



Programación en la Web

Taller 4

David Zapata - 2222709

Steven Rey - 2221937

Grupo: G2

Escuela de Ingeniería de Sistemas e Informática

Universidad Industrial de Santander

3 de abril de 2025

1. Servicios

1. ¿Cuál es la función principal de un servicio en una aplicación Spring Boot?

Un servicio encapsula la lógica de negocio de la aplicación y proporciona métodos reutilizables para ser utilizados por los controladores o por otros servicios.

2. ¿Cómo se define un servicio en Spring Boot y qué anotación se utiliza?

Un servicio se define como una clase Java con la anotación `@Service`. Ejemplo:

```
1  @Service
2  public class MiServicio {
3      public String obtenerMensaje() {
4          return "Hola desde el servicio";
5      }
6  }
```

3. ¿Cómo se inyectan dependencias en un servicio usando `@Autowired`?

Se usa `@Autowired` para inyectar automáticamente una instancia de un servicio en otra clase:

```
1  @Service
2  public class MiServicio {
3      public String obtenerMensaje() {
4          return "Hola desde el servicio";
5      }
6  }
7
8  @RestController
9  public class MiControlador {
10     private final MiServicio miServicio;
11
12     @Autowired
13     public MiControlador(MiServicio miServicio) {
14         this.miServicio = miServicio;
15     }
16
17     @GetMapping("/mensaje")
18     public String obtenerMensaje() {
19         return miServicio.obtenerMensaje();
20     }
21 }
```

2. Controladores

1. ¿Qué es un controlador en Spring Boot y cuál es su propósito?

Un controlador maneja las solicitudes HTTP y devuelve respuestas. Se define con la anotación `@RestController` o `@Controller`.

2. ¿Cómo se mapean las rutas en un controlador usando `@RequestMapping`?

`@RequestMapping` se usa para definir rutas base o específicas. Ejemplo:

```
1      @RestController
2      @RequestMapping("/api")
3      public class MiControlador {
4          @GetMapping("/saludo")
5          public String saludo() {
6              return "¡Hola desde Spring Boot!";
7          }
8      }
```

3. ¿Cuál es la diferencia entre `@GetMapping`, `@PostMapping`, `@PutMapping` y `@DeleteMapping`?

- `@GetMapping`: Maneja solicitudes GET (obtener datos).
- `@PostMapping`: Maneja solicitudes POST (crear recursos).
- `@PutMapping`: Maneja solicitudes PUT (actualizar recursos).
- `@DeleteMapping`: Maneja solicitudes DELETE (eliminar recursos).

3. Repositorios

1. ¿Qué es un repositorio en Spring Boot y para qué se utiliza?

Un repositorio es una capa de acceso a datos que permite interactuar con la base de datos sin necesidad de escribir consultas SQL.

2. ¿Cómo se crea un repositorio usando `JpaRepository`?

Se crea una interfaz que extiende `JpaRepository`, especificando la entidad y el tipo de su clave primaria:

```
1      @Repository
2      public interface UsuarioRepository extends JpaRepository<Usuario,
3          Long> {
4      }
```

3. ¿Qué métodos básicos proporciona JpaRepository para la manipulación de datos?

- `findAll()`: Obtiene todos los registros.
- `findById(id)`: Busca un registro por ID.
- `save(entidad)`: Guarda o actualiza un registro.
- `deleteById(id)`: Elimina un registro por ID.

4. Entidades

1. ¿Qué es una entidad en el contexto de JPA y Spring Boot?

Es una clase que representa una tabla en la base de datos.

2. ¿Cómo se define una entidad usando la anotación `@Entity`?

```
1      @Entity
2      public class Usuario {
3          @Id
4          @GeneratedValue(strategy = GenerationType.IDENTITY)
5          private Long id;
6          private String nombre;
7      }
```

3. ¿Qué rol juegan las anotaciones `@Id` y `@GeneratedValue` en una entidad?

- `@Id`: Define la clave primaria.
- `@GeneratedValue`: Indica cómo se genera el ID automáticamente.

5. Postman

1. ¿Qué es Postman y cómo se utiliza para probar APIs REST?

Postman es una herramienta para enviar solicitudes HTTP y probar APIs.

2. ¿Cómo se crea una nueva solicitud en Postman?

- Abrir Postman.
- Hacer clic en "New Request".
- Elegir el método (GET, POST, etc.).

- Especificar la URL de la API.
 - Enviar la solicitud.
3. ¿Cómo se configuran y envían solicitudes GET, POST, PUT y DELETE en Postman?
- **GET**: Introducir la URL y hacer clic en "Send".
 - **POST**: Seleccionar "Body", elegir raw", escribir JSON y enviar.
 - **PUT**: Igual que POST pero con datos para actualizar.
 - **DELETE**: Solo se introduce la URL y se envía.

6. Bases de datos H2

1. ¿Qué es H2 y por qué es útil en el desarrollo con Spring Boot?
- H2 es una base de datos en memoria usada para pruebas rápidas sin necesidad de configurar una base de datos real.
2. ¿Cómo se configura una base de datos H2 en el archivo `application.properties`?

```
1      spring.datasource.url=jdbc:h2:mem:testdb
2      spring.datasource.driverClassName=org.h2.Driver
3      spring.datasource.username=sa
4      spring.datasource.password=
5      spring.jpa.database-platform=org.hibernate.dialect.H2Dialect
6      spring.h2.console.enabled=true
```

3. ¿Cómo se accede a la consola H2 y qué se puede hacer desde allí?
- Ir a `http://localhost:8080/h2-console`.
 - Ingresar la URL de la base de datos (`jdbc:h2:mem:testdb`).
 - Ejecutar consultas SQL en la interfaz gráfica.

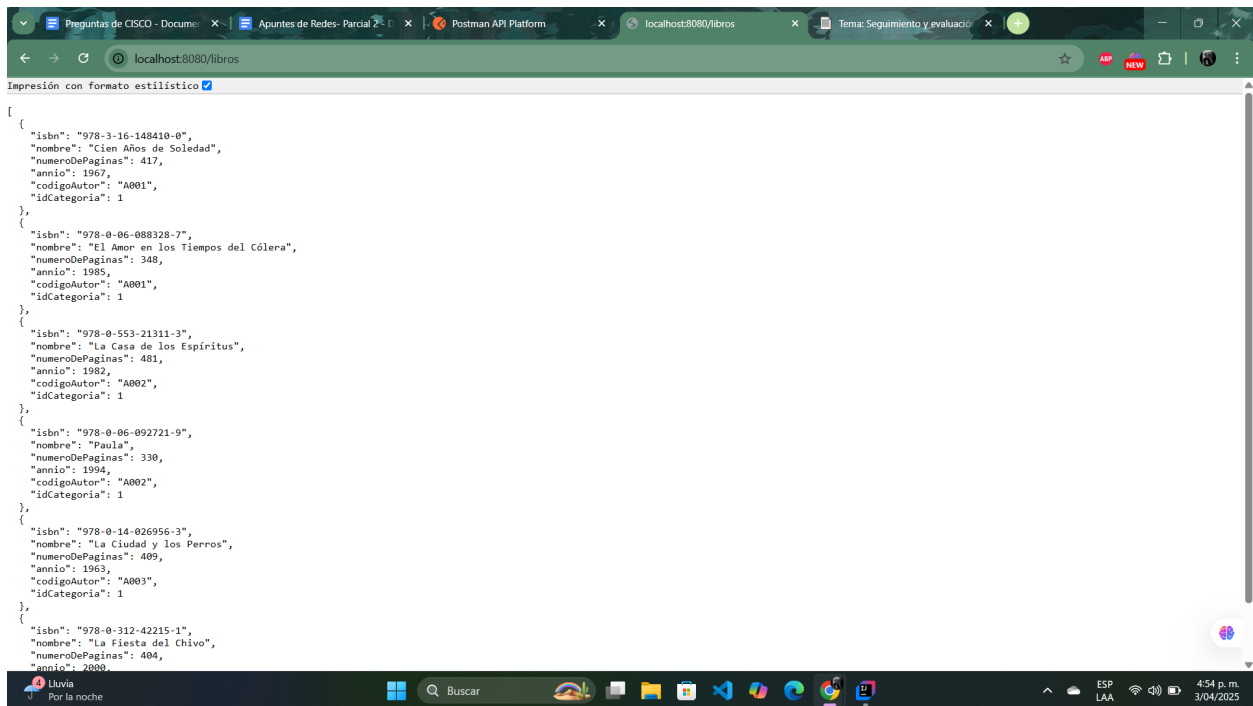


Figura 1: Ejecución