

Desarrollo de sistemas con tecnología Java

Práctica (participación)

Objetivo: En este proyecto, cada grupo deberá desarrollar una API REST utilizando Spring Boot, similar al ejemplo de "librería" presentado en clase, pero enfocándose en una temática diferente. El objetivo es aplicar los conceptos de Spring Boot para construir, desde cero, un sistema que permita gestionar entidades de su elección (como por ejemplo: películas, videojuegos, música, deportes, entre otros). Ustedes decidirán el tema y los atributos de las entidades.

Instrucciones:

Pasos para la Actividad:

1. Formación de Grupos:

- Cada grupo debe trabajar de forma colaborativa a través de Zoom.
- Uno de los participantes se encargará de realizar la codificación, el resto deben colaborar con guía y revisión.

2. Definición de Entidades:

- Como primer paso, deberán ponerse de acuerdo en el tema de su API.
- Una vez elegido el tema, definan al menos **3 atributos** para la entidad principal (pueden agregar más si lo consideran necesario).

3. Implementación del Proyecto:

- **Inicialización del Proyecto Spring Boot:**
 - Usen Spring Initializr (<https://start.spring.io/>) para crear el proyecto base. Incluir las siguientes dependencias:
 - **Spring Web** para construir la API REST.
 - (Opcional) **Spring DevTools** para facilitar el desarrollo.
 - Configuren correctamente las estructuras de paquetes y creen una clase principal que inicie la aplicación.
- **Controlador REST:**
 - Creen un controlador anotado con `@RestController` para gestionar las rutas de su API.
 - Definan las rutas básicas para las operaciones CRUD:
 - **GET:** Obtener todas las entidades y obtener una por su ID.
 - **POST:** Agregar una nueva entidad.
 - **PUT:** Reemplazar completamente una entidad existente por su ID.
 - **PATCH:** Actualizar parcialmente una entidad existente.
 - **DELETE:** Eliminar una entidad por su ID.
- **Simulación de Base de Datos:**

- Utilicen una colección como **HashMap** o **List** para simular el almacenamiento de las entidades.
- Inicialicen algunos datos en el constructor del controlador para facilitar las pruebas.

4. Requerimientos Técnicos:

- **ID Autogenerado:** Al agregar una nueva entidad, deberán implementar una lógica para generar un ID único de manera similar a lo que se vio en el ejemplo de "librería".
- **Validaciones:** Implementen validaciones básicas para los atributos de las entidades (por ejemplo, que el título no sea nulo).
- **Manejo de Errores:** Asegúrense de manejar casos en los que una entidad no se encuentre (ej., devolver un 404 Not Found si se solicita un ID que no existe).

5. End-point de créditos:

- Cada grupo deberá crear al menos un End-Point que devuelva una respuesta en formato **HTML** para desplegar la lista de participantes en el equipo.

6. Pruebas:

- Probar la API con herramientas como **Postman** o **curl** para asegurarse de que todos los métodos funcionan correctamente.
- Asegúrense de probar tanto casos exitosos como fallidos (ej., agregar un nuevo recurso, buscar uno inexistente, eliminar uno).

7. Entrega y Presentación:

- Al finalizar, cada grupo deberá:
 - Subir su proyecto a un repositorio en **GitHub** (con un README.md breve explicando el proyecto).
 - Explicar su proyecto, describiendo las rutas implementadas y cómo funciona la API.