



DOCUMENTO FUNCIONAL DEL PROYECTO

ALUMNO/GRUPO: Iván Martín Bodas 2DAM A

1. Introducción y contexto

Objetivo:

Se desarrollará una aplicación que permita a los clientes reservar citas, pagar por adelantado con Stripe y recibir sugerencias de cortes de cabello mediante inteligencia artificial. La aplicación está desplegada en producción y accesible desde cualquier dispositivo en <https://2look-app.vercel.app>.

2. Análisis de requisitos

2.1. Requisitos funcionales (RF)

Qué debe hacer el sistema. Enumera las funciones principales, numeradas como RF1, RF2, etc.

Código	Descripción del requisito funcional
RF1	El sistema permite registrar y gestionar usuarios (clientes y administradores).
RF2	El sistema permite iniciar sesión con email y contraseña, así como acceder como invitado sin necesidad de registro.
RF3	El sistema muestra un catálogo actualizado de barberos y servicios.
RF4	El sistema permite reservar citas indicando día, hora y barbero.

RF5	El sistema permite realizar pagos de forma segura mediante Stripe.
RF6	El sistema confirma la reserva mediante pantalla de confirmación y la guarda en la base de datos.
RF7	El sistema permite a los administradores consultar y cancelar reservas desde la aplicación.
RF8	El sistema almacena el historial de cortes y los barberos favoritos de cada cliente.
RF9	El sistema permite publicar y consultar opiniones y reseñas de barberos.
RF10	El sistema permite al usuario activar un recordatorio de cita tras confirmar el pago.
RF11	El sistema analiza la foto del cliente mediante inteligencia artificial y sugiere cortes de pelo.
RF12	El sistema permite subir una foto o usar la cámara del dispositivo para el análisis facial.
RF13	El sistema genera imágenes de referencia de cada corte recomendado mediante fal.ai (modelo fast-sdxl). Si la API de Gemini supera su límite de cuota, el sistema rota automáticamente entre tres claves independientes y, si todas fallan, activa un modo demo con resultados predefinidos.

2.2. Requisitos no funcionales (RNF)

Cómo debe comportarse el sistema.

Incluye aspectos como rendimiento, seguridad, compatibilidad o facilidad de uso.

Código	Descripción del requisito no funcional
RNF1	La interfaz será accesible desde dispositivos móviles y de escritorio.

RNF2	La aplicación debe cargar las páginas en menos de tres segundos.
RNF3	El sistema garantiza la seguridad de los datos de los usuarios y los pagos.
RNF4	El sistema debe ser estable y confiable en su funcionamiento.
RNF5	La aplicación cumple con la normativa de protección de datos vigente.

2.3. Restricciones

- Tecnologías obligatorias: TypeScript, Ionic + Vue 3 (frontend), Java + Spring Boot (backend), PostgreSQL (Aiven Cloud), Stripe API, Gemini API, fal.ai
- Entrega: La entrega será el 17 de mayo
- El personal: Puede ser un problema ya que para una persona es un proyecto ambicioso y el tiempo disponible es una limitación clave.
- Recursos disponibles: ordenador con entorno de desarrollo, conexión a internet, herramientas de diseño y programación.
- Dependencias técnicas: integración con APIs externas (IA, pagos), frameworks front-end y back-end.

3. Análisis de usuarios y roles

Objetivo: identificar quién usará el sistema y qué podrá hacer.

Describe los distintos tipos de usuario, sus necesidades y sus permisos.

Rol	Descripción	Permisos principales
Administrador	Gestiona usuarios, reservas y horarios	Crear, modificar y eliminar usuarios; consultar reservas; gestionar horarios y tarifas; asignar permisos

Barbero	Gestiona su agenda y ofrece servicios	Consultar las reservas asignadas y marcar citas como completadas
Cliente	Solicita servicios y recibe recomendaciones	Crear reservas, pagar, subir foto para IA, dejar reseñas, consultar historial
Visitante	Consulta información básica	Consultar catálogo de servicios y barberos

4. Casos de uso / Escenarios de uso

Objetivo: mostrar cómo interactúan los usuarios con el sistema.

Selecciona de tres a cinco casos principales y descríbelos brevemente.

Código	Nombre del caso de uso	Actor principal	Descripción	Resultado esperado
CU1	Registrar usuario	Visitante	El usuario introduce sus datos y se crea una cuenta.	Usuario registrado correctamente.
CU2	Iniciar sesión	Usuario	El usuario introduce sus credenciales y accede al sistema.	Acceso concedido.
CU3	Realizar una reserva	Cliente	El cliente selecciona día, hora y barbero y paga mediante Stripe.	Reserva creada y confirmada
CU4	Análisis de rostro	Cliente	El cliente sube una foto o utiliza la cámara para el análisis facial mediante IA.	El sistema recomienda cortes de pelo personalizados
CU5	Gestión de reservas	Administrador/Barbero	El administrador o barbero consulta reservas y bloquea horarios.	Horarios y reservas actualizadas correctamente

5. Modelo de datos o estructura de la información

Entidades principales:

- Persona(cliente, barbero, administrador)
- Reserva
- Barbero
- Servicio
- Historial de cortes
- Opinión

Estructura

1. Persona

(Contiene la información básica de todas las personas que acceden al sistema.)

Atributo Tipo de Dato Clave Restricciones

id INTEGER PK Clave primaria, autoincremental

nombre VARCHAR(100) Obligatorio

apellidos VARCHAR(100) Obligatorio

email VARCHAR(255) Única Obligatorio, usado para inicio de sesión

contraseña VARCHAR(255) Hash seguro

rol VARCHAR(20) ('Cliente', 'Barbero', 'Administrador')

fecha_registro TIMESTAMP Fecha de creación de la cuenta

2. Barbero

(Datos específicos del personal que realiza los servicios. Extiende la tabla Persona.)

Atributo Tipo de Dato Clave Restricciones

id_barbero INTEGER PK, FK Referencia a Persona.id (PK)

especialidad VARCHAR(100)

valoracion_media NUMERIC(2, 1) Valoración de 1.0 a 5.0

3. Servicio

(El catálogo de servicios que ofrece la barbería.)

Atributo Tipo de Dato Clave Restricciones

id INTEGER PK Clave primaria, autoincremental

nombre VARCHAR(100) Obligatorio (Ej: "Corte Caballero", "Arreglo Barba")

descripcion TEXT

precio NUMERIC(10, 2) Obligatorio

duracion_min INTEGER Duración del servicio en minutos

4. Reserva

(Tabla central que registra cada cita solicitada.)

Atributo Tipo de Dato Clave Restricciones

id INTEGER PK Clave primaria, autoincremental

cliente_id INTEGER FK Referencia a Persona.id

barbero_id INTEGER FK Referencia a Barbero.id_barbero

servicio_id INTEGER FK Referencia a Servicio.id

fecha DATE

hora_inicio TIME

estado VARCHAR(20) ('Confirmada', 'Pendiente', 'Cancelada', 'Completada')

pago_stripe_id VARCHAR(255) Única ID de la transacción en Stripe (RF5)

cantidad_pagada NUMERIC(10, 2) Cantidad final pagada por la reserva

5. Historial_de_Cortes

(Registro de cortes anteriores y análisis de IA para cada cliente.)

Atributo Tipo de Dato Clave Restricciones

id INTEGER PK Clave primaria, autoincremental

cliente_id INTEGER FK Referencia a Persona.id (RF8)

reserva_id INTEGER FK Referencia a Reserva.id

corte_sugerido_ia VARCHAR(255) Nombre del corte sugerido (RF11)

corte_realizado VARCHAR(255) Nombre del corte finalizado

foto_antes VARCHAR(255) URL de la imagen (Opcional, RF12)

6. Opinión

(Almacena el feedback de los clientes sobre los barberos.)

Atributo Tipo de Dato Clave Restricciones

id INTEGER PK Clave primaria, autoincremental

cliente_id INTEGER FK Referencia a Persona.id

barbero_id INTEGER FK Referencia a Barbero.id_barbero (RF9)

puntuación INTEGER Rango de 1 a 5

comentario TEXT

fecha TIMESTAMP

7. Favorito

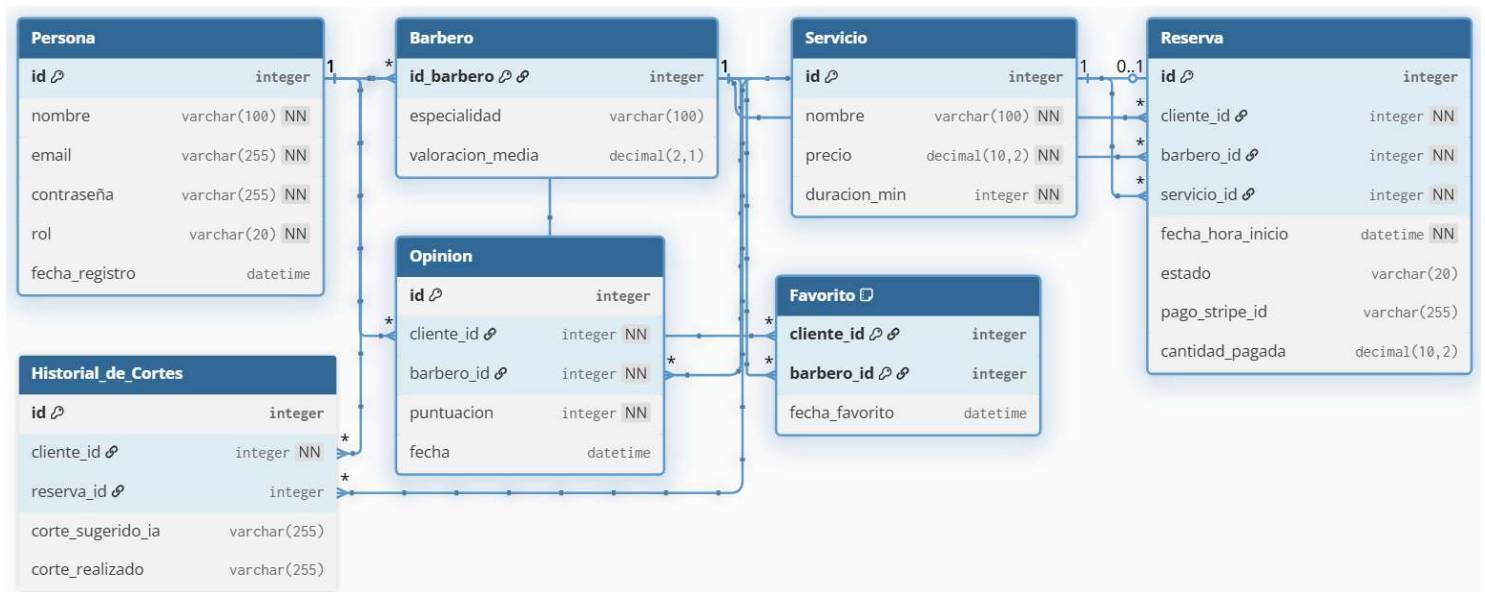
(Tabla de unión para manejar la relación N:M entre clientes y barberos.)

Atributo Tipo de Dato Clave

cliente_id INTEGER PK, FK

barbero_id INTEGER PK, FK

fecha_favorito TIMESTAMP



6. Diseño de la interfaz

Pantallas principales:

1. **Registro / Inicio de sesión:** Gestiona la creación de cuentas, autenticación y recuperación de contraseñas.
2. **Inicio / Catálogo:** Punto de partida para la reserva y vista de servicios
3. **Mis Citas:** Gestión de reservas activas
4. **IA / Sugerencias:** Acceso directo al Análisis Facial
5. **Historial:** Registro de cortes, reseñas y Favoritos
6. **Perfil:** Configuración de usuario y logout.

Cada pantalla está diseñada para ofrecer una navegación intuitiva, accesible desde dispositivos móviles y garantizar que las funcionalidades principales estén disponibles de manera clara y rápida.

7. Planificación técnica

Tecnologías y herramientas:

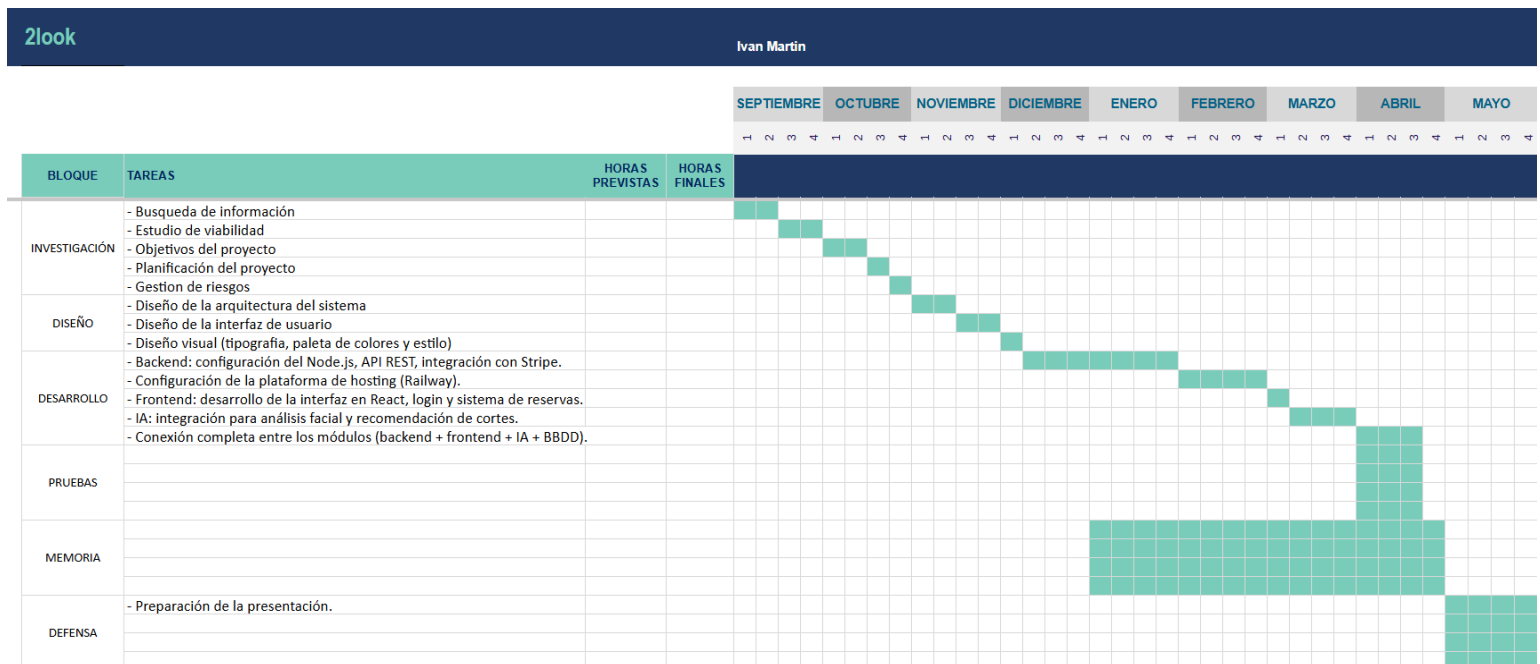
Capa	Componente	Tecnología Principal	Función y Propósito en el Proyecto
Cliente / Interfaz	Front-end	Ionic + Vue 3	Desarrollo de la app móvil multiplataforma (Android, iOS) con TypeScript. Conecta con el backend mediante peticiones HTTP y con Gemini y fal.ai directamente desde el cliente.
Servidor / Lógica	Back-end	Java + Spring Boot	Creación de la API REST. Gestiona la seguridad con Spring Security, los servicios de citas y la lógica de negocio principal.
Servidor / Persistencia	Base de Datos	PostgreSQL + Spring Data JPA	Generación de imágenes de datos. Hibernate mapeará automáticamente tus tablas basándose en tus clases Java (Entities).

Integración / Pagos	Pagos	Stripe Java SDK	Implementación segura de la pasarela de pagos en modo test para validar las transacciones de reserva (RF5).
Integración / IA	Inteligencia Artificial	Gemini API	Motor de IA para el análisis facial y la generación de sugerencias personalizadas de cortes de pelo (RF11).
Integración / Imágenes IA	Generación de imágenes	fal.ai	Generación de imágenes de referencia de los cortes recomendados por la IA mediante el modelo fast-sdxl (image generation). Las llamadas se realizan directamente desde el frontend.
Infraestructura	Despliegue	Render	Hosting del backend (Java + Docker, Java 21) en https://twolookapp.onrender.com . El frontend está desplegado en Vercel (https://2look-app.vercel.app). La base de datos PostgreSQL está en Aiven Cloud. UptimeRobot pingea el backend cada 5 minutos para evitar

			que Render duerma el servidor.
--	--	--	--------------------------------

Cronograma (puede incluir un diagrama de Gantt):

<https://docs.google.com/spreadsheets/d/1kkhkgdE267K4hviUemtx6EvZel9EicOd/edit?usp=sharing&oid=10386496771621855078&rtpof=true&sd=true>



8. Análisis de riesgos

8.1. Identificación de riesgos

- Falta de tiempo o retrasos en la planificación.
- Problemas técnicos relacionados con la integración de Stripe o la IA.
- Fallo en el despliegue del backend en la nube.
- Pérdida de datos de usuarios o reservas.
- Cambios en los requisitos durante el desarrollo

8.2. Valoración y respuesta

Clasifica cada riesgo según su probabilidad e impacto, e indica cómo se mitigará

Riesgo	Probabilidad	Impacto	Plan de prevención o contingencia
Falta de tiempo	Alta	Alta	Dividir tareas en entregas intermedias y adherirse estrictamente al MVP.
Problemas técnicos (Stripe/Gemini)	Media	Alta	Probar las integraciones en Render (la nube) lo antes posible. Usar fal.ai simplifica la gestión de imágenes.
Fallo en el despliegue	Media	Media	Usar Render con Docker para el backend y Vercel para el frontend, ambas plataformas con despliegue automático desde GitHub.

Pérdida de datos	Baja	Alta	Realizar copias de seguridad semanales de la base de datos (PostgreSQL).
Cambios en requisitos	Media	Media	Documentar y priorizar cambios según impacto en funcionalidades esenciales.

9. Validación y criterios de éxito

Criterios de aceptación:

- Los usuarios pueden registrarse, iniciar sesión y gestionar sus datos de manera segura.
- Los clientes pueden reservar citas indicando día, hora y barbero.
- Los pagos mediante Stripe se procesan de forma segura y correcta.
- La IA sugiere cortes de cabello personalizados basados en la forma del rostro del cliente.
- Los administradores pueden consultar y cancelar reservas, y disponen de intentos ilimitados en el asesor de IA.

Pruebas previstas:

- Funcionales: Verificación del registro, reservas y pagos.
- De usuario: Simulación de flujo completo de cliente y administrador.
- De rendimiento: Validación de tiempos de carga y respuesta de la aplicación.

Indicadores de calidad:

- Reserva y confirmación de citas sin incidencias.
- Sugerencias de cortes mediante IA acertadas en un porcentaje elevado de casos.
- Aplicación estable y sin errores críticos.

10. Conclusión

Resumen de decisiones principales:

- **Funciones del sistema:** El sistema permitirá registrar y gestionar usuarios, consultar catálogos de barberos y servicios, reservar citas indicando día, hora y barbero, realizar pagos seguros mediante Stripe, recibir confirmaciones de reserva, acceder a sugerencias de cortes personalizadas mediante IA, gestionar historial de cortes y barberos favoritos, publicar y consultar opiniones y activar recordatorios de cita.
- **Tecnologías a utilizar:** TypeScript e Ionic con Vue 3 para el front-end, Java con Spring Boot para el back-end, PostgreSQL como base de datos, Stripe para la gestión de pagos y la API de Gemini con rotación automática entre tres claves independientes para el análisis facial, y fal.ai para la generación de imágenes de referencia de los cortes recomendados.
- **Valor del proyecto:** El sistema ofrece una experiencia de usuario personalizada, mejora la organización y planificación de las reservas en barberías, reduce cancelaciones, facilita la gestión del negocio y aporta un componente innovador mediante recomendaciones de cortes basadas en inteligencia artificial.

Próximos pasos:

1. Preparar el entorno de desarrollo y configurar el repositorio inicial.
2. Crear la estructura de la base de datos y del backend.
3. Iniciar la implementación de los casos de uso prioritarios: registro e inicio de sesión, reservas y pagos, panel de administración y análisis facial mediante IA.

Reflexión final:

El análisis funcional proporciona una visión clara y organizada del proyecto, estableciendo una base sólida para la fase de desarrollo y asegurando que las funcionalidades cumplan con los objetivos propuestos.