

Gestor de Citas Médicas Online - Documentación Técnica Inicial

Objetivo del Proyecto

Desarrollar una aplicación backend para gestionar el sistema de citas médicas entre pacientes y médicos. El sistema permitirá a los pacientes registrarse, reservar citas, ver su historial y recibir notificaciones. Los médicos podrán gestionar su disponibilidad y ver sus próximas consultas.

Módulos funcionales

1. Autenticación y Autorización

- Registro y login para pacientes, médicos y administrador.
- Roles: PACIENTE, MEDICO, ADMIN.
- Seguridad con JWT + Spring Security.
- Endpoints protegidos por roles.

2. Gestión de usuarios

- CRUD de pacientes y médicos.
- Actualización de perfil.
- Habilitar/deshabilitar cuentas por el administrador.

3. Gestión de citas

- Solicitud y gestión de citas.
- Estados: PENDIENTE, CONFIRMADA, RECHAZADA, CANCELADA, COMPLETADA.
- Cancelación con justificación.

4. Historial clínico (opcional)

- Notas clínicas por médico.
- Visualización por el paciente.

5. Notificaciones (opcional)

- Notificación por correo sobre el estado de las citas.

Tecnologías

- Java 17+, Spring Boot
- Spring Security + JWT
- MySQL
- Spring Data JPA + Hibernate
- Swagger / OpenAPI
- Spring Boot Mail
- JUnit 5 + Mockito

Estructura de entidades principales

Usuario (abstracto): id, nombre, email, password, rol

Médico (hereda Usuario): especialidad, número de colegiado, disponibilidad

Paciente (hereda Usuario): edad, género, número de seguro médico

Cita: id, paciente_id, medico_id, fecha_hora, estado, motivo, nota_médica

Seguridad

- Registro/login con JWT
- Hash de contraseñas con BCrypt
- Roles gestionados con anotaciones @PreAuthorize

Endpoints REST (resumen)

- /auth/register (POST)

- /auth/login (POST)
- /usuarios/{id} (GET/PUT/DELETE)
- /medicos (GET)
- /citas (POST)
- /citas/{id} (PUT)
- /citas/paciente/{id} (GET)
- /citas/medico/{id} (GET)

Extras y buenas prácticas

- Uso de DTOs
- Validaciones con javax.validation
- Manejo de excepciones centralizado
- Log de errores