

EXAMEN SPRING BOOT 1ª FINAL ORDINARIA– 2º DAW DWES 07/03/2025

La clínica "San Juan de la Salud" requiere una aplicación backend desarrollada en Spring Boot, para gestionar sus citas médicas, pacientes y médicos, con el objetivo de optimizar la administración de consultas y mejorar la atención a los usuarios.

Parte 1 (Máximo 6.5 puntos): Implementación API REST

(Toma como base **appointments-api** que se adjunta al examen, concretamente **appointments-api-impl**).

Se requiere la creación de una API REST con la siguiente estructura de entidades y funcionalidades. La entidad principal es Cita Médica (Appointment), que está compuesta por un Paciente (Patient) que solicita la consulta y un Médico (Doctor) que la atiende. Recuerda insertar un conjunto de datos realistas para poblar las tablas en MySQL.

Appointment (Cita Médica):

- *id* (Long) – PK-Autoincremental
- *patient* (Patient) Join @ManyToOne – Referencia al paciente
- *doctor* (Doctor) Join @ManyToOne – Referencia al médico.
- *appointmentDate* (LocalDateTime) – Fecha y hora de la cita. "2025-03-07T09:00:00"
- *status* (Enum) – Estado: SCHEDULED, COMPLETED, CANCELLED
- *notes* (String) Nullabe – Observaciones médicas.

Doctor (Médico):

- *id* (Long) – PK- Autoincremental.
- *name* (String) – Nombre del médico
- *specialty* (Enum) – Especialidad: CARDIOLOGY, PEDIATRICS, DERMATOLOGY, NEUROLOGY, etc.
- *dni* (String) – Documento de identidad
- *email* (String) – Correo electrónico

Patient (Paciente):

- *id* (Long) – PK-Autoincremental
- *name* (String) – Nombre del paciente
- *dni* (String) – Documento de identidad
- *email* (String) – Correo electrónico
- *medicalHistory* (String) – Breve historial médico.

Endpoints RestController

Funcionalidad	Entrada	Salida	HTTP code
<i>Insertar una cita médica.</i> <i>El estado inicial es SCHEDULED</i>	RequestCreateAppointmentDTO {patientId, doctorId, appointmentDate}	ResponseEntity <AppointmentDTO>	<ul style="list-style-type: none">▪ 201 CREATED▪ 404 NOT FOUND- No existe el identificador del paciente o doctor.
<i>Obtener el número de citas de un paciente</i>	patientId	ResponseEntity <Integer>	<ul style="list-style-type: none">▪ 200 OK▪ 404 NOT FOUND (si el paciente no existe)

<i>Cambiar el estado de una cita médica.</i> Cuando una cita se encuentre en estado CANCELLED, no se puede cambiar a otro estado.	appointmentId newStatus	ResponseEntity <AppointmentDTO>	<ul style="list-style-type: none"> 200 OK 404 NOT FOUND (si la cita no existe) 400 BADREQUEST – No se puede cambiar el estado de una cita en estado CANCELLED.
Obtener la lista de pacientes atendidos por un médico	doctorId	ResponseEntity <List<PatientDTO>>	<ul style="list-style-type: none"> 200 OK, 404 NOT FOUND (no existe el doctor)

Para estructurar la aplicación, se utilizará una arquitectura por capas (Controller-Service-Repository). Se debe usar entidades (Entities) y DTOs para transferir datos entre las capas. Además, se debe implementar un manejador de excepciones (Exception Handler) para gestionar los errores que surjan en la lógica de negocio.

Parte 2 (Máximo 3.5 puntos): Proceso Batch y consumo de la API REST

(Toma como base **appointments-api**, concretamente **appointments-api-client**, para generar el cliente. Recuerda que puedes recuperar la especificación (Contrato) de la API REST en <http://localhost:8080/v3/api-docs.yaml>. Además de **appointments-batch-tasklet** para el proceso batch).

Se requiere la creación de proceso batch que consumirá la API REST definida en la primera parte mediante un cliente. El batch realizará una petición GET mediante el cliente para obtener la información necesaria para poder guardar el total de citas médicas (ósea count(AppointmentId) de un doctor en una tabla llamada “DoctorStatistics”, ejemplo de la tabla:

id	doctor_id	total_appointments
1	10	2
2	2	1
3	103	2

-
- No se permiten ejercicios prácticos anteriores, se tiene acceso a la documentación o apuntes.
 - Además del código fuente, es recomendable entregar la colección de peticiones de la aplicación con los datos usados o capturas de pantalla de las peticiones y sus datos.
 - **La entrega se realizará en un fichero zip cuyo nombre incluirá vuestro nombre y apellidos.**
 - *Para la nota se tiene en cuenta la arquitectura, funcionalidad implementada y codificación.*