

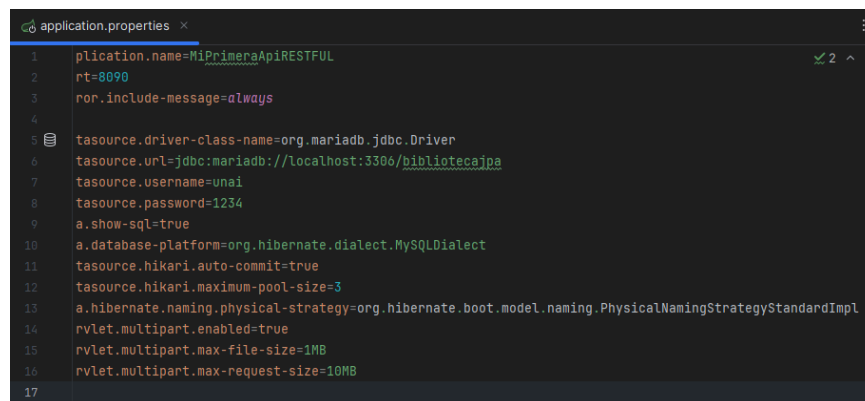
Mi primera API RESTFul con SpringBoot

Esta es mi primera API RESTFul con Spring Boot y he utilizado una base de datos de biblioteca, con las entidades libró, usuario, ejemplar y préstamo.

Primero mostraré la configuración que tengo en la API, en la application.properties tengo la configuración aportada por el profesor, solo le he añadido la linea

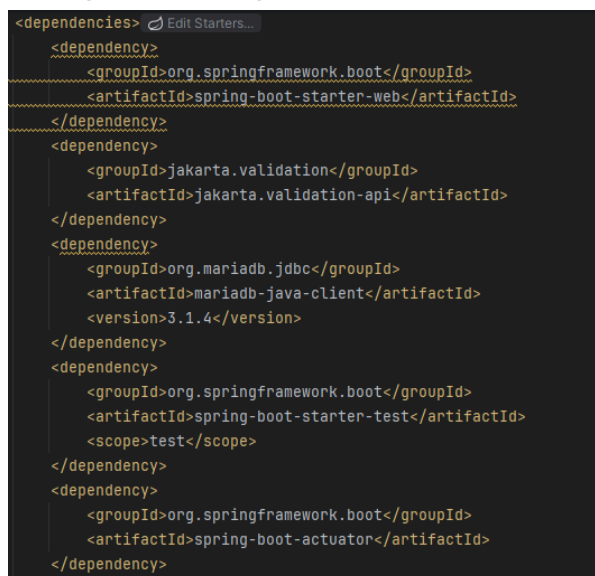
```
pring.jpa.hibernate.naming.physical-strategy=org.hibernate.boot.model.naming.PhysicalNamingStrategyStandardImpl
```

para que no den problemas los nombres del programa con la base de datos.

A screenshot of a code editor showing the 'application.properties' file. The file contains various configuration properties for a Spring Boot application, including application name, port, logging, database connection details (driver, URL, username, password), Hibernate settings (show-sql, dialect, naming strategy), and multipart settings. The naming strategy property is highlighted in green.

```
1 application.name=MiPrimeraApiRESTFUL
2 port=8090
3 logging.level.org.springframework.web=DEBUG
4 logging.level.org.springframework.web.servlet.mvc.annotation.support=DEBUG
5 logging.level.org.springframework.web.servlet.mvc.annotation.support=DEBUG
6 logging.level.org.springframework.web.servlet.mvc.annotation.support=DEBUG
7 logging.level.org.springframework.web.servlet.mvc.annotation.support=DEBUG
8 logging.level.org.springframework.web.servlet.mvc.annotation.support=DEBUG
9 logging.level.org.springframework.web.servlet.mvc.annotation.support=DEBUG
10 logging.level.org.springframework.web.servlet.mvc.annotation.support=DEBUG
11 logging.level.org.springframework.web.servlet.mvc.annotation.support=DEBUG
12 logging.level.org.springframework.web.servlet.mvc.annotation.support=DEBUG
13 logging.level.org.springframework.web.servlet.mvc.annotation.support=DEBUG
14 logging.level.org.springframework.web.servlet.mvc.annotation.support=DEBUG
15 logging.level.org.springframework.web.servlet.mvc.annotation.support=DEBUG
16 logging.level.org.springframework.web.servlet.mvc.annotation.support=DEBUG
17 logging.level.org.springframework.web.servlet.mvc.annotation.support=DEBUG
```

En el pom también tengo la configuración aportada por el profesor.

A screenshot of a code editor showing the 'pom.xml' file. The file contains XML dependencies for the Spring Boot application, including spring-boot-starter-web, jakarta.validation-api, mariadb-java-client, spring-boot-starter-test, and spring-boot-actuator. The dependencies are listed within a <dependencies> tag.

```
<dependencies>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
  </dependency>
  <dependency>
    <groupId>jakarta.validation</groupId>
    <artifactId>jakarta.validation-api</artifactId>
  </dependency>
  <dependency>
    <groupId>org.mariadb.jdbc</groupId>
    <artifactId>mariadb-java-client</artifactId>
    <version>3.1.4</version>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-test</artifactId>
    <scope>test</scope>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-actuator</artifactId>
  </dependency>
</dependencies>
```

```

<!-- Starter de Spring Data JPA -->
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-data-jpa</artifactId>
</dependency>
<dependency>
  <groupId>org.mariadb.jdbc</groupId>
  <artifactId>mariadb-java-client</artifactId>
  <version>3.1.4</version>
</dependency>
<dependency>
  <groupId>com.fasterxml.jackson.core</groupId>
  <artifactId>jackson-databind</artifactId>
</dependency>
<!-- Starter de Spring Cache -->
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-cache</artifactId>
</dependency>
<dependency>
  <groupId>org.projectlombok</groupId>
  <artifactId>lombok</artifactId>
  <version>1.18.20</version>
  <scope>provided</scope>
</dependency>

```

Ahora vamos con el código, empezaremos creando los DAOs, ya que serán la base, ya que son las clases que utilizaremos para mover los datos. Serán cuatro DAOs Ejemplar, Libro, Préstamo y Usuario.

```

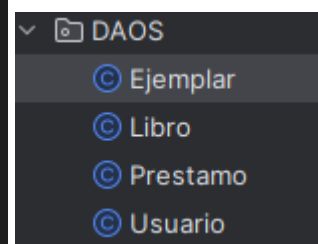
@Entity 17 usages ± Unai *
@Table(name = "ejemplar")
public class Ejemplar {
    // Al generar un ejemplar se le asignará un ID
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "id", nullable = false)
    private Integer id;

    // Es una clase la cual contiene un objeto que es otra clase, por ello tiene relación ManyToOne
    // y tiene un onDelete tipo cascade que si se elimina el libro que contiene también se borre
    // este objeto
    @NotNull 4 usages
    @ManyToOne(fetch = FetchType.LAZY, optional = false)
    @OnDelete(action = OnDeleteAction.CASCADE)
    @JoinColumn(name = "isbn", nullable = false)
    @JsonIgnoreProperties({"hibernateLazyInitializer", "handler"})
    private org.example.miprimeraapirestful.DAOS.Libro isbn;

    public Ejemplar(Libro isbn) { no usages ± Unai *
        this.isbn = isbn;
        this.estado = "Disponible";
    }

    public Ejemplar() { ± Unai *
    }
}

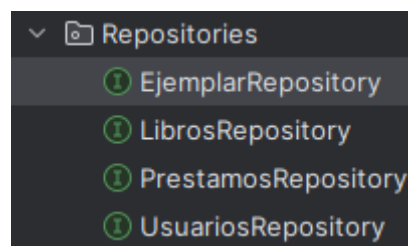
```



Las clases repository son interfaces las cuales instanciamos en las clases controller, estás hacen que podamos utilizar las funciones para poder manejar los datos de la base de datos, ya que vienen por defecto.

```
package org.example.miprimeraapiresful.Repositories;

> import ...
    ⚡
} public interface EjemplarRepository extends JpaRepository<Ejemplar, Integer> {
}
```



Ahora vamos con la clase que utiliza las clases anteriores para realizar las operaciones que pida el usuario, al principio de la clase, justo antes de declararla le ponemos `@RestController` para declarar que es un controlador y un `@RequestMapping` para darle la dirección. Dentro de la clase tenemos el `EjemplarRepository`, junto con dos Constructores, uno vacío y el otro para iniciar el `EjemplarRepository`, ahora creamos los `getEjemplares`, el primero que devuelva todos y el segundo para que devuelva uno en concreto metiéndole el ID.

```

@RestController  Unai
@RequestMapping("/ejemplar")
public class EjemplarControllerMOCK {
    EjemplarRepository ejemplarRepository; 6 usages

    @Autowired  Unai
    public EjemplarControllerMOCK(EjemplarRepository ejemplarRepository) {
        this.ejemplarRepository = ejemplarRepository;
    }
    public EjemplarControllerMOCK(){  Unai
    }

    @GetMapping  Unai
    public ResponseEntity<List<Ejemplar>> getEjemplares() {
        return ResponseEntity.ok(this.ejemplarRepository.findAll());
    }

    @GetMapping("/{id}")  Unai
    public ResponseEntity<Ejemplar> getEjemplarsJSON(@PathVariable int id) {
        Ejemplar e = this.ejemplarRepository.findById(id).get();
        return ResponseEntity.ok(e);
    }
}

```

Los métodos add, update y delete son los que hará los cambios en la base de datos, con add haremos un @PostMapping pidiendo al usuario un JSON de el Ejemplar que quiera añadir. Con update utilizamos @PutMapping, y lo mismo metemos en forma de JSON el ejemplar a actualizar con el ID del ejemplar existente. Por último el delete, el cual solo nos pedirá el ID del ejemplar a eliminar.

```

@PostMapping  Unai
public ResponseEntity<Ejemplar> addEjemplar(@RequestBody Ejemplar ejemplar) {
    Ejemplar e = this.ejemplarRepository.save(ejemplar);
    return ResponseEntity.ok().body(e);
}

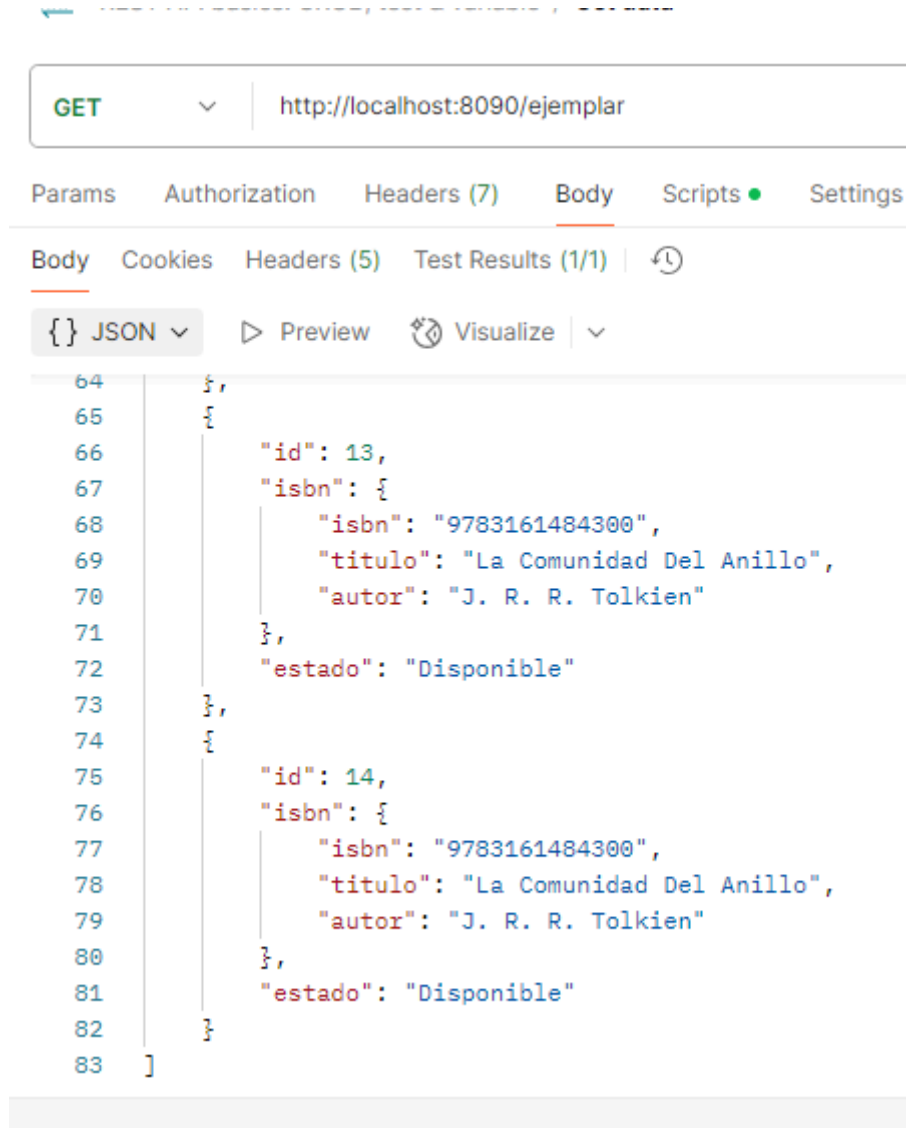
@PutMapping  Unai
public ResponseEntity<Ejemplar> updateEjemplar(@RequestBody Ejemplar ejemplar) {
    Ejemplar e = this.ejemplarRepository.save(ejemplar);
    return ResponseEntity.ok().body(e);
}

@DeleteMapping("/{id}")  Unai
public ResponseEntity<String> deleteEjemplar(@PathVariable int id) {
    this.ejemplarRepository.deleteById(id);
    String mensaje = "Ejemplar con ID " + id + " eliminado";
    return ResponseEntity.ok().body(mensaje);
}

```

Por último voy a hacer la comprobación de la API con PostMan, voy a utilizar solo la de ejemplar ya que el resto son iguales.

Método get



Método add

POST

http://localhost:8090/ejemplar

ParamsAuthorizationHeaders (9)Body ●Scripts ●Settings

none

form-data

x-www-form-urlencoded

raw

binary

Gr

```
1 {
2   "isbn": {
3     "isbn": "9783161484300",
4     "titulo": "La Comunidad Del Anillo",
5     "autor": "J. R. R. Tolkien"
6   },
7   "estado": "Disponible"
8 }
```

BodyCookiesHeaders (5)Test Results (1/1)🔄































{ } JSON

▶ Preview

🔗 Visualize

▼

```
1 {
2   "id": 17,
3   "isbn": {
4     "isbn": "9783161484300",
5     "titulo": "La Comunidad Del Anillo",
6     "autor": "J. R. R. Tolkien"
7   },
8   "estado": "Disponible"
9 }
```

<input type="checkbox"/>		Editar		Copiar		Borrar	4	9789876543210	Prestado
<input type="checkbox"/>		Editar		Copiar		Borrar	5	9789876543210	Dañado
<input type="checkbox"/>		Editar		Copiar		Borrar	6	9781111111111	Prestado
<input type="checkbox"/>		Editar		Copiar		Borrar	7	9781111111111	Disponible
<input type="checkbox"/>		Editar		Copiar		Borrar	8	9783161484100	Disponible
<input type="checkbox"/>		Editar		Copiar		Borrar	11	9780306406157	Prestado
<input type="checkbox"/>		Editar		Copiar		Borrar	12	9783161484300	Disponible
<input type="checkbox"/>		Editar		Copiar		Borrar	13	9783161484300	Disponible
<input type="checkbox"/>		Editar		Copiar		Borrar	14	9783161484300	Disponible
<input type="checkbox"/>		Editar		Copiar		Borrar	17	9783161484300	Disponible

Método update

PUT

http://localhost:8090/ejemplar

ParamsAuthorizationHeaders (9)Body ●Scripts ●Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

```
1 {
2   "id": 17,
3   "isbn": {
4     "isbn": "9783161484300",
5     "titulo": "La Comunidad Del Anillo",
6     "autor": "J. R. R. Tolkien"
7   },
8   "estado": "Dañado"
9 }
```





BodyCookiesHeaders (5)Test Results (1/1)

{ } JSON

Preview

Visualize

```
1 {
2   "id": 17,
3   "isbn": {
4     "isbn": "9783161484300",
5     "titulo": "La Comunidad Del Anillo",
6     "autor": "J. R. R. Tolkien"
7   },
8   "estado": "Dañado"
9 }
```

<input type="checkbox"/>		Editar		Copiar		Borrar	4	9789876543210	Prestado
<input type="checkbox"/>		Editar		Copiar		Borrar	5	9789876543210	Dañado
<input type="checkbox"/>		Editar		Copiar		Borrar	6	9781111111111	Prestado
<input type="checkbox"/>		Editar		Copiar		Borrar	7	9781111111111	Disponible
<input type="checkbox"/>		Editar		Copiar		Borrar	8	9783161484100	Disponible
<input type="checkbox"/>		Editar		Copiar		Borrar	11	9780306406157	Prestado
<input type="checkbox"/>		Editar		Copiar		Borrar	12	9783161484300	Disponible
<input type="checkbox"/>		Editar		Copiar		Borrar	13	9783161484300	Disponible
<input type="checkbox"/>		Editar		Copiar		Borrar	14	9783161484300	Disponible
<input type="checkbox"/>		Editar		Copiar		Borrar	17	9783161484300	<u>Dañado</u>

Método delete

DELETE

http://localhost:8090/ejemplar/17

ParamsAuthorizationHeaders (7)BodyScripts ●Settings

none

form-data

x-www-form-urlencoded

raw




























binary

1

BodyCookiesHeaders (5)Test Results (1/1)

RawPreviewVisualize

1Ejemplar con ID 17 eliminado

					id	isbn	estado
<input type="checkbox"/>		Editar		Copiar		Borrar	4 9789876543210 Prestado
<input type="checkbox"/>		Editar		Copiar		Borrar	5 9789876543210 Dañado
<input type="checkbox"/>		Editar		Copiar		Borrar	6 9781111111111 Prestado
<input type="checkbox"/>		Editar		Copiar		Borrar	7 9781111111111 Disponible
<input type="checkbox"/>		Editar		Copiar		Borrar	8 9783161484100 Disponible
<input type="checkbox"/>		Editar		Copiar		Borrar	11 9780306406157 Prestado
<input type="checkbox"/>		Editar		Copiar		Borrar	12 9783161484300 Disponible
<input type="checkbox"/>		Editar		Copiar		Borrar	13 9783161484300 Disponible
<input type="checkbox"/>		Editar		Copiar		Borrar	14 9783161484300 Disponible