitaciones y condiciones:

Categoría	Restricción aplicada
Políticas de la empresa	Cumplimiento con la Ley de Protección de Datos Personales en Chile y políticas internas de confidencialidad.
Limitaciones del hardware	Funcionamiento en servidores cloud de capacidad media (AWS/Azure/GCP). Acceso desde navegadores modernos en dispositivos móviles y de escritorio.
Interfaces con otras aplicaciones	Integración futura con pasarelas de pago (WebPay, MercadoPago) y facturación electrónica. Deben cumplirse formatos y protocolos definidos por terceros.
Operaciones paralelas	Manejo de múltiples reservas concurrentes garantizando consistencia de datos mediante control transaccional en la base de datos (MySQL).
Funciones de auditoría	Registro de accesos, transacciones y modificaciones en bitácora segura para trazabilidad y control de incidencias.
Funciones de control	Control de acceso basado en roles: cliente, administrador, terapeuta. Operaciones críticas limitadas a administradores.
Lenguajes de programación	Frontend: HTML5, CSS3, Bootstrap, React JavaScript. Backend: Spring Boot. Base de datos: PostgreSQL/MySQL.

Protocolos de comunicación	Todas las comunicaciones deberán realizarse bajo HTTPS con certificados SSL. Autenticación mediante JWT u OAuth2.
Requisitos de habilidad	Clientes: uso básico de aplicaciones web/móviles. Administradores: gestión de reservas y reportes. Terapeutas: consulta de agenda en interfaz simplificada.
Criticidad de la aplicación	El sistema es crítico para la operación del SPA: caídas impactan reservas, pagos y atención al cliente.
Consideraciones de seguridad	Cifrado de contraseñas con bcrypt, control de sesiones activas, políticas de contraseñas seguras y cumplimiento de buenas prácticas OWASP (prevención de SQL Injection, XSS, CSRF).

2.5. Suposiciones y Dependencias

Tipo	Descripción
Suposición	Los clientes del SPA cuentan con acceso estable a internet desde dispositivos móviles o computadoras personales.
Suposición	Los usuarios utilizarán navegadores modernos y actualizados (Chrome, Firefox, Edge, Safari) que soporten HTML5, CSS3 y JavaScript ES6.
Suposición	El personal administrativo y de caja posee conocimientos básicos de ofimática y manejo de sistemas web.
Suposición	Las imágenes y contenidos (textos, promociones, servicios) serán provistos por el área de marketing del SPA y actualizados periódicamente.
Dependencia	La aplicación depende de servicios de terceros como la pasarela de pago (WebPay, MercadoPago u otros) para procesar transacciones en línea.
Dependencia	El sistema requiere servicios de hosting/cloud (AWS/Azure/GCP) para la base de datos y backend.
Dependencia	Para envío de notificaciones por correo electrónico, el sistema dependerá de un proveedor externo (ej. SendGrid, Amazon SES, Gmail API).
Dependencia	El sistema de reservas depende de la correcta sincronización con el calendario interno y la disponibilidad de terapeutas.

Dependencia	El correcto desempeño de la aplicación requiere la implementación de certificados SSL para la comunicación segura bajo HTTPS.
Dependencia	La integración con facturación electrónica dependerá de las normativas del SII (Servicio de Impuestos Internos, Chile).

2.6. Requisitos Futuros

El sistema actual se desarrollará con una arquitectura web modular, permitiendo futuras mejoras e integraciones. A continuación, se describen los posibles requisitos futuros que podrían ser implementados:

1. Aplicación móvil nativa

- a. Desarrollo de aplicaciones móviles en Android e iOS, conectadas al mismo backend mediante APIs REST/GraphQL.

2. Integración con sistemas externos

- a. Conexión con software de contabilidad y facturación (ejemplo: ERP o módulos del SII).
- b. Sincronización con Google Calendar o equivalentes para reservas automáticas.

3. Expansión a múltiples sucursales

- a. Soporte para la gestión de varias sucursales del SPA con administración centralizada.

4. Módulo de fidelización de clientes

- a. Implementación de giftcards digitales, puntos de fidelidad y membresías VIP.

5. Inteligencia artificial y Analítica avanzada

6. Soporte multilingüe

- a. Versión de la plataforma en inglés y portugués para clientes internacionales.

7. Automatización de marketing

- a. Envío de promociones personalizadas por correo, SMS o WhatsApp.

8. Monitoreo en tiempo real

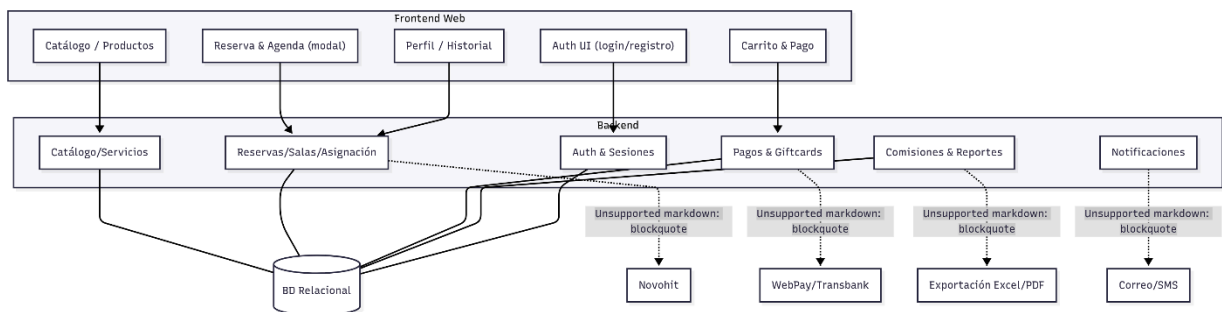
- a. Dashboard administrativo con indicadores de ocupación, ventas y satisfacción del cliente.

3. Requisitos Específicos

En esta sección se detallan los requisitos funcionales y no funcionales del sistema, a un nivel de detalle suficiente para permitir su diseño, implementación y pruebas.

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

El sistema contará con una serie de interfaces gráficas accesibles a través de un navegador web responsivo (desktop y mobile). Las interfaces están diseñadas bajo principios de simplicidad, consistencia visual y accesibilidad, utilizando componentes Bootstrap 5 para garantizar compatibilidad en distintos dispositivos.

A continuación, se describen las principales vistas del sistema:

Vista Home

- Propósito: Pantalla inicial de acceso al portal.
- Elementos clave:
 - Banner principal con imagen y bienvenida.
 - Navegación a categorías destacadas (masajes, corporales, circuitos, programas, etc.).
 - Sección de servicios destacados con opción de “Ver Detalle” y “Agregar al Carrito”.
 - Acceso rápido a login/registro.

Vista Registro

- Propósito: Permitir al usuario crear una cuenta.
- Elementos clave:
 - Formulario con campos: nombre, apellido, correo, contraseña, confirmación de contraseña, teléfono, región y comuna.
 - Validaciones en tiempo real para formato de correo, longitud de contraseña y número de teléfono.
 - Botón Registrarse que almacena el usuario en la base de datos.
 - Link a la vista Login para usuarios existentes.

Vista Productos

- Propósito: Listar los servicios disponibles organizados por categoría.
- Elementos clave:
 - Índice de categorías (masajes, corporales, circuitos, programas individuales/pareja, escapada de amigas).
 - Cards dinámicas renderizadas desde base de datos/JS con imagen, título, precio, botón Ver detalle, Agendar y botón Agregar al carrito.
 - Navegación por anclas a cada categoría.
 - Botón “Volver arriba” ↑.

Vista Detalle de Producto

- Propósito: Mostrar información específica de un servicio.
- Elementos clave:
 - Nombre del servicio, categoría, duración, precio y descripción extendida.
 - Botones Agregar al Carrito, Volver a servicios y Agendar (que abre modal de reserva con fecha/horario/terapeuta).
 - Breadcrumb de navegación para volver a la categoría.

Vista Carrito

- Propósito: Centralizar la gestión de reservas/servicios seleccionados.
- Elementos clave:
 - Tabla con servicios agregados: nombre, precio unitario, cantidad, subtotal y total.
 - Botones: Vaciar carrito, Seguir agregando servicios, Proceder a reservar/pagar.
 - Estado vacío del carrito con mensaje de invitación a explorar servicios.
 - Validación: si el usuario no está logueado, se redirige a Login.

Vista Blogs

- Propósito: Mostrar contenido de bienestar, consejos y noticias del SPA.
- Elementos clave:
 - Listado de artículos con título, fecha y vista previa.
 - Opción de leer artículo completo.
 - Solo el administrador puede crear/editar entradas.

Vista Nosotros

- Propósito: Presentar información institucional.
- Elementos clave:
 - Misión, visión y valores del SPA.
 - Equipo de terapeutas destacados con fotos y descripción.
 - Información de contacto resumida.

Vista Contacto

- Propósito: Permitir la comunicación directa con el SPA.
- Elementos clave:
 - Formulario de contacto (nombre, correo, mensaje).
 - Validaciones de campos obligatorios.
 - Mapa de ubicación y horarios de atención.

Vista Administración (Backoffice)

- Propósito: Gestión interna de productos, usuarios y contenidos.
- Perfiles autorizados: Administrador.
- Elementos clave:
 - Gestión de Servicios: CRUD completo de servicios (crear, editar, eliminar, listar).
 - Gestión de Usuarios: Crear usuarios, asignar roles (cliente, terapeuta, admin), deshabilitar cuentas.
 - Gestión de Blogs: Crear y modificar entradas de contenido.
 - Dashboard con métricas básicas (reservas, servicios más vendidos).

Ver Anexo 3 – Wireframes de Interfaces

Estándares y lineamientos

- La UI debe ser responsive (mobile-first). Breakpoints de referencia (Bootstrap):
 - xs <576px, sm ≥576px, md ≥768px, lg ≥992px, xl ≥1200px, xxl ≥1400px.
- Accesibilidad (WCAG 2.1 AA):
 - Contraste mínimo 4.5:1 para texto.
 - Enfoque visible en elementos interactivos.
 - Labels, aria-* y mensajes de error claros en formularios.
 - Navegación por teclado sin bloqueos.
- Idiomas: interfaz en es-CL, preparada para i18n (textos centralizados).
- Consistencia visual: diseño basado en Bootstrap 5 + paleta corporativa (variables CSS).
 - 0CA695 → verde hoja (principal)
 - 99D2CB → verde agua (suave)
 - 9C9C9C → gris medio (texto secundario)
 - B6B6B6 → gris claro (borde)
 - F0F6F4 → verde muy claro (fondo sutil)
- Patrones de interacción:
 - Listado de servicios con cards + modal de agendamiento.
 - Carrito persistente, contador visible en navbar.
 - Flujo de login/registro con feedback (toasts/alerts).
 - Panel Admin/Jefatura con tablas filtrables y exportación (CSV/XLSX).

- Validación de formularios:
 - Validación en cliente (HTML5 + JS) y en servidor (Spring Validation).
 - Mensajes de error específicos (campo, regla rota, ejemplo válido).
- Estados vacíos y errores:
 - Listas vacías con CTA claro (p. ej., "Agregar servicio").
 - Errores recuperables con explicación y opción de reintentar.
- Rendimiento percibido:
 - Carga diferida de imágenes (loading="lazy").
 - Dividir bundles (code splitting) en React.
 - Skeletons/placers en pantallas con datos remotos.

Vistas mínimas

- Públicas: Home, Catálogo/Servicios, Detalle de servicio, Blogs (listado + detalle), Contacto, Nosotros.
- Autenticadas (Cliente): Carrito, Reserva (modal + confirmación), Historial, Perfil.
- Administración/Jefatura: CRUD Servicios, Usuarios/Roles, Agenda (asignaciones), Reportes/Comisiones.

Criterios de aceptación

- La UI se visualiza sin desbordes ni superposiciones en resoluciones $\geq 360 \times 640$.
- Todos los formularios muestran errores inline y no permiten envío si hay reglas incumplidas.
- Lectores de pantalla anuncian títulos, enlaces, botones y toasts.

3.1.2 Interfaces de hardware

Clientes (usuarios finales)

- Dispositivos: smartphones Android/iOS (2 últimos SO), tabletas y PC.
- Navegadores soportados (últimas 2 versiones): Chrome, Edge, Firefox, Safari (iOS).
- Conectividad: $\geq 3G$ estable (recomendado ≥ 10 Mbps para mejor experiencia).

Servidor (referencial en cloud)

- Contenedor Java 17 (Spring Boot) + Node (para build del frontend, si aplica).
- vCPU: 2–4 | RAM: 4–8 GB | Disco SSD: 20–50 GB.
- Base de datos MySQL administrada (db.t3.small o equivalente) con backups diarios.
- Balanceador/Reverse proxy (Nginx/ALB) con TLS.

Criterios de aceptación

- Con 100 usuarios concurrentes navegando catálogo, el tiempo de respuesta P95 < 2 s.
- Backups automáticos diarios con retención ≥ 7 días.

3.1.3 Interfaces de software

Frontend

- React, Bootstrap 5, Fetch/axios para consumo de API.
- Manejo de estado (Context/Redux) para auth y carrito.
- Ruteo de SPA (React Router).

Backend

- Spring Boot (Maven), Java 17.
- Validación: Bean Validation (Jakarta Validation).
- Seguridad: Spring Security + JWT.
- Persistencia: Spring Data JPA + MySQL
- Documentación API: OpenAPI/Swagger.

API REST (formato)

- JSON en request/response. Zona horaria América/Santiago en fechas/horas.
- Convenciones de rutas (ejemplos):
 - o GET /api/v1/servicios?categoria=masajes
 - o GET /api/v1/servicios/{sku}
 - o POST /api/v1/reservas (Auth requerida)
 - o GET /api/v1/reservas?desde=2025-09-01&hasta=2025-09-30
 - o POST /api/v1/auth/login · POST /api/v1/auth/registro · POST /api/v1/auth/refresh
 - o GET /api/v1/reportes/comisiones?mes=2025-09 (rol: jefatura/admin)
- Códigos HTTP:
 - o 200/201 OK/Created · 204 No Content · 400 Bad Request · 401 Unauthorized · 403 Forbidden · 404 Not Found · 409 Conflict · 422 Unprocessable Entity · 500 Server Error.
- Contrato de error (estándar):

```
{
  "timestamp": "2025-09-12T12:34:56Z",
  "status": 422,
  "code": "VALIDATION_ERROR",
  "message": "Datos inválidos",
  "details": [
    { "field": "email", "error": "Formato inválido" },
    { "field": "password", "error": "Mínimo 8 caracteres" }
  ],
  "path": "/api/v1/auth/registro",
  "traceId": "c1a2b3..."
}
```

- Autenticación/Autorización
 - o Login: POST /auth/login → { accessToken, refreshToken, user }
 - o En cada request protegida: Authorization: Bearer <accessToken>
 - o Renovación de token: POST /auth/refresh
 - o **Roles:** CLIENTE, TERAPEUTA, ADMIN (y JEFATURA si lo separas).

- Integraciones (futuras)
 - Pasarela de pago (WebPay/MercadoPago): endpoints de callback/return.
 - Correo transaccional (SES/SendGrid): servicio SMTP/API.
 - ERP/Hotel (Novohit): sincronización de reservas/facturas (batch o webhooks).

Criterios de aceptación

- La especificación Swagger expone al menos: auth, servicios, reservas, usuarios, reportes.
- Todos los errores devuelven JSON con code y message; nunca HTML.

3.1.4 Interfaces de comunicación

Protocolos y seguridad

- Todo el tráfico HTTPS (TLS 1.2+). Redirección HTTP→HTTPS.
- CORS configurado (orígenes permitidos del frontend).
- JWT en cabecera Authorization. Expiración de accessToken (ej. 15–30 min) y refreshToken (ej. 7–30 días).
- Rate limiting en endpoints sensibles (auth/pago) para mitigar abuso.
- Headers de seguridad: HSTS, X-Content-Type-Options, X-Frame-Options (DENY), Referrer-Policy.

Puertos y topología

- Puerto público 443 (HTTPS). Backend detrás de reverse proxy/balanceador.
- Acceso a BD restringido a red privada/VPC (no expuesta a Internet).

Observabilidad

- Logs estructurados (JSON) con traceId y userId.
- Métricas: salud (/actuador/health), latencias, uso de CPU/memoria, errores por endpoint.
- Alertas: indisponibilidad, latencias > umbral, tasa de errores.

Criterios de aceptación

- Rechazar cualquier petición sin Authorization válida en endpoints protegidos (401).
- CORS solo permite los orígenes configurados; preflight responde en < 200 ms.
- Logs no almacenan datos sensibles (no emails completos en trazas, no contraseñas).

3.2 Requisitos funcionales

3.2.1 Requisito funcional 1 (RF01) – Registro e inicio de sesión

Actores: Cliente, Administrador, Terapeuta.

Descripción: El usuario debe poder registrarse e iniciar sesión para acceder a funcionalidades protegidas (reservar, ver historial, administrar).

Entradas y validaciones

- email (obligatorio, formato válido, único en el sistema).
- password (obligatorio, mínimo 8 caracteres, políticas de complejidad).
- nombre, apellido, teléfono (obligatorios).
- Validación de credenciales en login (email+password).

Secuencia

1. El usuario completa el formulario de registro y envía.
2. El sistema valida reglas y verifica unicidad de email.
3. El sistema crea la cuenta y retorna confirmación.
4. En login, el sistema valida credenciales y emite token de sesión (JWT) con rol.
5. El sistema redirige a la pantalla correspondiente (Home/Anterior).

Salidas

- Mensajes de éxito/error por campo.
- Token de sesión (cliente/terapeuta/admin) y datos básicos del usuario.
- Redirección a la ruta objetivo.

Excepciones y recuperación

- Email duplicado → 409 Conflict.
- Credenciales inválidas → 401 Unauthorized.
- Formato inválido → 422 Unprocessable Entity con detalles de campos.

Relación entradas – salidas

- Entradas válidas → alta de usuario y sesión creada.
- Entradas inválidas → no se crea usuario/sesión; se devuelven errores por campo.

Requisitos lógicos de datos (BD)

- Tabla usuarios(id, email*, password_hash, nombre, apellido, telefono, rol, creado_en).
- Índice único por email.

Criterios de aceptación

- No permite registro con email repetido.
- Login con credenciales correctas emite JWT con rol correcto.

3.2.2 Requisito funcional 2 (RF02) – Gestión de catálogo de servicios

Actores: Administrador/Jefatura (gestión); Cliente (consulta).

Descripción: Administrar y consultar la oferta de servicios (masajes, corporales, programas, circuitos).

Entradas y validaciones

- sku (obligatorio, único), nombre, categoría, precio, duracionMin, descripcion, estado.
- Validaciones: precio > 0; duración > 0; categoría ∈ {masajes, corporales, circuitos, individuales, parejas, escapada-amigas}.

Secuencia (Admin)

1. Crear/editar/eliminar servicio desde panel.
2. Validar reglas y persistir cambios.

Salidas

- Listados paginados/filtrados.
- Confirmación de alta/edición/baja.

Excepciones

- sku duplicado → 409 Conflict.
- Eliminación de servicio con reservas futuras → bloquear o solicitar confirmación (regla de negocio).

Relación entradas – salidas

- Alta/edición válida → registro actualizado visible en catálogo.
- Eliminación → no visible para clientes.

Requisitos lógicos de datos (BD)

- Tabla servicios(sku*, nombre, categoria, precio, duracion_min, descripcion, estado, creado_en, actualizado_en).

Criterios de aceptación

- Un cliente puede filtrar y ver detalles en < 2s (P95)

3.2.3 Requisito funcional 3 (RF03) – Carrito y checkout (pre-reserva / compra)

Actores: Cliente.

Descripción: El cliente puede agregar servicios a un carrito y proceder al flujo de pago/reserva.

Entradas y validaciones

- sku, cantidad (por defecto 1), modo ∈ {reserva, giftcard}.
- Validar existencias/reglas de servicio (ej. no giftcard para ciertos servicios si así se define).

Secuencia (Admin)

1. Agregar al carrito desde catálogo/detalle.
2. Ver carrito: modificar cantidades o eliminar ítems.
3. Iniciar checkout → si no está autenticado, login/registro.
4. Confirmar modo (reserva o giftcard) y datos de contacto.

Salidas

- Resumen de ítems, subtotal, total.
- Mensajes de confirmación en cada paso.

Excepciones

- Ítems inexistentes o desactivados → remover del carrito.
- Total = 0 → bloquear checkout.

Relación entradas – salidas

- Ítems válidos → carrito actualizable y total recalculado.
- Entrada inválida → mensaje y no aplica cambio.

Requisitos lógicos de datos (BD)

- Tabla carritos(id, usuario_id, estado) + carrito_items(carrito_id, sku, qty, precio_unitario).

Criterios de aceptación

- Actualizar cantidad refleja subtotal y total inmediatamente.

3.2.4 Requisito funcional 4 (RF04) – Agendamiento de servicios (reserva)

Actores: Cliente (solicita), Admin/Jefatura (gestiona), Terapeuta (consulta).

Descripción: Permite agendar fecha/hora para un servicio, considerando salas disponibles y duración.

Entradas y validaciones

- usuario_id, sku, fecha, hora, sala_id (o “asignación posterior”), terapeuta_id (opcional en la solicitud).
- Validar: disponibilidad de sala/slot, duración compatible, límites de anticipación y cancelación.

Secuencia

1. Cliente abre modal de agenda desde servicio.

2. Selecciona fecha/hora; sistema muestra disponibilidad real (bloques libres).
3. Confirma reserva → se crea en estado “pendiente de pago” o “confirmada” según política.
4. Admin/Jefatura puede reasignar sala/terapeuta.

Salidas

- Código de reserva, estado (pendiente/confirmada/cancelada), detalle (fecha, hora, sala, terapeuta).

Excepciones

- Conflicto de disponibilidad (condición de carrera) → informar y solicitar nuevo slot.
- Vencimiento de ventana de pago para reservas pendientes → cambio automático a “cancelada”.

Relación entradas – salidas

- Entradas válidas → reserva creada y visible en agendas.
- Conflicto → no se crea y se indica alternativa.

Requisitos lógicos de datos (BD)

- reservas(id, usuario_id, sku, fecha_hora_inicio, fecha_hora_fin, sala_id, terapeuta_id, estado, creado_en).
- Índices por sala_id + fecha_hora_inicio y terapeuta_id + fecha_hora_inicio.

Criterios de aceptación

- No es posible reservar dos servicios que se solapan en la misma sala/terapeuta. Debe haber un lapso de 10 minutos entre servicios.

3.2.5 Requisito funcional 5 (RF05) – Pagos y giftcards

Actores: Cliente, Admin/Jefatura.

Descripción: Registrar pagos de reservas y emisión/canje de giftcards.

Entradas y validaciones

- monto, moneda (=CLP), medio_pago (WebPay/efectivo), id_reserva o giftcard_id.
- Validar: totales coinciden con carrito, giftcard vigente/saldo suficiente.

Secuencia

1. Cliente inicia pago; sistema redirige a pasarela y espera callback.
2. Si aprobado, se marca la reserva como pagada.

3. Emisión de giftcard (si aplica): genera código único con saldo y vencimiento.
4. Canje de giftcard: descuenta saldo y cierra transacción.

Salidas

- Comprobante de pago/boleto (PDF opcional), estado de reserva/giftcard.

Excepciones

- Pago rechazado o timeout → reserva sigue pendiente; se ofrece reintento.
- Giftcard sin saldo/vencida → 422 Unprocessable Entity.

Relación entradas – salidas

- Pago aprobado → saldo/estado actualizados; se registra transacción.
- Pago fallido → no cambia estado; log de intento.

Requisitos lógicos de datos (BD)

- pagos(id, reserva_id?, giftcard_id?, monto, estado, medio, fecha_pago, referencia_tx).
- giftcards(id, codigo*, saldo, vence_en, estado, usuario_id?).

Criterios de aceptación

- Doble click en “pagar” no duplica transacciones (idempotencia).

3.2.6 Requisito funcional 6 (RF06) – Registro de atención y cálculo de comisiones

Actores: Terapeuta (registra), Admin/Jefatura (controla), Contabilidad (consulta).

Descripción: Al finalizar una terapia, se registra la atención y se calcula comisión.

Entradas y validaciones

- reserva_id, terapeuta_id, realizada (sí/no), observaciones (opcional).
- Validar: que la reserva exista, esté confirmada/pagada, y la fecha/hora ya haya ocurrido.

Secuencia

1. Terapeuta/recepción marca la reserva como “realizada”.
2. El sistema calcula comisión según reglas (por categoría/servicio/rol).
3. Suma a acumulado diario/semanal/mensual del terapeuta.

Salidas

- Estado de atención, monto de comisión, acumulados.

Excepciones

- Marca de atención antes del horario → bloquear con mensaje.
- Reverso de atención (error) → requiere rol Admin/Jefatura y justificación.

Relación entradas – salidas

- Atención realizada → comisión calculada y acumulada.
- Atención no realizada (no show) → sin comisión; se registran métricas.

Requisitos lógicos de datos (BD)

- atenciones(id, reserva_id*, terapeuta_id, estado, timestamp, observaciones).
- comisiones(id, terapeuta_id, periodo, monto, detalle_json).

Criterios de aceptación

- Los totales de comisión por periodo cuadran con el detalle de atenciones.

3.2.7 Requisito funcional 7 (RF07) – Administración de usuarios y roles

Actores: Administrador/Jefatura.

Descripción: Alta, baja, modificación y asignación de roles (cliente, terapeuta, admin).

Entradas y validaciones

- email único, rol válido, datos personales mínimos.

Secuencia

1. Listar usuarios con filtros.
2. Crear/editar usuario; asignar roles/permisos.
3. Desactivar usuario (no borrar histórico).

Salidas

- Confirmaciones y listados actualizados.

Excepciones

- Cambio de rol inválido (conflicto de negocio) → rechazo con motivo.

Relación entradas – salidas

- Cambios válidos → usuario actualizado sin perder historial.

Requisitos lógicos de datos (BD)

- usuarios_roles(usuario_id, rol).
- Auditoría de cambios (quién, cuándo, qué).

Criterios de aceptación

- Un terapeuta desactivado no puede ser asignado a nuevas reservas.

3.2.8 Requisito funcional 8 (RF08) – Reportes y exportación (comisiones, reservas)

Actores: Jefatura, Contabilidad, Administrador.

Descripción: Generar reportes por rango de fechas y exportar a Excel/PDF.

Entradas y validaciones

- desde, hasta, tipo \in {comisiones, reservas, ventas}, filtros (terapeuta, categoría, estado).

Secuencia

1. Usuario define parámetros.
2. Sistema consulta y agrega datos.
3. Renderiza tabla y permite exportar.

Salidas

- Tabla en pantalla y archivo descargable (CSV/XLSX/PDF).

Excepciones

- Rangos inválidos \rightarrow mensaje y bloqueo.
- Sin datos \rightarrow estado vacío informativo

Relación entradas – salidas

- Filtros válidos \rightarrow resultados consistentes y exportables.

Requisitos lógicos de datos (BD)

- Vistas/materialized views para agregados frecuentes.
- Bitácora de exportaciones (quién, cuándo, qué parámetros)

Criterios de aceptación

- El reporte de comisiones por mes coincide con la suma de atenciones realizadas y reglas vigentes.

3.2.9 Requisito funcional 9 (RF09) – Notificaciones (email/SMS)

Actores: Cliente, Terapeuta, Admin/Jefatura.

Descripción: Enviar notificaciones transaccionales (registro, confirmación de reserva/pago, recordatorios, cambios de agenda, cancelaciones).

Entradas y validaciones

- tipo \in {registro, confirmacion_reserva, recordatorio_reserva, pago_aprobado, cambio_horario, cancelacion}.
- Destinatario (email válido; opcional telefono si SMS).
- Payload con datos mínimos (nombre, fecha/hora, servicio, código reserva).
- Validar existencia del canal configurado (email/SMS habilitado).

Secuencia

1. Usuario define parámetros.
2. Backend genera plantilla y cola el envío.
3. Proveedor externo procesa; backend registra resultado.

Salidas

- Estado de entrega (enviado, fallido, reintento).
- Log de notificación con traceid.

Excepciones

- Falla de proveedor \rightarrow encolar reintento exponencial (máx. N).
- Email inválido \rightarrow marcar como “rebotado” y alertar.

Relación entradas – salidas

- Evento válido + destinatario válido \rightarrow notificación enviada.
- Datos incompletos \rightarrow no se envía y se registra error.

Requisitos lógicos de datos (BD)

- Notificaciones(id, tipo, destinatario, canal, payload_json, estado, intento, proveedor_msg_id, creado_en)

Criterios de aceptación

- Recordatorios se envían 24 h y 2 h antes de la reserva (configurable).
- Tasa de reintentos ≤ 3 por notificación fallida.

3.2.10 Requisito funcional 10 (RF10) – Blog / Contenido (CMS básico)

Actores: Admin/Jefatura (gestiona), Cliente (lee).

Descripción: Publicar artículos (bienestar, promociones), con portada, categorías y detalle.

Entradas y validaciones

- titulo (oblig.), slug único, resumen, contenido_html, categoria, portada_url, estado ∈ {borrador, publicado}.
- Validar tamaño de imágenes y formato.

Secuencia

1. Admin crea/edita artículo y guarda como borrador o publica
2. Cliente ve listado paginado/filtrado y abre detalle por slug.

Salidas

- Listado paginado con portada y resumen; página de detalle.

Excepciones

- Slug duplicado → 409 Conflict.
- Imagen no válida → 422

Relación entradas – salidas

- Publicado → visible al cliente
- Borrador → oculto a público.

Requisitos lógicos de datos (BD)

- posts(id, titulo, slug*, categoria, resumen, contenido_html, portada_url, estado, publicado_en, creado_en, actualizado_en).

Criterios de aceptación

- El sitemap/listado responde en < 2 s (P95) con 20 ítems por página.

3.2.11 Requisito funcional 11 (RF11) – Promociones, cupones y descuentos

Actores: Admin/Jefatura (define), Cliente (usa).

Descripción: Aplicar promociones por servicio/categoría/fecha y cupones de descuento.

Entradas y validaciones

- tipo \in {porcentaje, monto_fijo, 2x1, paquete}.
- alcance \in {sku, categoria, total_carrito}.
- vigencia_desde/hasta, tope_descuento, stock_cupones, codigo (si cupón).
- Validar solapamientos y tope monetario

Secuencia

1. Admin crea promoción/cupón con reglas y vigencias.
2. Cliente aplica cupón en carrito; sistema valida y recalcula.
3. En checkout, se congela beneficio aplicado.

Salidas

- Totales con detalle de descuentos aplicados.
- Mensajes de éxito/error al aplicar cupón.

Excepciones

- Cupón vencido/sin stock \rightarrow 422.
- Regla no aplicable \rightarrow mensaje explicativo (por qué no aplica).

Relación entradas – salidas

- Regla válida \rightarrow descuento aplicado visible en resumen.
- No cumple \rightarrow sin descuento y feedback.

Requisitos lógicos de datos (BD)

- promociones(id, tipo, alcance, valor, vigencia_desde, vigencia_hasta, tope, reglas_json, estado).
- cupones(id, codigo*, promo_id, stock, canjeados, vigencia_desde, vigencia_hasta, estado).

Criterios de aceptación

- No se pueden combinar promociones conflictivas (se elige la mejor para el cliente si así se define).
- Cada cupón se descuenta del stock exactamente una vez por canje válido (idempotencia).

3.2.12 Requisito funcional 12 (RF12) – Gestión de salas y disponibilidad

Actores: Admin/Jefatura (gestiona), Cliente (consume disponibilidad).

Descripción: Mantener catálogo de salas (2 dobles, 1 individual) y sus bloqueos/maintenance.

Entradas y validaciones

- Sala(nombre, tipo \in {doble, individual}, estado).
- Bloqueo(sala_id, desde, hasta, motivo).
- Validar conflictos de bloqueos con reservas existentes (solo admin puede forzar)

Secuencia

1. Admin crea/edita salas.
2. Admin crea bloqueos por mantención; sistema recalcula disponibilidad.
3. Cliente ve solo slots libres considerando duración del servicio.

Salidas

- Agenda por sala con estados (libre, ocupado, bloqueado).
- Mensajes si un slot se vuelve inválido por bloqueo nuevo.

Excepciones

- Bloquear sala con reservas \rightarrow advertir y ofrecer reprogramación.

Relación entradas – salidas

- Bloqueo válido \rightarrow slots ocultos a clientes.
- Eliminación de bloqueo \rightarrow slots vuelven a mostrarse.

Requisitos lógicos de datos (BD)

- salas(id, nombre, tipo, estado).
- bloqueos_sala(id, sala_id, desde_ts, hasta_ts, motivo, creado_en).

Criterios de aceptación

- El motor de disponibilidad no ofrece slots que se solapen con bloqueos ni reservas confirmadas).

3.2.13 Requisito funcional 13 (RF13) – Reseñas y satisfacción (NPS)

Actores: Cliente (opina), Admin/Jefatura (lee).

Descripción: Permitir calificar y comentar tras una atención, y medir NPS.

Entradas y validaciones

- reserva_id, rating (1–5), comentario (máx. 500), nps (0–10).
- Validar que la reserva fue “realizada” y pertenece al cliente.

Secuencia

1. Tras la atención, se invita a calificar por email/app.

2. Cliente envía rating y comentario (opcional).
3. Sistema agrega al promedio del servicio/terapeuta y muestra en admin.

Salidas

- Promedios por servicio/terapeuta; listado moderable.

Excepciones

- Intento de reseñar una reserva no realizada → bloquear.
- Lenguaje inapropiado → marcar para moderación (reglas básicas).

Relación entradas – salidas

- Opinión válida → visible (si política es pública) o solo en admin.

Requisitos lógicos de datos (BD)

- resenas(id, reserva_id*, usuario_id, sku, terapeuta_id?, rating, comentario, nps, estado, creado_en).

Criterios de aceptación

- Un cliente solo puede reseñar una vez por reserva.
- La media mostrada corresponde a los últimos 12 meses (configurable).

3.2.14 Requisito funcional 14 (RF14) – Contacto y tickets de soporte

Actores: Cliente, Admin/Jefatura (responde).

Descripción: Canal para enviar consultas/reclamos y darles seguimiento.

Entradas y validaciones

- asunto, mensaje, email, adjuntos (opcional).
- Validar formato de email y tamaño/tipo de adjuntos.

Secuencia

1. Cliente envía formulario de contacto.
2. Sistema crea ticket y notifica a admin.
3. Admin responde y cambia estado (abierto, en_progreso, resuelto, cerrado).

Salidas

- ID de ticket; comunicaciones por email.

Excepciones

- Adjuntos no permitidos → 415 Unsupported Media Type.
- Respuesta sin ticket válido → 404.

Relación entradas – salidas

- Formulario válido → ticket creado, notificación enviada.

Requisitos lógicos de datos (BD)

- tickets(id, usuario_id?, email, asunto, mensaje, estado, prioridad, creado_en, actualizado_en).

Criterios de aceptación

- Los tickets reciben acknowledgment automático por email con su ID.

3.2.15 Requisito funcional 15 (RF15) – Políticas de cancelación y reembolsos

Actores: Cliente, Admin/Jefatura.

Descripción: Gestionar cancelaciones con reglas por ventana de tiempo y posibles cargos.

Entradas y validaciones

- reserva_id, motivo, solicitante (cliente/admin), momento (fecha/hora de solicitud).
- Validar ventana de cancelación (ej.: gratis ≥ 24 h; cargo 50% < 24 h; no reembolsable < 2 h).

Secuencia

1. Cliente solicita cancelación; sistema evalúa regla y calcula cargo.
2. Si pagado, se gestiona reversa parcial/total (según pasarela).
3. Estado de reserva → cancelada y notificación.

Salidas

- Confirmación de cancelación con detalle de cargos/reembolsos.

Excepciones

- No se puede cancelar tras “atendida” → 409.
- Error en reversa con pasarela → registrar disputa y escalar.

Relación entradas – salidas

- Dentro de ventana → reembolso/cargo según política.
- Fuera de ventana → rechazo justificado.

Requisitos lógicos de datos (BD)

- cancelaciones(id, reserva_id*, solicitante, motivo, cargo, reembolso, estado, creado_en).
- politicas_cancelacion(id, regla_json, vigente_desde, vigente_hasta, estado).

Criterios de aceptación

- La política aplicada queda auditada (qué regla, versión, cálculo).

3.2.16 Requisito funcional 16 (RF16) – Integración con pasarela de pago (callback)

Actores: Cliente, Pasarela (sistema externo), Admin/Jefatura.

Descripción: Procesar callbacks de la pasarela y consolidar el estado del pago.

Entradas y validaciones

- callback firmado (token o HMAC), status \in {aprobado, rechazado, pendiente}, monto, orden_id.
- Validar integridad del mensaje y que orden_id exista y esté en estado correcto.

Secuencia

1. Cliente paga y la pasarela llama el endpoint de callback/return.
2. Backend valida firma y actualiza transacción/reserva.
3. Se notifica a cliente (email) y se registra auditoría.

Salidas

- Estado final de la transacción y de la reserva.

Excepciones

- Firma inválida \rightarrow 401 + log de seguridad.
- Monto inconsistente \rightarrow 409 y marcar como “en revisión”.

Relación entradas – salidas

- Aprobado \rightarrow reserva “pagada” y confirmada.
- Rechazado \rightarrow reserva “pendiente de pago” o cancelada según regla.

Requisitos lógicos de datos (BD)

- pagos ya definido: agregar campos pasarela, raw_callback_json, firma_valida.
- audit_logs(id, actor, accion, entidad, entidad_id, detalle_json, creado_en).

Criterios de aceptación

- Los callbacks son idempotentes: múltiples llamados del mismo orden_id no duplican cambios.
- Todo callback se traza con traceld y queda disponible en auditoría

3.3 Requisitos no funcionales

3.3.1 Requisitos de rendimiento

Especificación de los requisitos relacionados con la carga que se espera tenga que soportar el sistema.

- Rendimiento percibido (frontend)
 - RP-01: El First Contentful Paint (FCP) ≤ 2.0 s en 4G “Good” (P95).
 - RP-02: El Time to Interactive (TTI) ≤ 3.0 s en 4G “Good” (P95).
 - RP-03: Las navegaciones internas SPA deben renderizar vista en < 600 ms (P95).
- Latencia de API (backend)
 - RP-04: Endpoints de consulta (GET /servicios, GET /reservas) deben responder en < 300 ms (P95) y < 800 ms (P99).
 - RP-05: Endpoints de mutación (POST /reservas, POST /pagos/callback) deben responder en < 700 ms (P95) y < 1500 ms (P99).
- Capacidad / Concurrencia
 - RP-06: El sistema soporta 200 usuarios concurrentes con $\leq 1\%$ de error 5xx y manteniendo latencias P95 definidas.
 - RP-07: El motor de agenda soporta 10 transacciones/seg de creación/cambio de reservas sin conflictos perdidos (ver ACID).
- Carga y datos
 - RP-08: Exportación de reportes de hasta 50.000 filas en < 10 s (genera CSV/XLSX de forma asíncrona).
- Recursos
 - RP-09: Uso de CPU backend $< 70\%$ sostenido a carga nominal; memoria $< 75\%$ (promedio en 15 min).

3.3.2 Seguridad

Especificación de elementos que protegerán al software de accesos, usos y sabotajes maliciosos, así como de modificaciones o destrucciones maliciosas o accidentales. Los requisitos pueden especificar:

- Autenticación / Autorización
 - SEG-01: Autenticación JWT (access + refresh), expiración: access ≤ 30 min, refresh ≤ 30 días.

- SEG-02: Control de acceso por roles (CLIENTE, TERAPEUTA, ADMIN/JEFATURA) a nivel de endpoint y UI.
- Protección de datos
 - SEG-03: HTTPS obligatorio (TLS 1.2+); redirección HTTP→HTTPS.
 - SEG-04: Contraseñas cifradas con bcrypt (cost ≥ 10). Nunca se loguean.
 - SEG-05: Datos sensibles en tránsito cifrados; BD en reposo con cifrado provisto por el cloud.
- OWASP / Mitigaciones
 - SEG-06: Validación de entrada en backend; uso de prepared statements/JPA (prevención de SQLi).
 - SEG-07: CSRF mitigado (tokens/headers apropiados) donde aplique.
 - SEG-08: XSS mitigado (escape de salida; CSP recomendada).
 - SEG-09: Rate limiting en /auth/* y callbacks de pago (ej. ≤ 10 req/min/IP).
- Logs / Auditoría
 - SEG-10: Logs estructurados (JSON) con `traceId`, sin PII sensible.
 - SEG-11: Auditoría para acciones críticas (login, CRUD servicios, estado de reservas/pagos).
- Gestión de sesiones
 - SEG-12: Revocación de refresh tokens en logout; lista de deny en servidor.

3.3.3 Fiabilidad

Especificación de los factores de fiabilidad necesaria del sistema. Esto se expresa generalmente como el tiempo entre los incidentes permisibles, o el total de incidentes permisible.

- Errores e incidentes
 - FIAB-01: Tasa de errores 5xx $< 0.5\%$ mensual.
 - FIAB-02: MTTR (tiempo medio de recuperación) < 2 h para incidentes de severidad alta.
- Persistencia
 - FIAB-03: BD con backups diarios automáticos, retención ≥ 7 días, prueba de restauración mensual.
- Idempotencia
 - FIAB-04: Endpoints de pago/callback son **idempotentes** (reintentos no duplican efectos).

3.3.4 Disponibilidad

Especificación de los factores de disponibilidad final exigidos al sistema.

- DISP-01: Disponibilidad del servicio (ventana operativa) $\geq 99.0\%$ mensual.
- DISP-02: Mantenimiento programado anunciado con ≥ 24 h de anticipación.
- DISP-03: Health checks y endpoints /actuador/health monitorizados.

3.3.5 Mantenibilidad

Identificación del tipo de mantenimiento necesario del sistema.

- Código / Arquitectura
 - MANT-01: Separación frontend (React) / backend (Spring Boot) con contratos OpenAPI versionados.
 - MANT-02: Cobertura de pruebas unitarias $\geq 60\%$ backend; E2E críticos (auth, reserva, pago).
 - MANT-03: Linter/formatter activos; revisión de PRs (4-eyes rule) para código de producción.
- Despliegue
 - MANT-04: CI/CD con build reproducible (Maven), despliegues blue/green o rolling.
- Observabilidad
 - MANT-05: Métricas (latencia, error rate, throughput) y alertas (umbral P95, 5xx, espacio BD, GC).

3.3.6 Portabilidad

Especificación de atributos que debe presentar el software para facilitar su traslado a otras plataformas u entornos.

- PORT-01: Frontend compatible con Chrome, Edge, Firefox y Safari (últimas 2 versiones) y iOS/Android modernos.
- PORT-02: Backend ejecutable en JDK 17, empaquetado en container OCI (Docker) y orquestable (ECS/K8s).
- PORT-03: Persistencia en PostgreSQL ≥ 13 ; capa de acceso via JPA (evitar SQL vendor-locking).
- PORT-04: Almacenamiento de archivos estáticos (imágenes) desacoplado (ej. S3/Blob Storage).

3.4 Otros Requisitos

- Internacionalización (i18n)
 - OR-01: Textos centralizados y preparados para multi-idioma (ES, EN).
- Cumplimiento legal
 - OR-02: Adecuación a Ley 19.628 (Chile) sobre protección de datos personales; consentimiento explícito y política de privacidad visible.
- Usabilidad
 - OR-03: Tareas clave (buscar servicio, reservar, pagar) completables por usuarios novatos en ≤ 3 min (pruebas de usabilidad).
- Soporte y atención
 - OR-04: Tiempo de respuesta a tickets de clientes < 24 h hábiles; SLA interno.

Ver Anexo 2 – Planilla de requerimientos