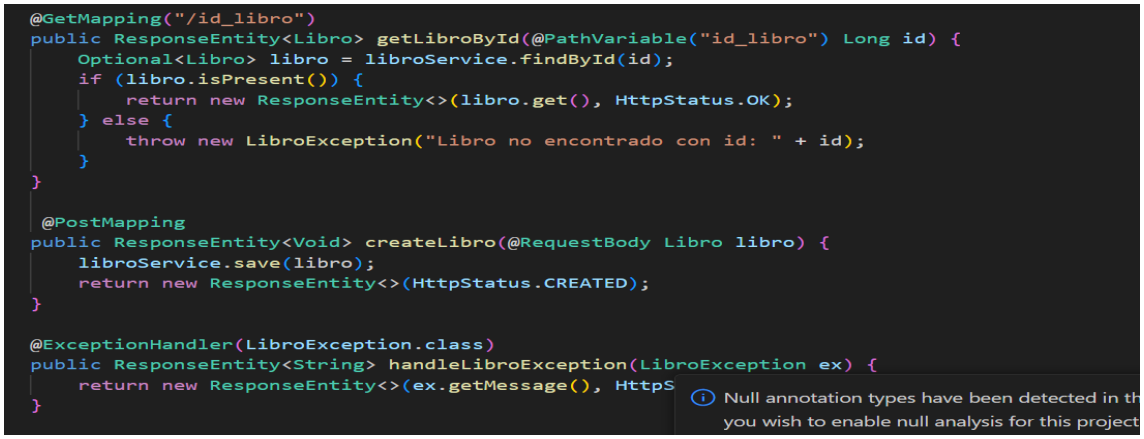
MODIFICACIONES DEL CÓDIGO

```java
@GetMapping
public List<Libro> getAllLibros() {
    return libroService.findAll();
}

@GetMapping("/id_libro")
public ResponseEntity<Libro> getLibroById(@PathVariable("id_libro") Long id) {
    Optional<Libro> libro = libroService.findById(id);
    if (libro.isPresent()) {
        return new ResponseEntity<>(libro.get(), HttpStatus.OK);
    } else {
        throw new LibroException("Libro no encontrado con id: " + id);
    }
}
```

Null annotation types have been detected in th

@GetMapping("/id_libro"): Maneja solicitudes GET para obtener un libro específico por su ID.
@PostMapping: Maneja solicitudes POST para crear un nuevo libro.



```java
@GetMapping("/id_libro")
public ResponseEntity<Libro> getLibroById(@PathVariable("id_libro") Long id) {
    Optional<Libro> libro = libroService.findById(id);
    if (libro.isPresent()) {
        return new ResponseEntity<>(libro.get(), HttpStatus.OK);
    } else {
        throw new LibroException("Libro no encontrado con id: " + id);
    }
}

@PostMapping
public ResponseEntity<Void> createLibro(@RequestBody Libro libro) {
    libroService.save(libro);
    return new ResponseEntity<>(HttpStatus.CREATED);
}

@ExceptionHandler(LibroException.class)
public ResponseEntity<String> handleLibroException(LibroException ex) {
    return new ResponseEntity<>(ex.getMessage(), HttpS
}
```

Null annotation types have been detected in th
you wish to enable null analysis for this project

@ExceptionHandler(LibroException.class): Captura excepciones de tipo LibroException y devuelve una respuesta con un mensaje de error y el estado HTTP NOT_FOUND.