Prueba Técnica – Kettre
Aplicativo – API REST: Sistema de Consulta Médica
Autoría: Joseph Jean Pierre Giraldo Scarpetta

📘 Documentación del API - Sistema de Consulta Médica

Base URL: http://localhost:8080/api

🔹 Paciente

Método	Endpoint	Descripción
GET	/pacientes	Listar todos los pacientes
GET	/pacientes/{id}	Obtener paciente por ID
POST	/pacientes	Crear paciente
PUT	/pacientes/{id}	Actualizar paciente
DELETE	/pacientes/{id}	Eliminar paciente

📌 JSON esperado (POST/PUT):

- nombre: Texto
- documentoIdentidad: Texto
- fechaNacimiento: YYYY-MM-DD
- telefono: Texto
- correoElectronico: Texto
- direccion: Texto

🔹 Medico

Método	Endpoint	Descripción
GET	/medicos	Listar médicos
GET	/medicos/{id}	Obtener médico por ID

POST	/medicos	Crear médico
PUT	/medicos/{id}	Actualizar médico
DELETE	/medicos/{id}	Eliminar médico

📌 JSON esperado (POST/PUT):

- nombre: Texto
- apellido: Texto
- especialidad: Texto
- numeroLicencia: Texto
- telefono: Texto
- correoElectronico: Texto

🔷 Consulta

Método	**Endpoint**	**Descripción**
GET	/consultas	Listar consultas
GET	/consultas/{id}	Obtener consulta por ID
POST	/consultas	Crear consulta
PUT	/consultas/{id}	Actualizar consulta
DELETE	/consultas/{id}	Eliminar consulta

📌 JSON esperado (POST/PUT):

- fecha: YYYY-MM-DD
- hora: HH:mm:ss
- motivoConsulta: Texto
- diagnostico: Texto
- paciente: { idPaciente: ID }
- medico: { idMedico: ID }

Prueba Técnica – Kettre
Aplicativo – API REST: Sistema de Consulta Médica
Autoría: Joseph Jean Pierre Giraldo Scarpetta

◈ Receta

Método	Endpoint	Descripción
GET	/recetas	Listar recetas
GET	/recetas/{id}	Obtener receta por ID
POST	/recetas	Crear receta
PUT	/recetas/{id}	Actualizar receta
DELETE	/recetas/{id}	Eliminar receta

📌 JSON esperado (POST/PUT):

- medicamento: Texto
- dosis: Texto
- instrucciones: Texto
- consulta: { idConsulta: ID }

◈ Usuario

Método	Endpoint	Descripción
GET	/usuarios	Listar usuarios
GET	/usuarios/{id}	Obtener usuario por ID
POST	/usuarios	Crear usuario
PUT	/usuarios/{id}	Actualizar usuario
DELETE	/usuarios/{id}	Eliminar usuario

📌 JSON esperado (POST/PUT):

- username: Texto
- password: Texto
- rol: ADMIN | MEDICO | USUARIO | SOPORTE

Prueba Técnica – Kettre
Aplicativo – API REST: Sistema de Consulta Médica
Autoría: Joseph Jean Pierre Giraldo Scarpetta

□ Documentación Proyecto - Sistema de Consulta Médica

Librería	Descripción
spring-boot-starter-web	Para crear APIs REST (usa Spring MVC + Tomcat)
spring-boot-starter-data-jpa	Para conexión con base de datos vía JPA/Hibernate
spring-boot-starter-actuator	Monitoreo de la aplicación (salud, métricas, etc.)
spring-boot-devtools	Recarga automática en desarrollo
spring-boot-starter-test	Herramientas para testing (JUnit, Mockito, etc.)
spring-boot-maven-plugin	Plugin de compilación y ejecución del proyecto
lombok	para generar getters/setters automáticamente. En tu caso, no lo estás usando activamente.
postgresql	Driver JDBC para conectarse a la base de datos PostgreSQL.

En este proyecto se utilizó una **arquitectura en capas (layered architecture)**, que es uno de los enfoques más comunes en aplicaciones Spring Boot.

Prueba Técnica – Kettre
Aplicativo – API REST: Sistema de Consulta Médica
Autoría: Joseph Jean Pierre Giraldo Scarpetta

Arquitectura en Capas (Layered Architecture)

La aplicación está organizada en **cuatro capas principales**, cada una con responsabilidades bien definidas:

1. Modelo (Model o Entity)

- Paquete: model
- Contiene las clases que representan las entidades del sistema (Paciente, Medico, Consulta, etc.).
- Se mapean directamente a las tablas de la base de datos usando anotaciones JPA (@Entity, @Table, etc.).

2. Repositorio (Repository)

- Paquete: repository
- Interfaces que extienden JpaRepository y se encargan de las operaciones CRUD con la base de datos.
- Ejemplo: PacienteRepository, ConsultaRepository.

3. Servicio (Service)

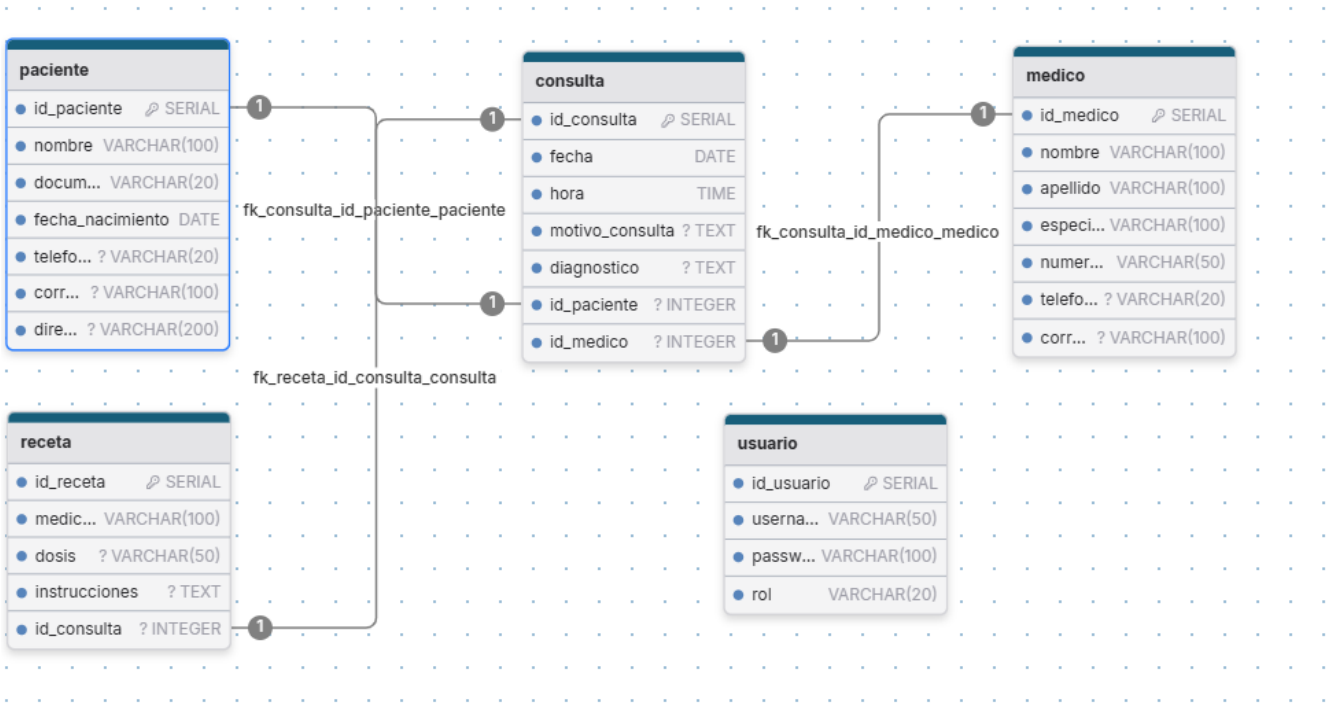
- Paquete: service
- Contiene la lógica de negocio.
- Se comunica con los repositorios y puede incluir validaciones, transformaciones, etc.
- Ejemplo: PacienteService, MedicoService.

4. Controlador (Controller)

- Paquete: controller
- Expone los endpoints REST usando anotaciones como @RestController y @RequestMapping.
- Recibe las solicitudes HTTP y delega la lógica al servicio.

Prueba Técnica – Kettre
Aplicativo – API REST: Sistema de Consulta Médica
Autoría: Joseph Jean Pierre Giraldo Scarpetta

DISEÑO MER



Base de datos: sist_consul_medico

Estimación de Tiempo

	Actividad	Tiempo estimado	Recurso
☒	Diseño MER	2 horas	https://www.drawdb.app/editor?shareId=a168771e2b9058797a8aa78b29e806d4
☒	Configuración del proyecto Spring Boot	1 hora	

Prueba Técnica – Kettre
Aplicativo – API REST: Sistema de Consulta Médica
Autoría: Joseph Jean Pierre Giraldo Scarpetta

☒	Configuración PostgreSQL + conexión	1 hora	
☒	Implementación entidad **Paciente**	3 horas	
☒	Implementación entidad **Médico**	3 horas	
☒	Implementación entidad **Consulta**	4 horas	
☒	Implementación entidad **Receta**	3 horas	
☒	Implementación entidad **Usuario**	2 horas	Hacer uso de Spring Security
☒	Manejo de errores y validaciones básicas	0 horas	No viable por ahora
☒	Pruebas con Postman (todos los servicios)	2 horas	
☒	Documentación básica del API	3 horas	
	Total:	24 horas	