

TASK 3 – CRUD E-commerce

Na data 10/10 criamos o CRUD do nosso projeto integrador. Realizamos a criação das classes de Controller e Repository: ProdutosController, CategoriaController, ProdutosRepository e CategoriaRepository. Também criamos nossas chaves estrangeiras.

Nas nossas classes Controller, criamos os seguintes métodos:

findAll()

findAllById()

findAllByTipos()

Put

Post

Delete

Depois testamos no Insomnia e subimos as alterações realizadas no Github.

CategoriaController.java

```
1 package com.capijava.capijava.controller;
2
3 import java.util.List;
4
5
6
7 @RestController
8 @RequestMapping("/categoria")
9 @CrossOrigin(origins = "*", allowedHeaders = "*")
10 public class CategoriaController {
11
12
13     @Autowired
14     private CategoriaRepository categoriaRepository;
15
16
17     @GetMapping
18     public ResponseEntity<List<CategoriaModel>> getAll() {
19         return ResponseEntity.ok(categoriaRepository.findAll());
20     }
21
22     @GetMapping("/{id}")
23     public ResponseEntity<CategoriaModel> getById(@PathVariable Long id) {
24         return categoriaRepository.findById(id).map(resposta -> ResponseEntity.ok(resposta))
25             .orElse(ResponseEntity.status(HttpStatus.NOT_FOUND).build());
26     }
27
28     @GetMapping("/tipos/{tipos}")
29     public ResponseEntity<List<CategoriaModel>> getByTitle(@PathVariable String tipos) {
30         return ResponseEntity.ok(categoriaRepository.findAllByTiposContainingIgnoreCase(tipos));
31     }
32
33     @PostMapping
34     public ResponseEntity<CategoriaModel> post(@Valid @RequestBody CategoriaModel categoria) {
35         return ResponseEntity.status(HttpStatus.CREATED).body(categoriaRepository.save(categoria));
36     }
37 }
```

```

63 public ResponseEntity<CategoriaModel> put(@Valid @RequestBody CategoriaModel categoria) {}
69
70 @ResponseStatus(HttpStatus.NO_CONTENT)
71 @DeleteMapping("/{id}")
72 public void delete(@PathVariable Long id) {
73     Optional<CategoriaModel> categoria = categoriaRepository.findById(id);
74
75     if (categoria.isEmpty())
76         throw new ResponseStatusException(HttpStatus.NOT_FOUND);
77
78     categoriaRepository.deleteById(id);
79
80 }
81 }
82

```

ProdutosController.java

```

1 package com.capijava.capijava.controller;
2
3 import java.util.List;
4
5
6
7
8
9
10 @RestController
11 @RequestMapping("/produtos")
12 @CrossOrigin(origins = "*", allowedHeaders = "*")
13 public class ProdutosController {
14
15     @Autowired
16     private ProdutosRepository produtosRepository;
17
18     @Autowired
19     private CategoriaRepository categoriaRepository;
20
21     @GetMapping
22     public ResponseEntity<List<ProdutosModel>> getAll() {
23         return ResponseEntity.ok(produtosRepository.findAll());
24     }
25
26     @GetMapping("/{id}")
27     public ResponseEntity<ProdutosModel> getById(@PathVariable Long id) {
28         return produtosRepository.findById(id).map(resp -> ResponseEntity.ok(resp))
29             .orElse(ResponseEntity.status(HttpStatus.NOT_FOUND).build());
30     }
31
32     @GetMapping("/nome/{nome}")
33     public ResponseEntity<List<ProdutosModel>> GetByNome(@PathVariable String nome) {
34         return ResponseEntity.ok(produtosRepository.findAllByNomeContainingIgnoreCase(nome));
35     }
36
37     @PostMapping
38     public ResponseEntity<ProdutosModel> post(@Valid @RequestBody ProdutosModel produto){
39         if (categoriaRepository.existsById(produto.getCategoria().getId()))
40             return ResponseEntity.status(HttpStatus.CREATED)
41                 .body(produtosRepository.save(produto));
42     }
43
44 }
45

```

```

70● @PutMapping
71 public ResponseEntity<ProdutosModel> put(@Valid @RequestBody ProdutosModel produto){
72     if (produtosRepository.existsById(produto.getId())){
73
74         if (categoriaRepository.existsById(produto.getCategoria().getId()))
75             return ResponseEntity.status(HttpStatus.OK)
76                 .body(produtosRepository.save(produto));
77
78         return ResponseEntity.status(HttpStatus.BAD_REQUEST).build();
79     }
80
81     return ResponseEntity.status(HttpStatus.NOT_FOUND).build();
82 }
83
84● @ResponseStatus(HttpStatus.NO_CONTENT)
85 @DeleteMapping("/{id}")
86 public void delete(@PathVariable Long id) {
87     Optional<ProdutosModel> produto = produtosRepository.findById(id);
88
89     if(produto.isEmpty())
90         throw new ResponseStatusException(HttpStatus.NOT_FOUND);
91
92     produtosRepository.deleteById(id);
93 }
94
95
96 }

```

ProdutosRepository.java

```

1 package com.capijava.capijava.repository;
2
3● import java.util.List;
10
11 @Repository
12 public interface ProdutosRepository extends JpaRepository<ProdutosModel, Long>{
13
14     public List<ProdutosModel>findAllByNomeContainingIgnoreCase(@Param("nome") String nome);
15
16
17
18
19 }
20

```

CategoriaRepository.java

```

1 package com.capijava.capijava.repository;
2
3● import java.util.List;
9
10 @Repository
11 public interface CategoriaRepository extends JpaRepository<CategoriaModel,Long> {
12
13     public List<CategoriaModel> findAllByTiposContainingIgnoreCase(String tipos);
14 }
15
16
17

```

Chave estrangeira - Classe CategoriaModel.java

```
32
33 @OneToMany(mappedBy = "categoria", cascade = CascadeType.REMOVE)
34 @JsonIgnoreProperties("categoria")
35 private List<ProdutosModel> produtos;
36
37 public Long getId() {
38     return id;
39 }
40
41 public List<ProdutosModel> getProdutos() {
42     return produtos;
43 }
44
45 public void setProdutos(List<ProdutosModel> produtos) {
46     this.produtos = produtos;
47 }
48
```

Chave Estrangeira – Classe ProdutosModel.java

```
52 @ManyToOne
53 @JsonIgnoreProperties("produto")
54 private CategoriaModel categoria;
55
56
57
```

```
114 public CategoriaModel getCategoria() {
115     return categoria;
116 }
117
118 public void setCategoria(CategoriaModel categoria) {
119     this.categoria = categoria;
120 }
121
122
```

Testes no Insomnia



