



ANÁLISE E PROJETO ORIENTADO A OBJETOS

Trabalho 1

(Bruno Moretto, Mateus Dotto e Paulo Pires de Avila)



Planejamento

- Linguagem JAVA
- Framework: Spring Boot
 - Spring Web
 - Spring Data JPA
 - H2 Database



Aplicação

- API REST
- Postman
 - Chamadas/Requisições

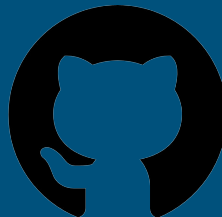


POSTMAN

Organização do Trabalho

- GitHub
 - Versionamento
 - KANBAN

Source Code
Management



Implementação do Trabalho

- Dois projetos (um para cada exercício)
- Cada projeto foi dividido em três partes
 - Criação e configuração inicial
 - Criação das entidades e sua integração com o Banco de Dados
 - Criação dos recursos para atender os requisitos do trabalho

Snippets

- Exercício 1

```
1 package com.poo.trab1.entity;
2
3 import lombok.AllArgsConstructor;
4 import lombok.Builder;
5 import lombok.Data;
6 import lombok.NoArgsConstructor;
7
8 import javax.persistence.*;
9 import javax.validation.constraints.NotBlank;
10 import java.util.List;
11
12 @Entity
13 @Table(name = "EMPRESA")
14 @Data
15 @AllArgsConstructor
16 @NoArgsConstructor
17 @Builder
18 public class Empresa {
19
20     @Id
21     @GeneratedValue(strategy = GenerationType.IDENTITY)
22     private Long id;
23
24     @Column(name = "NOME")
25     private String nome;
26
27     @Column(name = "CNPJ")
28     private String cnpj;
29
30     @OneToMany(fetch = FetchType.LAZY, mappedBy = "empresa")
31     private List<Funcionario> funcionarios;
32
33 }
```

Snippets

- Exercício 2

```
1 package com.poo.trab1.entity;
2
3 import lombok.AllArgsConstructor;
4 import lombok.Builder;
5 import lombok.Data;
6 import lombok.NoArgsConstructor;
7
8 import javax.persistence.Entity;
9 import java.time.LocalDate;
10
11 @Entity
12 @NoArgsConstructor
13 public class Gerente extends Funcionario {
14
15     @Override
16     public Long getId() {
17         return super.getId();
18     }
19
20     @Override
21     public String getNome() {
22         return super.getNome() + " | GERENTE";
23     }
24
25     @Override
26     public Double getSalario() {
27         return super.getSalario();
28     }
29 }
```

```
29
30 @Override
31 public LocalDate getDtAdmissao() {
32     return super.getDtAdmissao();
33 }
34
35 @Override
36 public Empresa getEmpresa() {
37     return super.getEmpresa();
38 }
39
40 @Override
41 public Departamento getDepartamento() {
42     return super.getDepartamento();
43 }
44
45 @Override
46 public void aumentaSalario() {
47     this.setSalario(this.getSalario() + this.getSalario() * 0.15);
48 }
49
50 }
```

Demonstração

- Execução e demonstração ao vivo dos projetos