

Health-Hub

La app que revoluciona tu forma de pedir cita

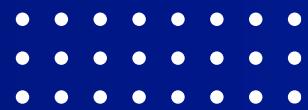
Presented By

Javier Balbás
Andrea Bonzi
David Perez
Julio Semper

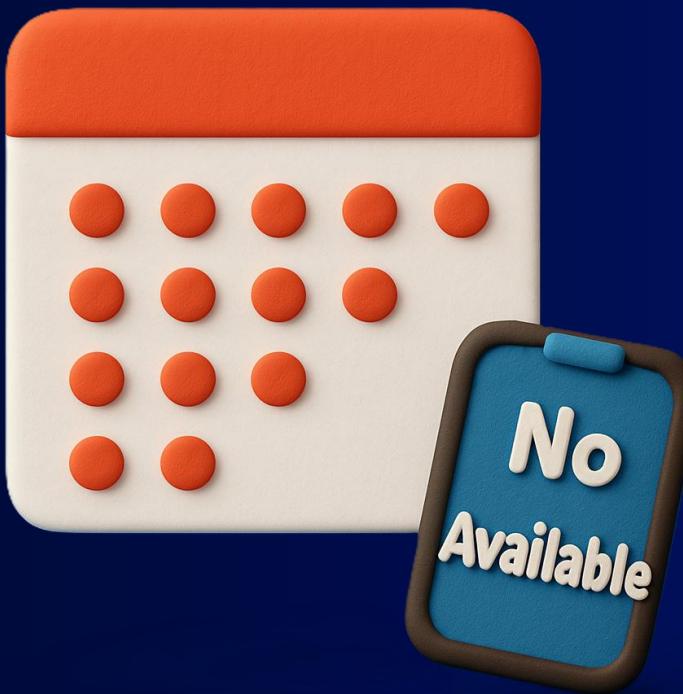
Date

9 Diciembre 2025

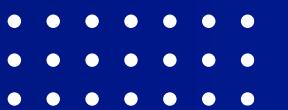




Problema Actual



- 1- Cada aseguradora y centro medico utiliza su propio sistema de gestión**
- 2- Cuando un paciente recibe la indicación de realizarse una prueba, debe llamar a todos los centros que su aseguradora le proporciona**
- 3- Cuando la consigue, debe volver a contactar para cancelar y quedarse con la más temprana**
- 4- No existe un lugar único donde consultar la disponibilidad de distintos centros y aseguradoras**
- 5- Esto convierte el proceso en algo fragmentado e ineficiente, generando frustración tanto en pacientes como en el personal sanitario**



Propuesta

El paciente deja de perseguir citas

Único lugar donde consultar disponibilidad de todos los centros en el radio

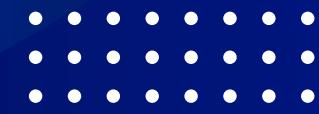
Experiencia

El paciente obtiene automáticamente la mejor opción disponible entre su aseguradora y centros asociados

Eficiencia para centros y aseguradoras

*Menos llamadas y más tiempo para lo que realmente importa:
las personas*





¿Qué aporta?

Pacientes

Un único lugar para ver disponibilidad real de múltiples centros
Menos llamadas, menos esperas y menos frustración
Citas más rápidas al mostrar siempre la opción más temprana

Médicos

Gestión de agenda más sencilla
Mejor seguimiento clínico
Comunicación directa y clara

Centro

Gestión centralizada de usuarios y personal
Métricas y analítica integradas
Sistema estable y rápido





Autenticación y Acceso

- Registro de nuevos usuarios.
- Inicio de sesión seguro.
- Obtención del perfil del usuario autenticado.
- Redirección según rol y protección de páginas privadas.

Gestión de Citas (Paciente)

- Buscar disponibilidad por especialidad y centro.
- Listado de médicos según filtros.
- Crear una nueva cita médica.
- Ver y cancelar citas propias.

• • • •

Requisitos Funcionales (RF)

Los requisitos funcionales son las funciones que el sistema debe realizar, como crear una cita o iniciar sesión

Gestión de Citas (Médico)

- Ver agenda completa.
- Marcar cita como completada.
- Gestionar órdenes médicas para pacientes.
- Responder mensajes del paciente.

Administración del Sistema

- Listar usuarios del sistema.
- Crear, editar y eliminar usuarios (CRUD).
- Crear, actualizar y eliminar centros médicos.
- Activar/desactivar centros médicos.
- Búsqueda en tablas administrativas.

Experiencia de Usuario

- Pagos online para citas.
- Cálculo detallado del costo (subtotal + IVA + descuentos).
- Sistema de puntos y recompensas.
- Centro de notificaciones.
- Panel de métricas y analíticas.



Seguridad

- JWT + bcrypt para autenticación segura.
- Control de acceso por roles.
- Protección XSS y rate-limiting.

Rendimiento

- Respuesta de API < 500ms.
- Dashboards rápidos (LCP < 2.5s).

Requisitos No Funcionales (RNF)

Los requisitos no funcionales describen cómo debe funcionar el sistema, por ejemplo, rápido, seguro y accesible.

Usabilidad / Accesibilidad:

- Diseño responsive mobile-first.
- Cumplimiento WCAG.
- Feedback visual inmediato (toasts).

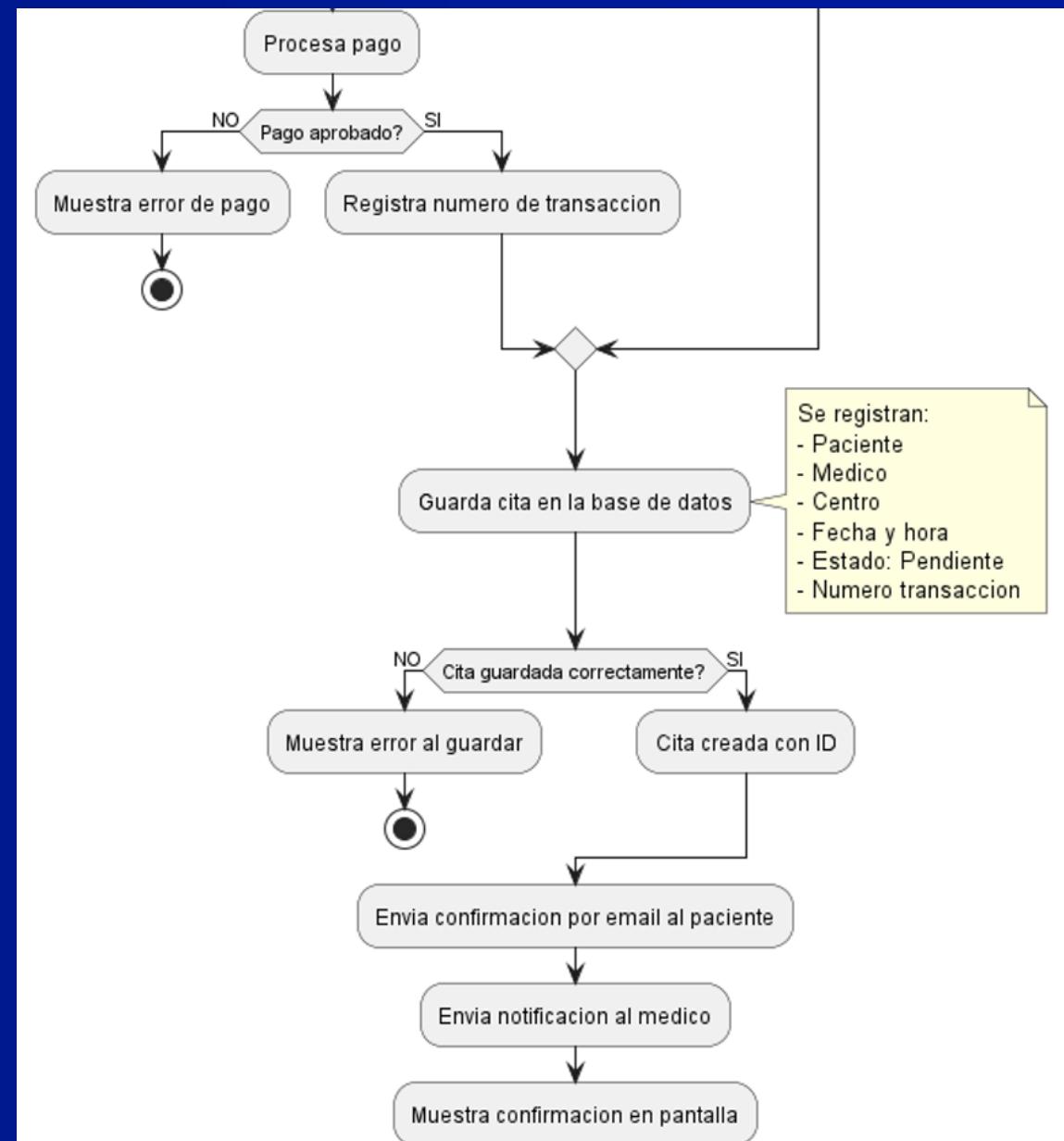
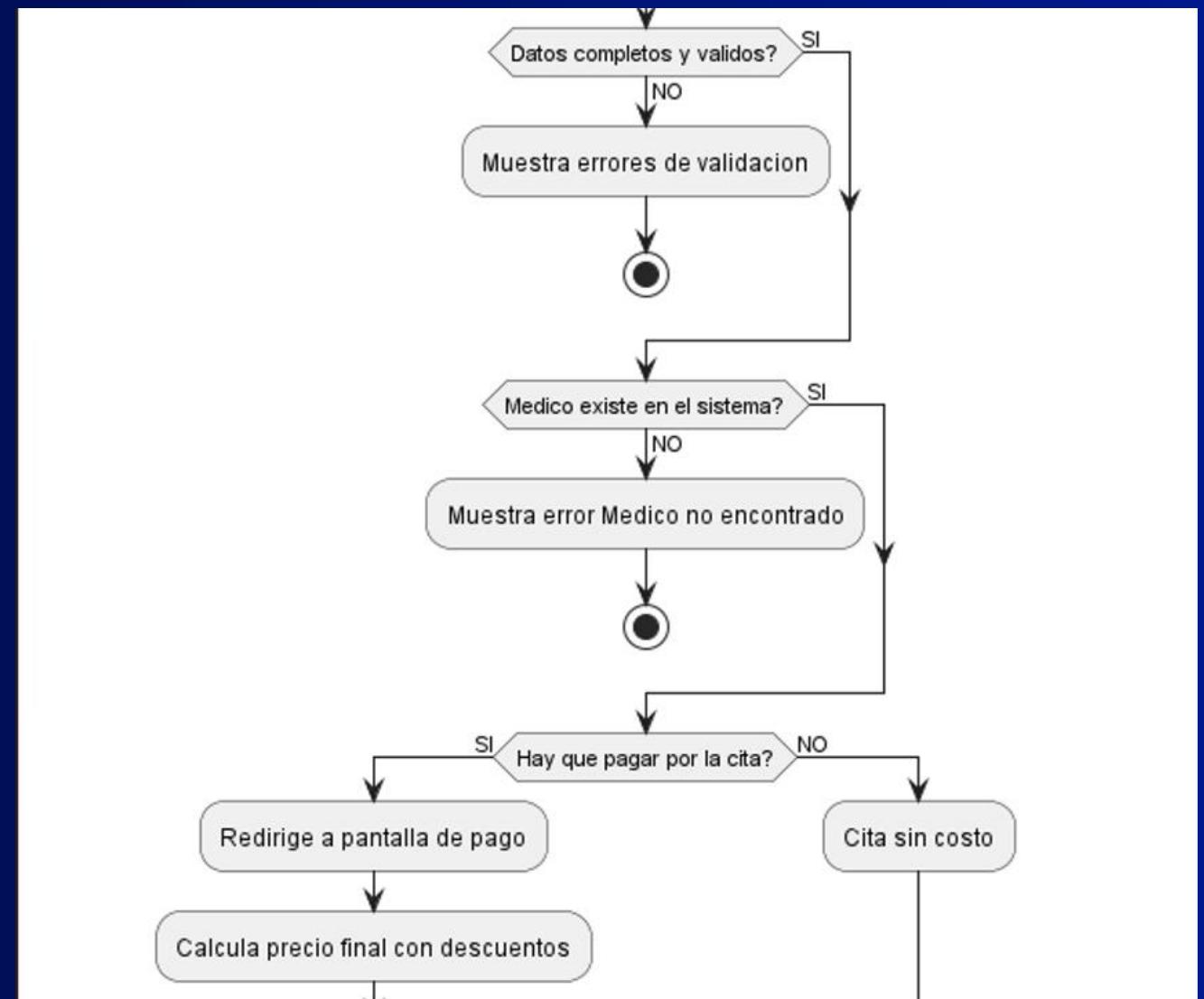
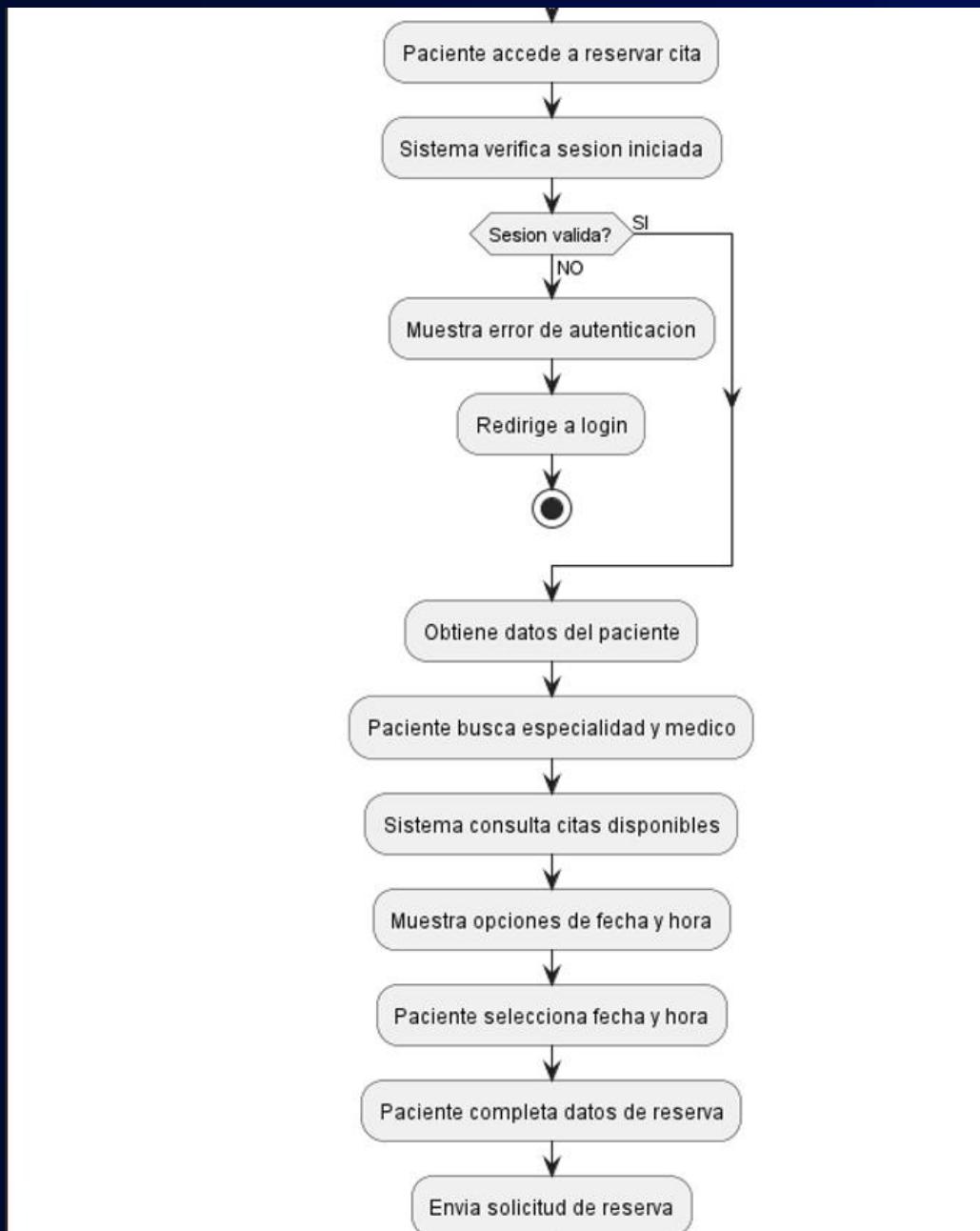
Mantenibilidad / Escalabilidad

- Arquitectura por capas (rutas, controladores, modelos).
- Base de datos flexible (MongoDB).





"Reservar cita"

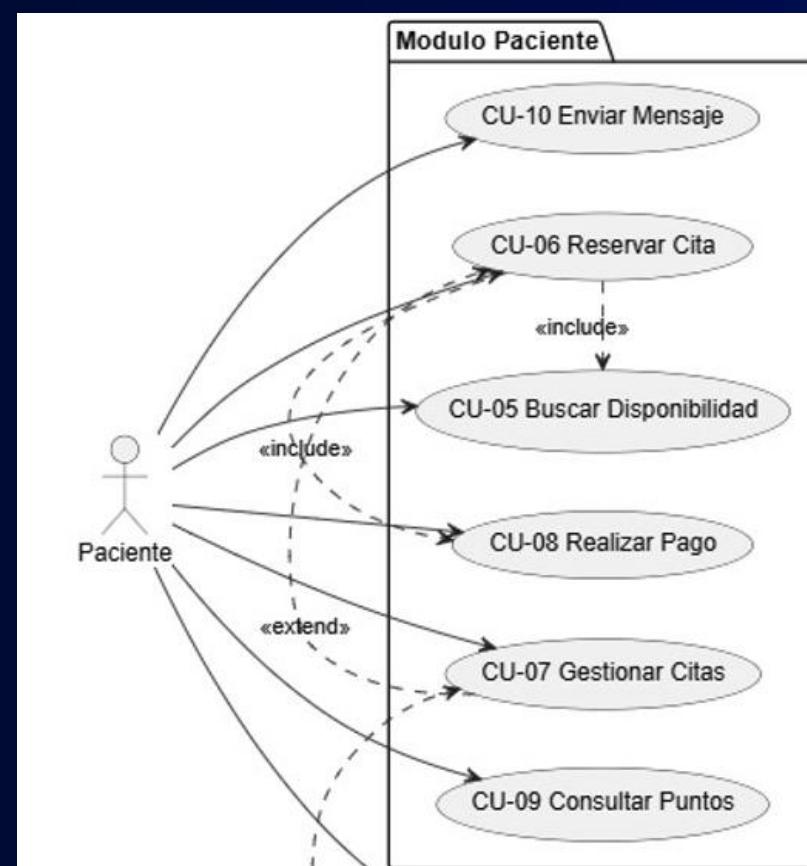




Casos de Uso

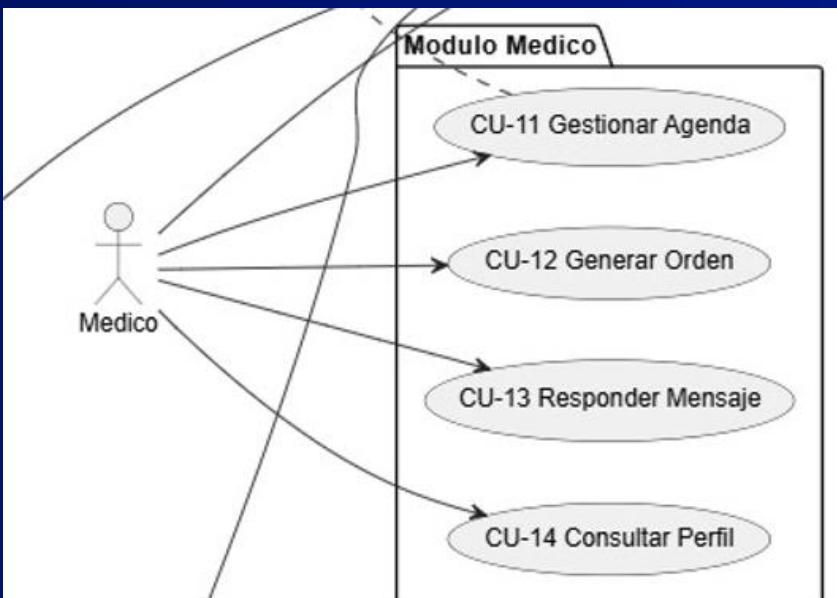
Paciente

- El **Paciente** puede buscar disponibilidad, reservar citas, pagarlas, gestionarlas y comunicarse con su médico.



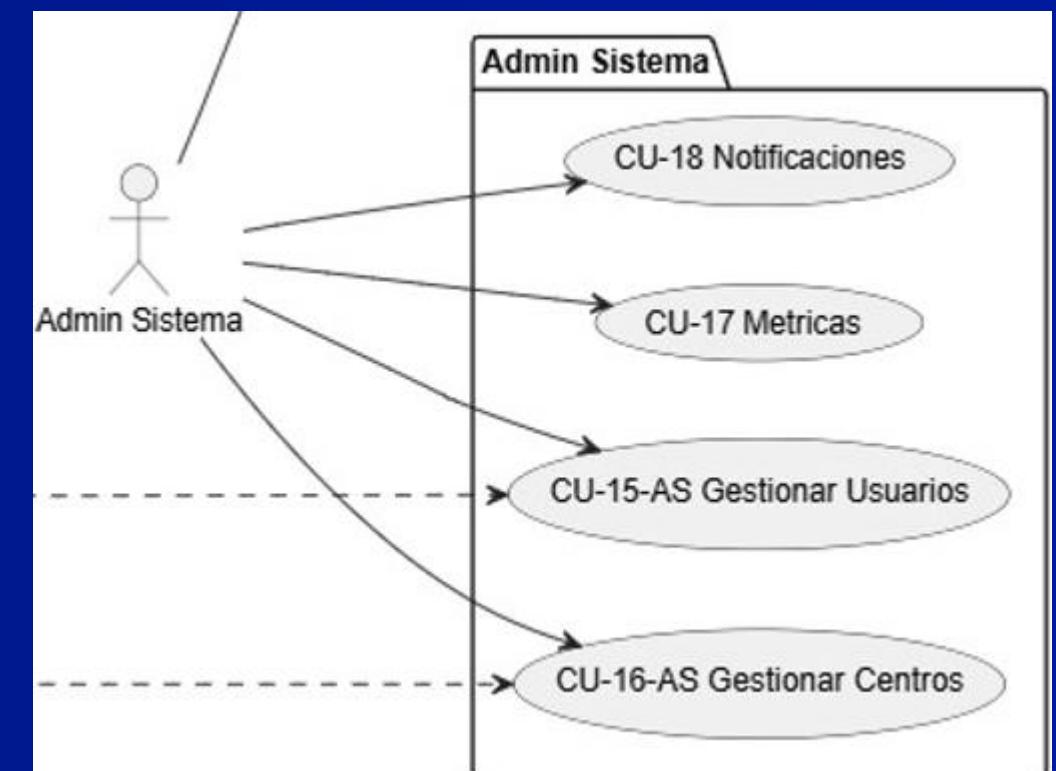
Médico

- El Médico puede consultar su agenda, completar citas, generar órdenes médicas y responder mensajes.



Administrador del Sistema

- El **Administrador del Sistema** gestiona usuarios, centros, métricas y notificaciones.



Administrador del Centro

- El **Administrador de Centro** administra únicamente los recursos y personal de su propio centro.



Diagrama de secuencia

Fase de autenticación y disponibilidad

En la primera fase, el paciente inicia sesión enviando su correo y contraseña.

El servidor verifica las credenciales consultando la colección *User* en la base de datos.

Si son correctas, devuelve un *token JWT*, que el navegador guarda para poder autenticar todas las peticiones siguientes.

Esta fase garantiza que solo usuarios válidos puedan reservar una cita.

En la fase de búsqueda de disponibilidad, el paciente utiliza el token para solicitar las especialidades, centros y médicos disponibles.

El servidor primero verifica el token y luego consulta la base de datos para enviar la información actualizada.

El frontend muestra la lista de opciones y el paciente puede ir filtrando médicos según la especialidad seleccionada.

Aquí comienza el proceso interactivo donde el paciente construye su elección.

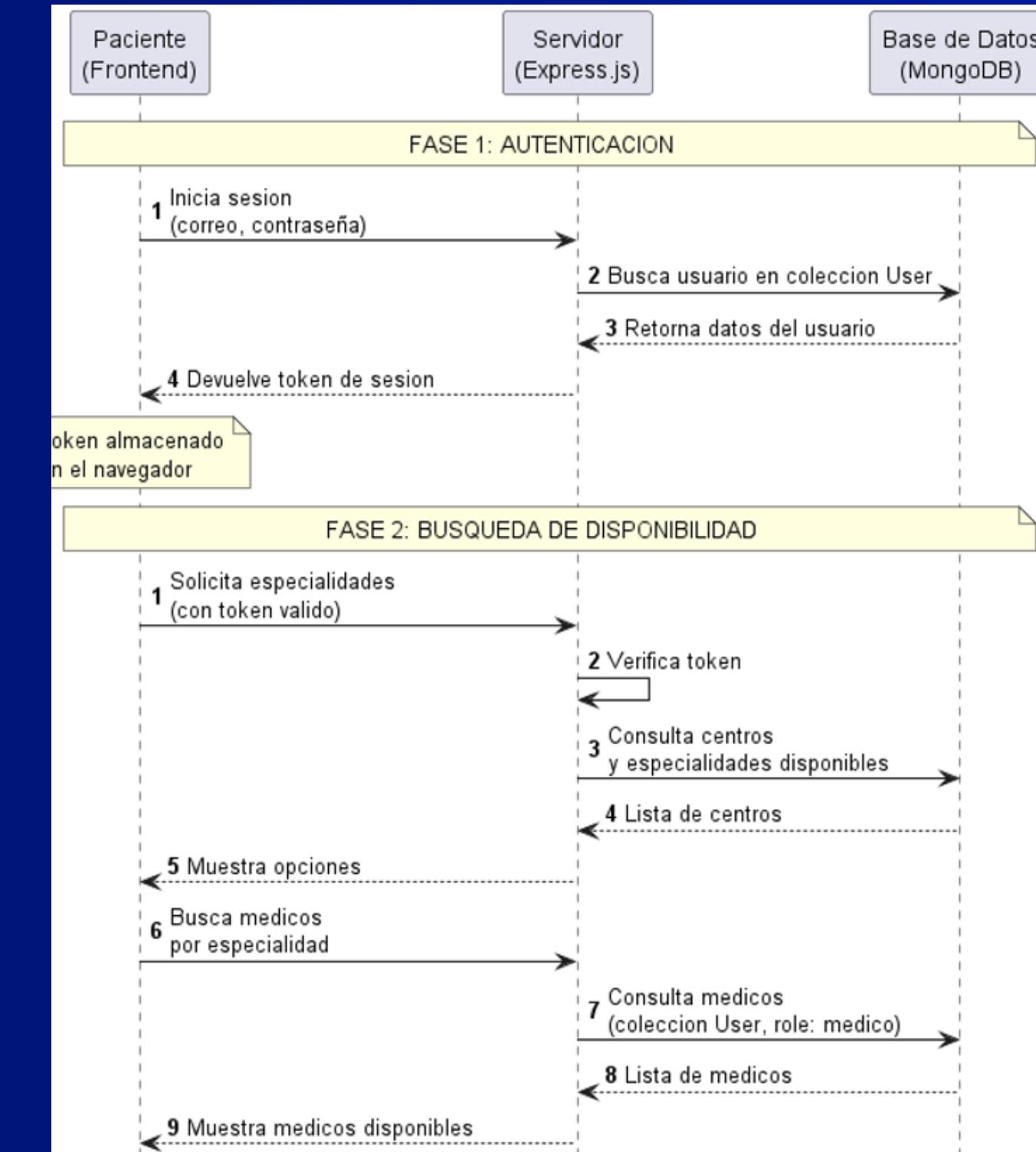




Diagrama de secuencia

Verificación de disponibilidad

En esta fase, el sistema debe validar si realmente hay horarios disponibles para el médico y el centro seleccionados.

El paciente selecciona una fecha, y el servidor consulta todas las citas ya registradas en ese día.

A partir de esa información, calcula automáticamente qué horas siguen libres.

Finalmente, devuelve al paciente solo las horas disponibles para que pueda elegir correctamente.

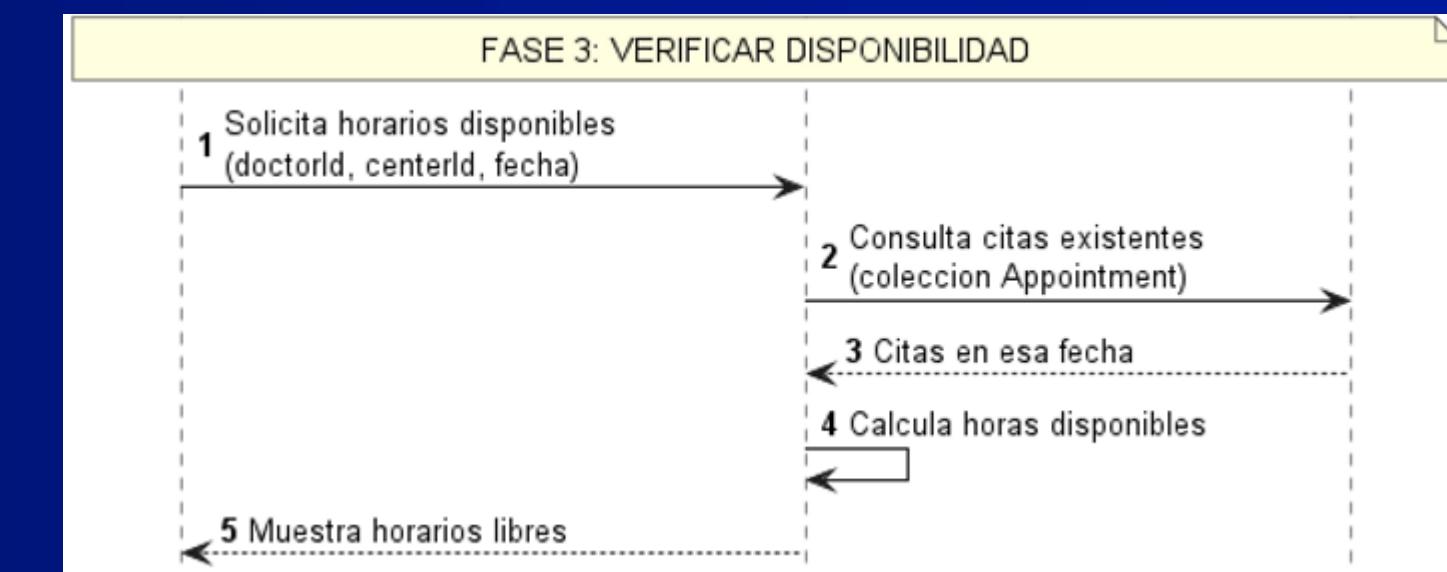


Diagrama de secuencia

Crear cita y procesar pago

Cuando el paciente confirma la reserva, el frontend envía toda la información de la cita: médico, centro, fecha y hora.

El servidor valida los datos y verifica que el médico exista y que la franja horaria siga libre.

Si la cita tiene costo, se calcula el precio final y se procesa el pago en la pasarela.

Si el pago es aprobado, se obtiene un número de transacción que se asocia a la cita.

Si no hay costo, el flujo continúa directamente.

Esta fase asegura que los datos estén completos y que el pago sea confiable.

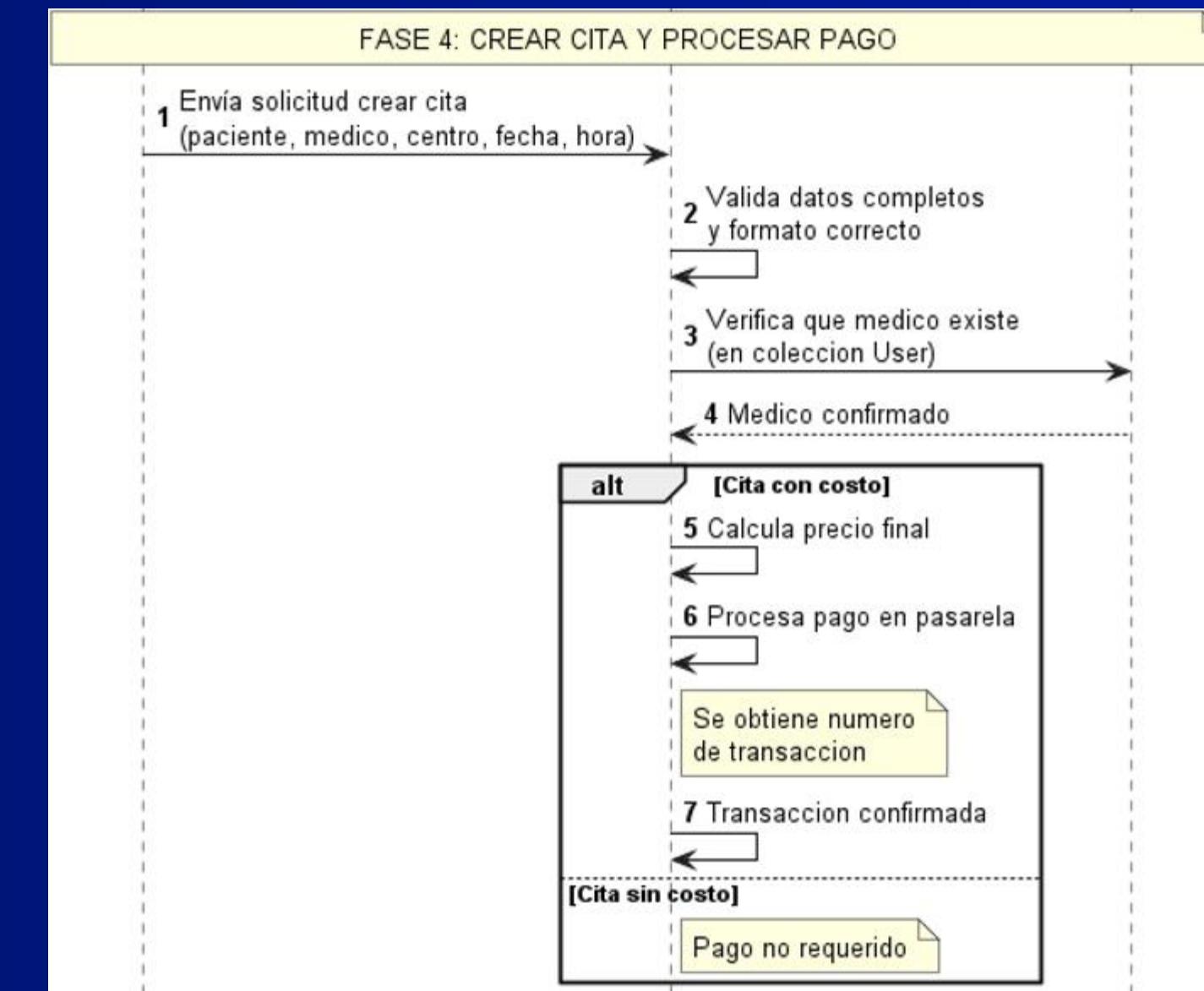




Diagrama de secuencia

Guardar en base de datos

En la última fase, el servidor registra definitivamente la cita en la base de datos creando un documento *Appointment* con todos sus datos.

Después, genera un ID único, prepara las notificaciones y devuelve al frontend una confirmación con el número de cita.

El sistema envía un email al paciente y una notificación al médico.

Finalmente, la pantalla del paciente muestra que la cita ha sido registrada exitosamente.

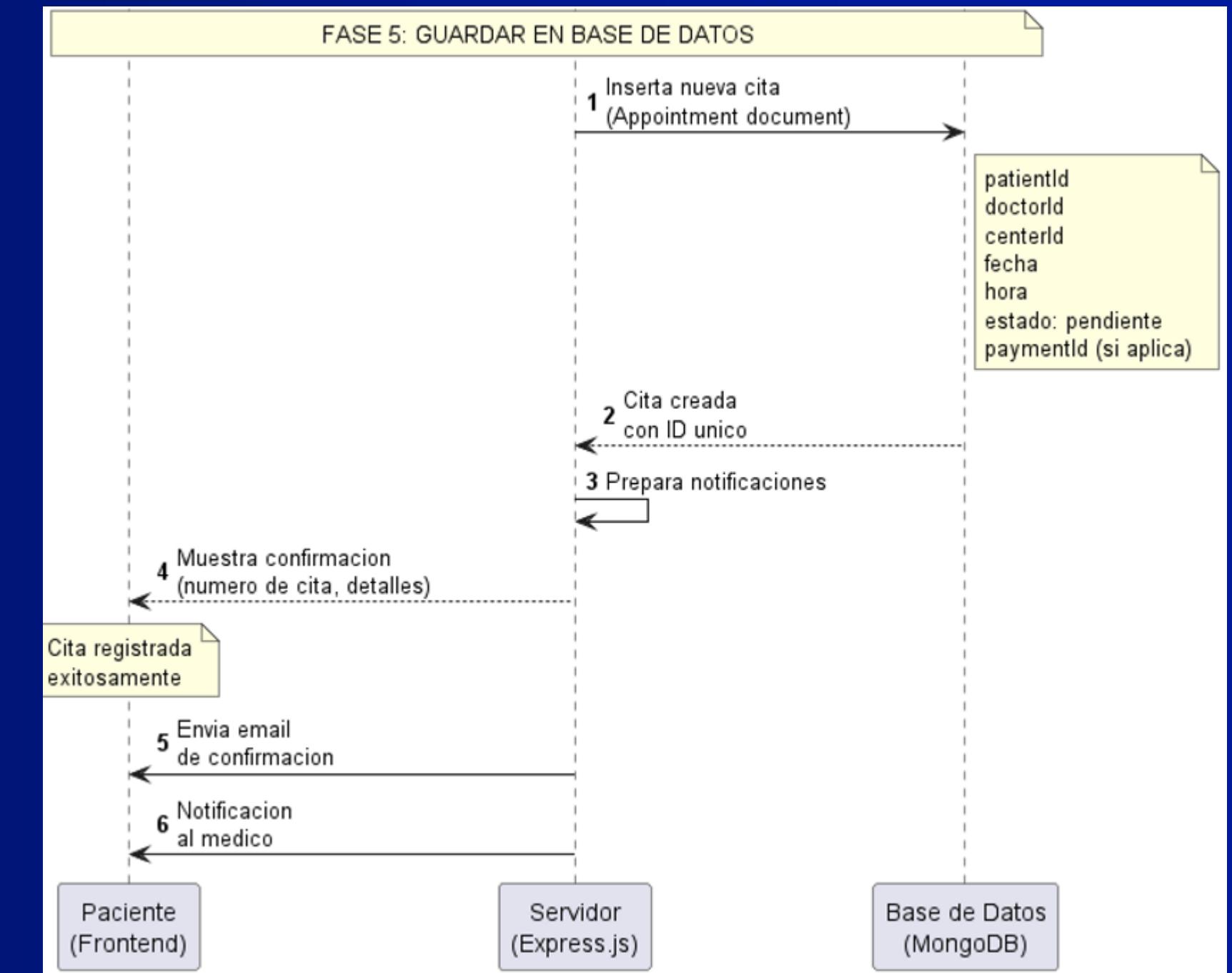
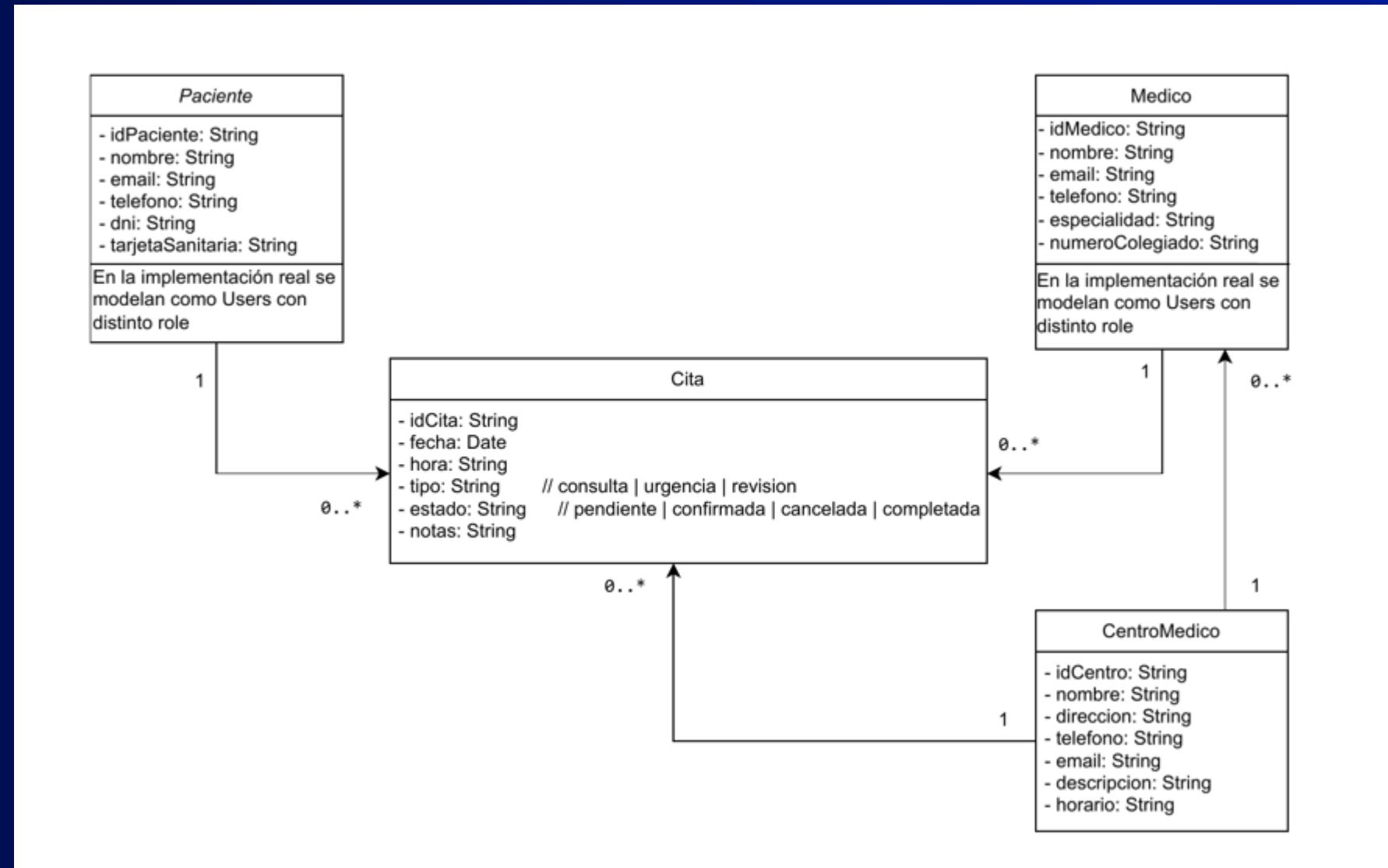




Diagrama de clases



El diagrama de clases representa la estructura principal del sistema.

Incluye las entidades Paciente, Médico, Centro Médico y Cita, junto con sus atributos y relaciones.

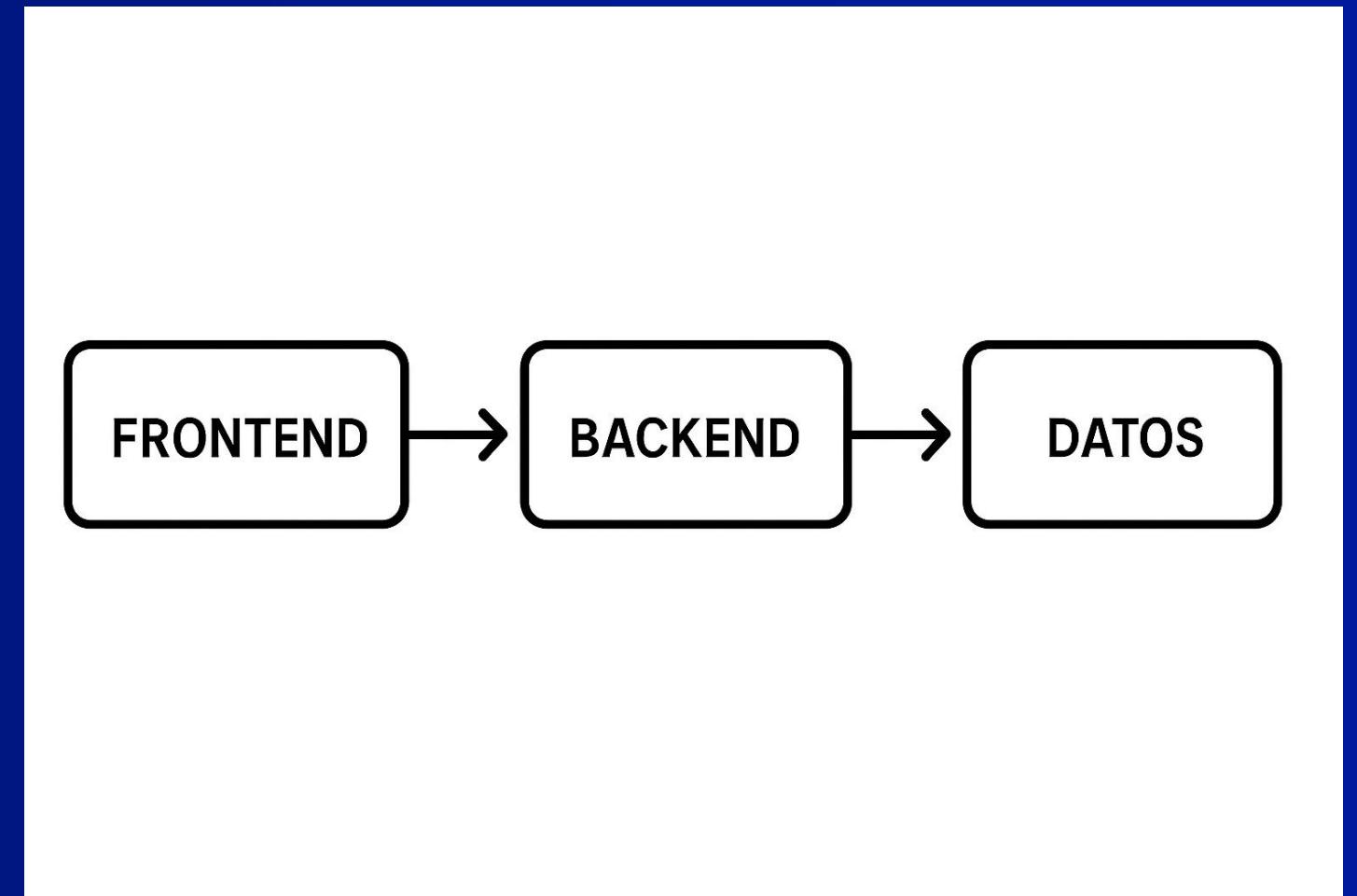
Cada paciente y cada médico pueden tener múltiples citas, y cada cita pertenece a un médico y a un centro médico.

Este modelo define la base lógica para el almacenamiento y la gestión de citas en la plataforma.

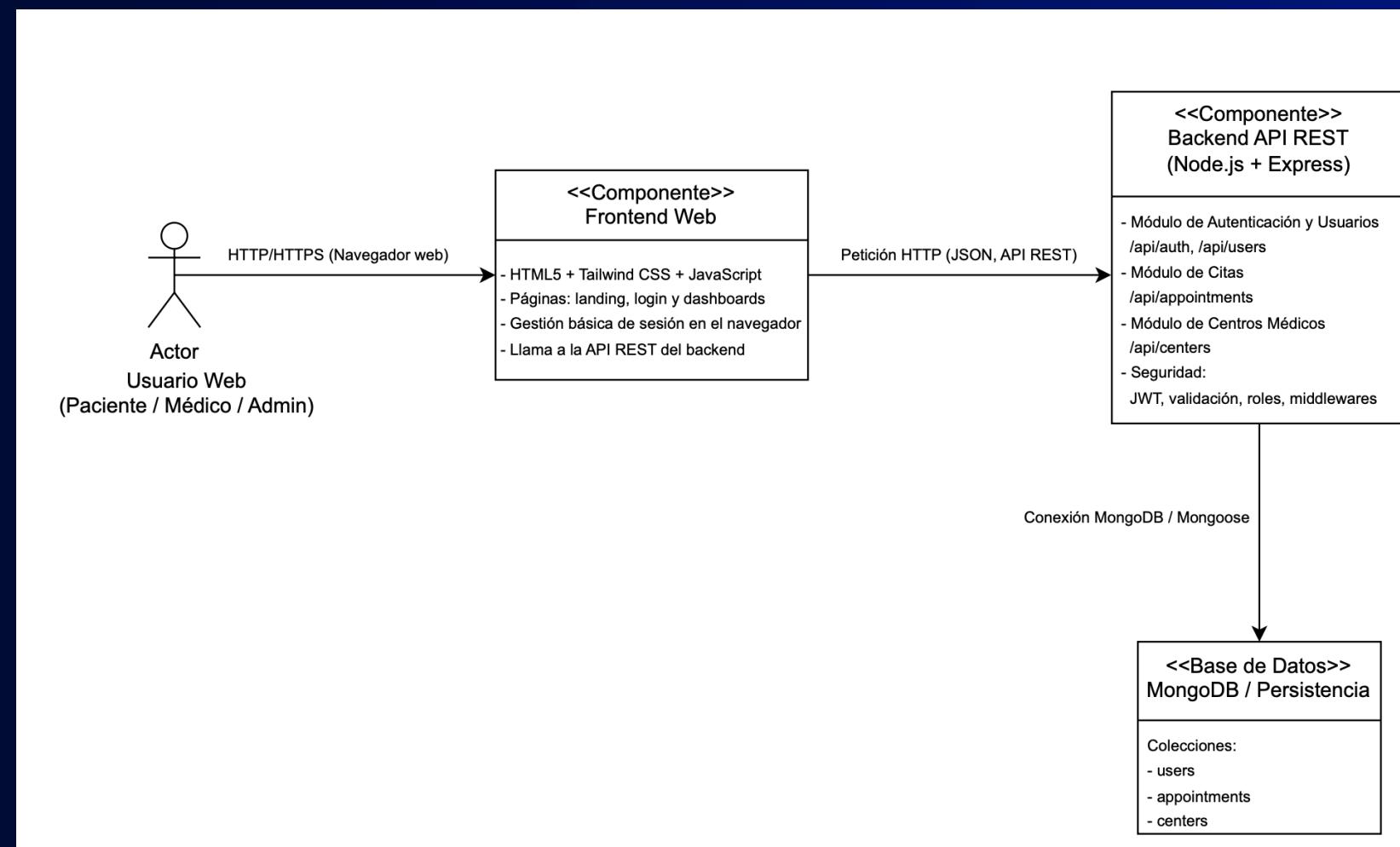


Arquitectura por capas

- Frontend web para experiencia por rol
- Backend API REST con lógica y seguridad
- Persistencia de usuarios, citas y centros
- Comunicación HTTP/JSON



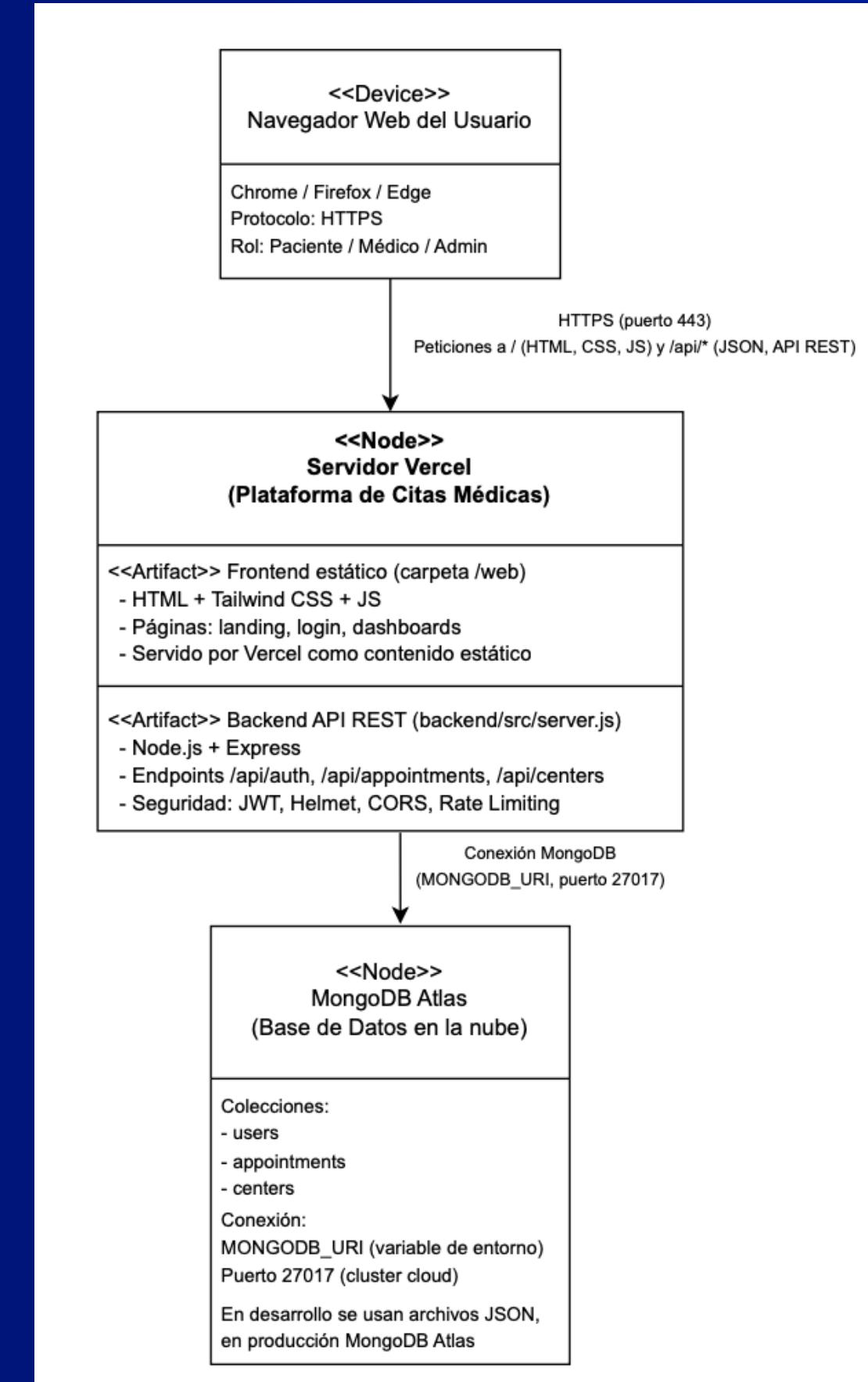
Componentes principales

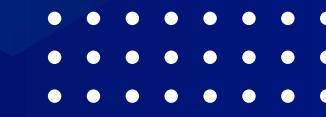


- **Frontend Web**
- **Módulos Backend: Auth/Users, Appointments, Centers**
- **BD con colecciones principales**
- **Más fácil crecer y mantener el sistema**

Despliegue y infraestructura base

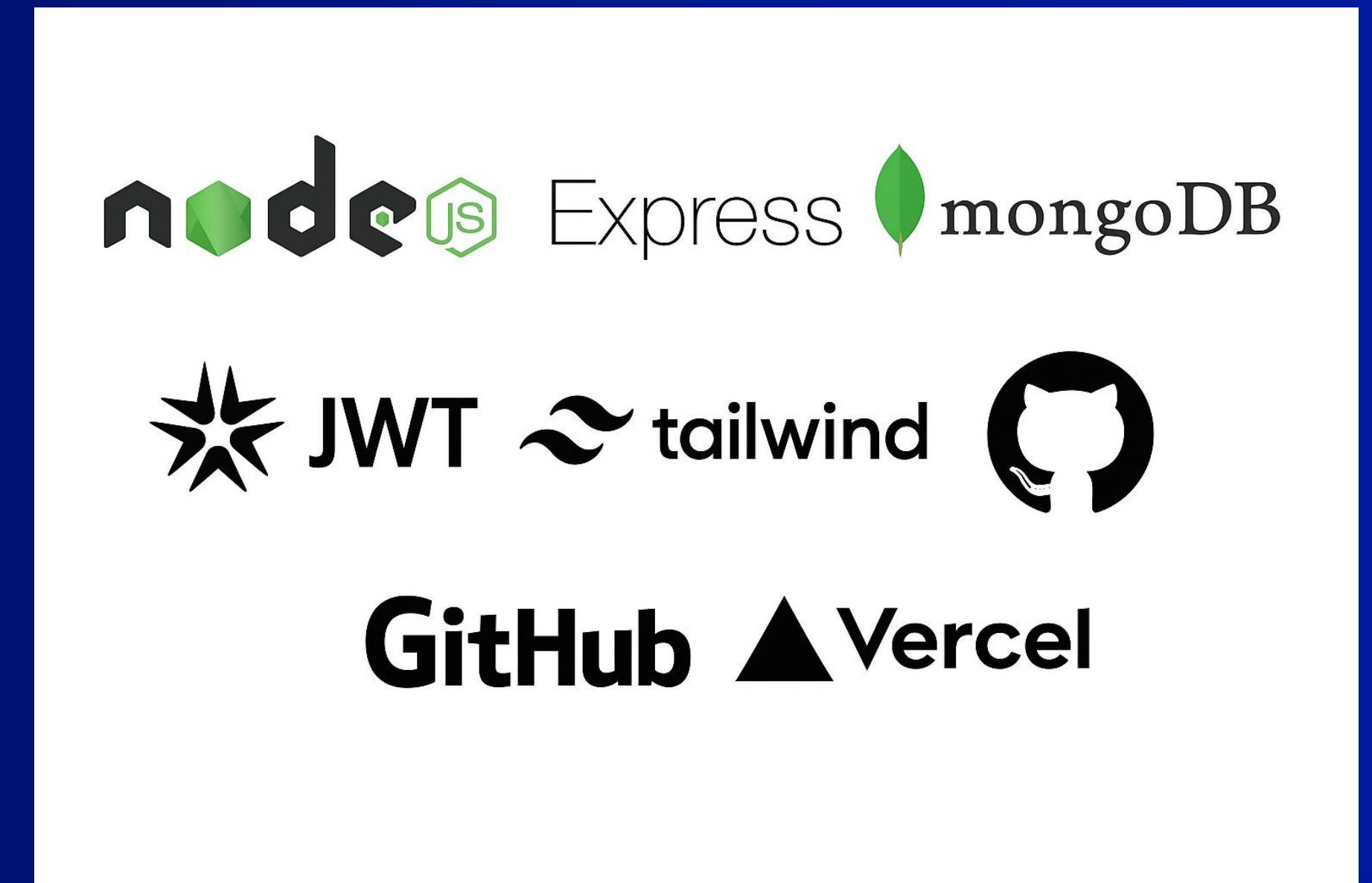
- Vercel sirve frontend estático
- Backend Node/Express expone /api/*
- MongoDB Atlas para producción
- HTTPS + configuración limpia mediante variables de entorno





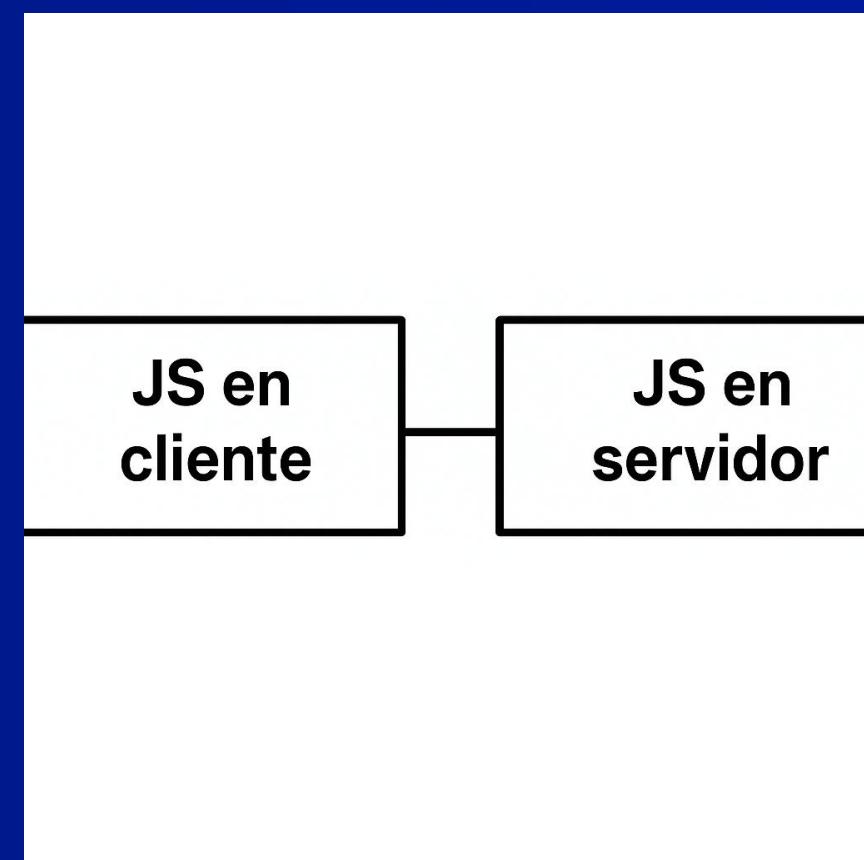
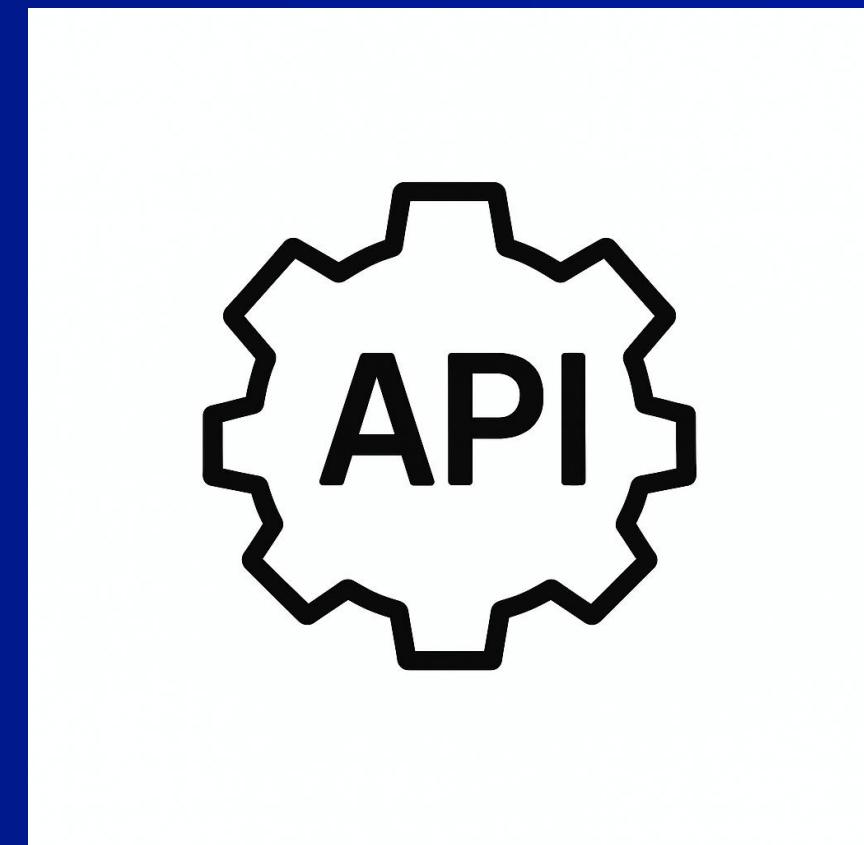
Stack tecnológico elegido

- **Backend:** Node.js + Express
- **BD:** MongoDB (Atlas)
- **Frontend:** HTML + Tailwind + JS
- **Seguridad:** JWT + bcrypt
- **API REST**



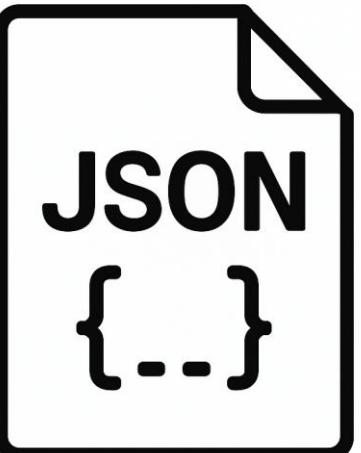
Backend: por qué Node + Express

- Un solo lenguaje en todo el proyecto
- Express es ligero y suficiente para el alcance
- Facilita el trabajo del equipo y el uso de JSON
- Organización clara: rutas, controladores, middlewares
- Integración directa con seguridad



Datos: por qué MongoDB

- **Modelo documental natural para nuestro dominio**
- **Flexibilidad de esquema**
- **Atlas facilita despliegue cloud**
- **Desarrollo simple → producción real**



Colección
users
appointments
centers

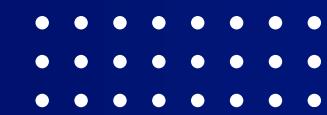
Ejemplos de campos
nombre, email, rol, teléfono
fecha, hora, tipo, estado, notas, idPaciente,
idMedico
nombre, dirección, teléfono, horario

Por qué encaja en JSON
Cada usuario es un objeto independiente
Representa una cita como un documento
con todas sus propiedades
Datos fijos y fácilmente extensibles

Frontend: por qué HTML + Tailwind + JS

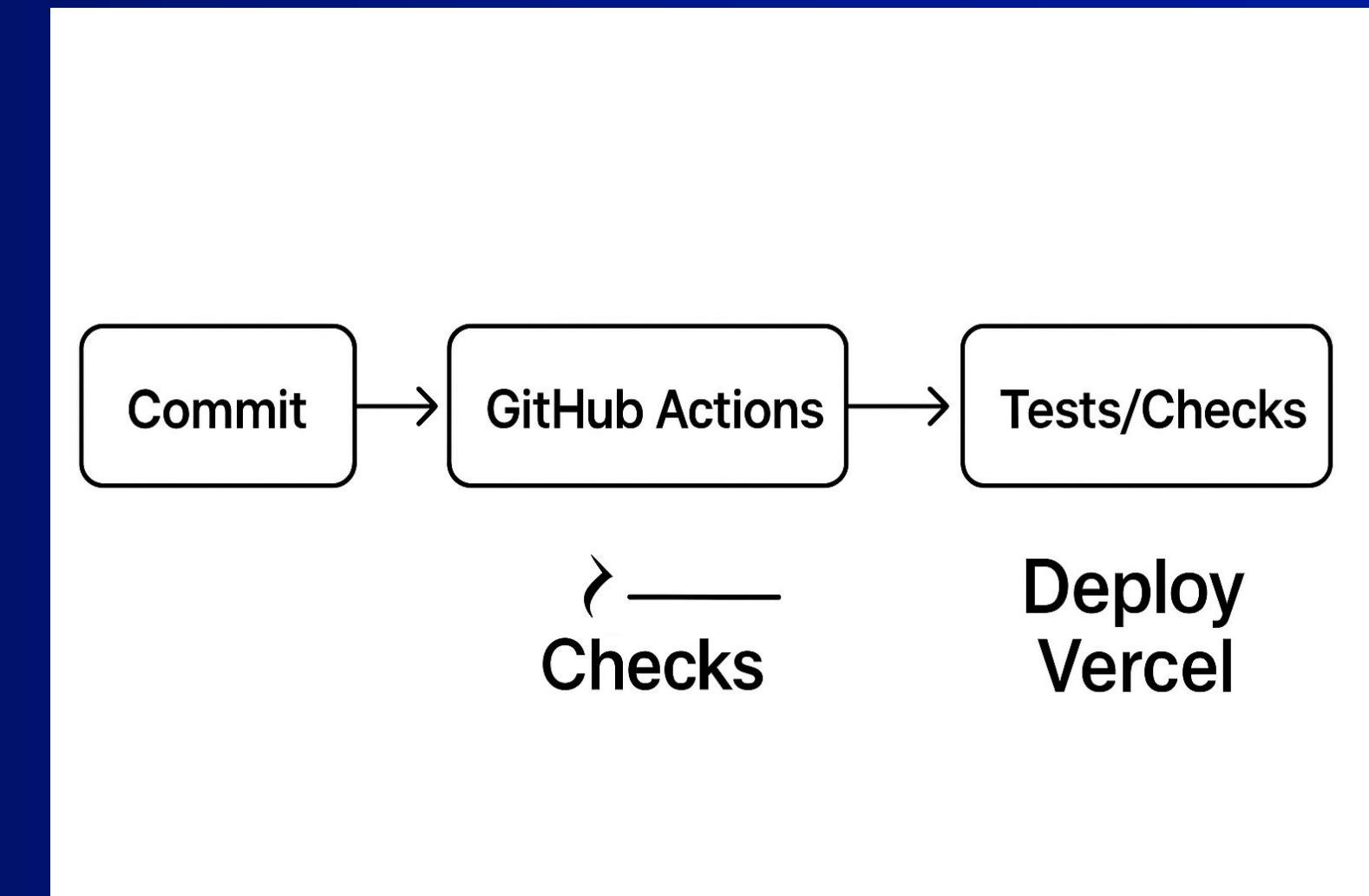
- Más simple que React/Angular para un proyecto académico
- Dashboards rápidos y responsivos
- Menor complejidad de build
- Enfoque en flujos y roles

A screenshot of a medical appointment booking platform. The header includes the logo 'Plataforma de Citas Médicas', navigation links for 'Inicio' (which is underlined), 'Características', 'Cómo funciona', 'Ayuda', and a 'Iniciar sesión' button. The main content features a large call-to-action button with the text 'Reserva tu cita médica más rápido y desde un solo lugar'. Below it, a subtext reads 'Centralizamos la disponibilidad de hospitales y clínicas para que elijas la opción más rápida y cercana.' At the bottom are two buttons: 'Iniciar sesión' and 'Conoce más'. To the right is a circular graphic showing a doctor at a computer monitor displaying a heart rate graph, with medical equipment like a water bottle and a blood bag nearby. A small circular icon with a refresh symbol is in the bottom right corner of the slide.



CI/CD y seguridad en infraestructura

- GitHub como base del flujo del equipo
- Actions para automatizar pruebas y checks
- Vercel para despliegue continuo
- Variables de entorno y HTTPS
- Seguridad alineada con JWT/Helmet/CORS
- Un stack simple, coherente y práctico para una plataforma de citas





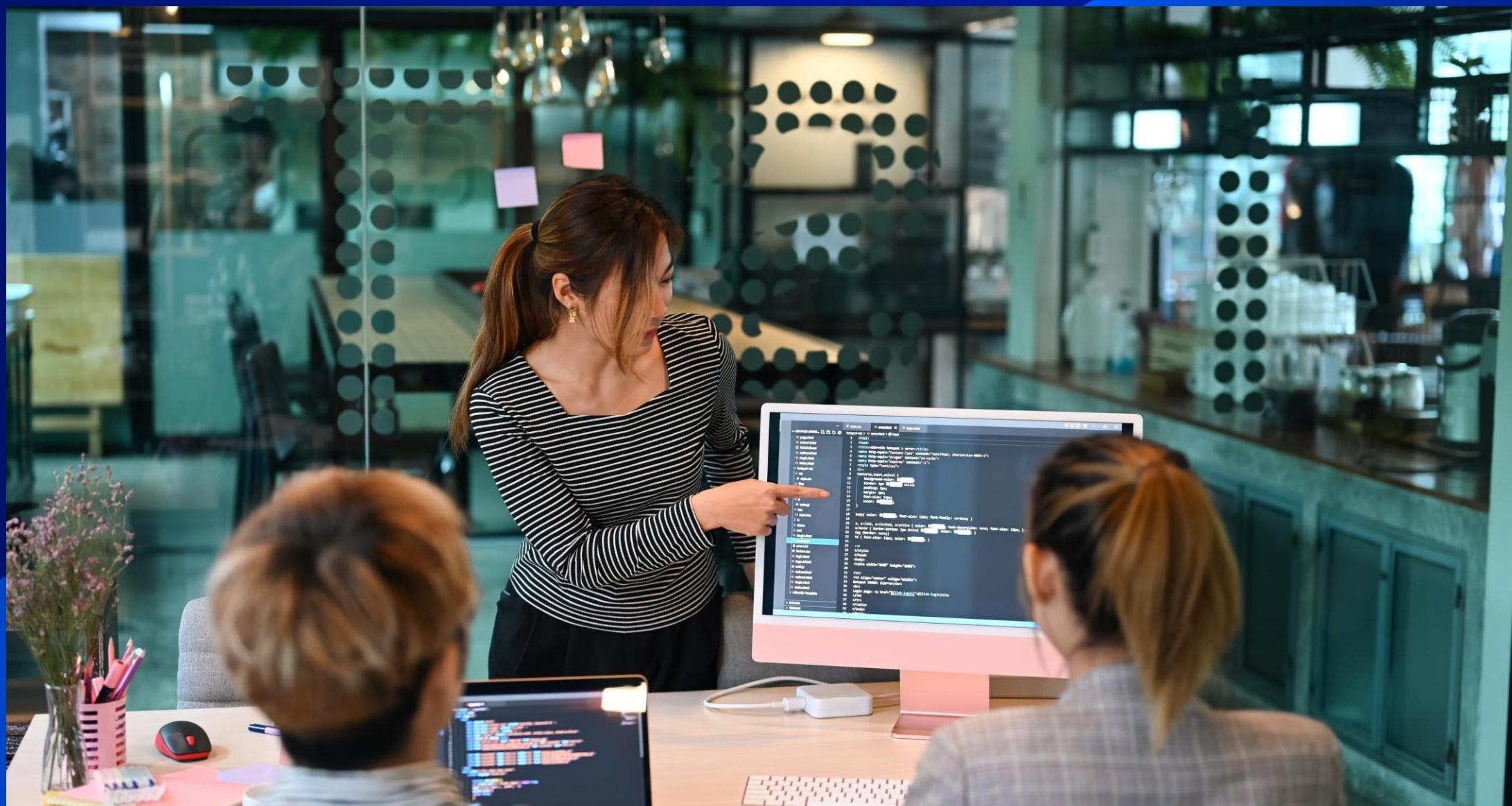
Introducción a Healt-Hub

Conectando Pacientes con Sanitarios

Optimizando la gestión de citas

Mejorando la experiencia en salud

Plataforma integral de gestión de citas médicas que facilita la reserva de consultas, el seguimiento de pacientes y la administración de centros médicos. Sistema desarrollado con arquitectura MVC, autenticación JWT y dashboards personalizados por rol de usuario, brindando una solución eficiente y segura para modernizar la atención médica.



Por qué importa este Sistema de Citas

Esta plataforma de citas médicas nace para digitalizar un proceso que todavía hoy se gestiona muchas veces por teléfono, papel o sistemas aislados. Nuestro objetivo es ofrecer una solución única donde pacientes, médicos y administradores puedan gestionar citas, horarios y disponibilidad de forma sencilla, segura y desde cualquier dispositivo

➔ Resuelve problemas Reales

Evita pérdidas de citas, duplicidades entre centros y falta de información, ofreciendo a cada rol un panel claro con sus tareas y próximas consultas.

➔ Conecta a Personas

Acerca a pacientes, profesionales y centros médicos en una única plataforma, facilitando la comunicación y el acceso a la atención sanitaria.

➔ Mejora la eficiencia

Automatiza la reserva y gestión de citas, reduciendo llamadas telefónicas, solapamientos y tiempos muertos en la agenda de los médicos.





→ Planificar la solución

Definimos el problema de las citas médicas, los roles del sistema y los requisitos funcionales y no funcionales. A partir de ahí diseñamos la arquitectura y los flujos principales, como el proceso completo de reservar una cita.

→ Construir la plataforma

Implementamos la API REST con Node.js, Express y MongoDB, junto con un frontend modular. Añadimos autenticación JWT, gestión de roles y dashboards específicos para pacientes, médicos y administradores.

→ Probar y Mejorar

Aplicamos un plan de pruebas con tests unitarios, de integración y de aceptación. A partir de los resultados corregimos bugs, optimizamos el rendimiento y dejamos preparado el roadmap de mejoras futuras.

Pasos Claves en el Desarrollo

Nuestro proceso de trabajo sigue los principios de la ingeniería del software: primero entendemos el problema, después construimos la solución y, finalmente, la validamos con un plan de pruebas estructurado. Esto nos permite entregar una plataforma de citas médicas fiable, mantenible y lista para evolucionar en siguientes versiones.



El papel del EQUIPO

En este proyecto de Ingeniería del Software no solo hemos implementado una plataforma de citas médicas, sino que también hemos aplicado buenas prácticas de desarrollo, revisión de código y pruebas. Cada miembro del equipo asume responsabilidades concretas para garantizar que el sistema sea funcional, seguro y mantenible a largo plazo.

Definir y documentar el comportamiento

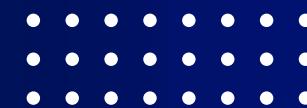
Traducimos los requisitos en historias de usuario, casos de uso y diagramas UML. Esto nos permite acordar qué debe hacer la plataforma antes de programar y revisar que todo lo implementado cumple con las necesidades reales de pacientes, médicos y centros.

Corregir errores y mejorar el código

Aplicamos análisis estático, revisiones de código y pruebas automatizadas para detectar bugs cuanto antes. A partir de cada iteración se refactoriza y se optimiza la lógica de negocio, mejorando la calidad interna del sistema.

Asegurar que el sistema sea fiable

Aplicamos análisis estático, revisiones de código y pruebas automatizadas para detectar bugs cuanto antes. A partir de cada iteración se refactoriza y se optimiza la lógica de negocio, mejorando la calidad interna del sistema.



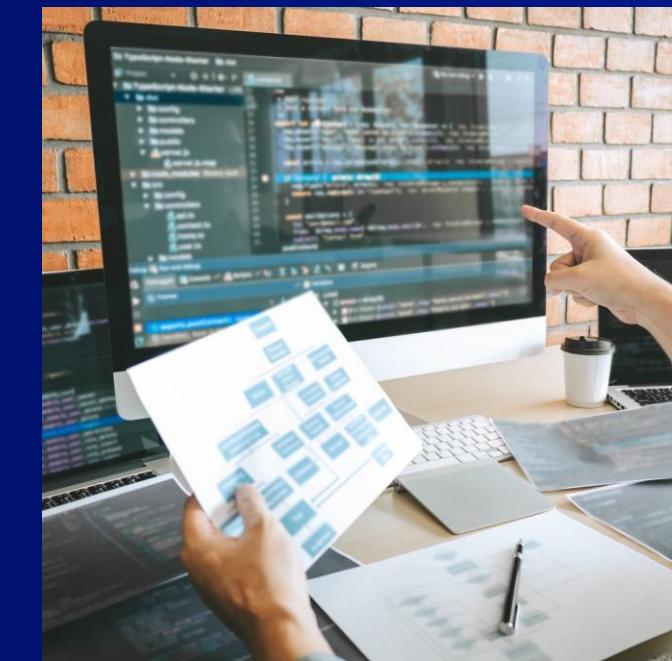
Tipos de SOFTWARE

La plataforma de citas médicas se puede clasificar como software de gestión web, orientado a la administración de procesos clínicos y a la comunicación entre pacientes, profesionales y centros. Se ejecuta en un entorno cliente-servidor y se accede desde el navegador.

Combina varios tipos de software: aplicación web empresarial en el backend, interfaz de usuario interactiva en el frontend y servicios de integración (API REST, base de datos, autenticación). Este enfoque nos permite escalar el sistema y adaptarlo a distintos dispositivos y escenarios de uso.

Beneficios de crear un buen Software

Un buen diseño de software no solo hace que la plataforma de citas médicas funcione, sino que también reduce errores, mejora la experiencia del usuario y facilita el mantenimiento. Aplicar buenas prácticas de ingeniería nos permite ofrecer un sistema fiable y preparado para crecer. Gracias a la planificación, al control de versiones y a un plan de pruebas estructurado, podemos iterar sobre el producto sin romper funcionalidad crítica. Cada versión es más estable que la anterior y corrige incidencias detectadas en entornos de prueba y preproducción.



Ahorra tiempo y esfuerzo

Automatiza la reserva, modificación y cancelación de citas. Disminuye llamadas telefónicas, trabajo manual de recepción y re-planificación de agendas, liberando tiempo tanto a pacientes como a personal sanitario.

Mejora precisión y calidad

Reduce errores humanos en la gestión de horarios, evita solapamientos y garantiza que cada rol vea solo la información que necesita. Las pruebas y validaciones continuas aseguran que los datos de citas y pacientes sean coherentes y estén siempre actualizados.



→ Detectar y corregir errores

Utilizamos pruebas unitarias, de integración y revisión de código para localizar bugs en la lógica de negocio, en los endpoints de la API y en los flujos de usuario. Los errores detectados se registran, se priorizan y se re-prueban tras la corrección

→ Cambios en necesidades y requisitos

Durante el proyecto han surgido nuevos requisitos y ajustes en el flujo de “reservar cita”. Para gestionarlos, actualizamos los modelos y los casos de prueba, evitando regresiones y manteniendo la coherencia entre lo que se pide y lo que realmente hace la plataforma.

Retos en el desarrollo de la Plataforma

Desarrollar una plataforma real de citas médicas implica enfrentarse a errores, cambios de requisitos y limitaciones técnicas. Para gestionarlo, hemos aplicado prácticas de ingeniería del software: iteraciones cortas, revisión de código y un plan de pruebas que nos permite detectar incidencias pronto.

- Cada nueva funcionalidad pasa por fases de diseño, implementación y validación. Cuando aparecen nuevos requisitos o ajustes del profesor/cliente, adaptamos los casos de prueba y repetimos las ejecuciones para asegurar que el sistema sigue siendo estable.

GRACIAS POR LA ATENCIÓN

