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

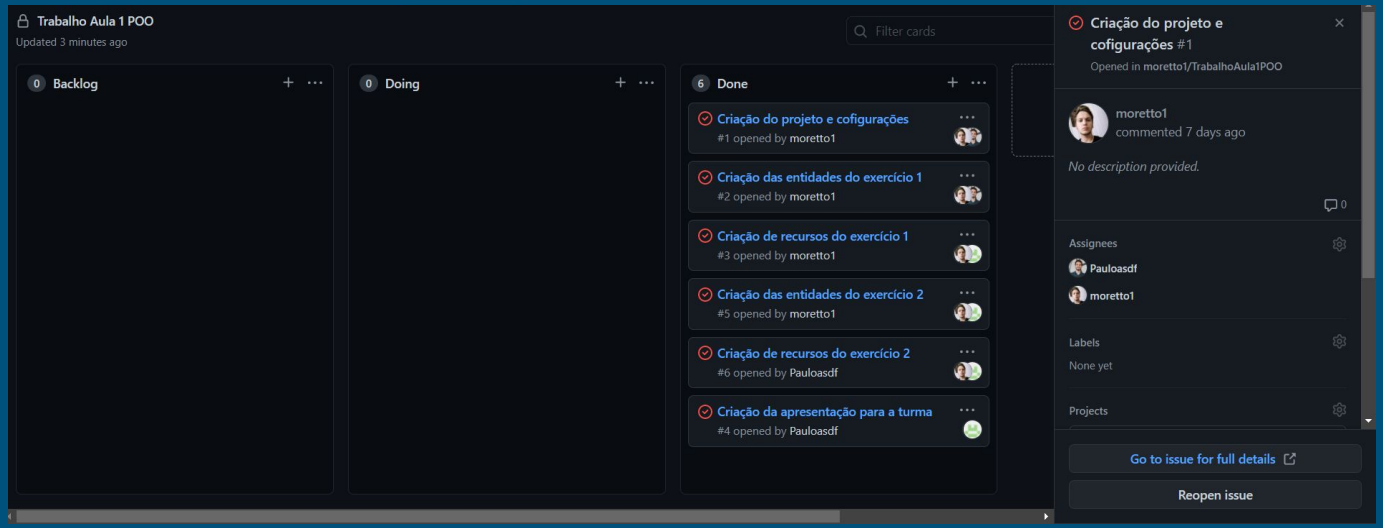
Aplicação

- API REST
- Postman
  - Chamadas/Requisições

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

```java
package com.poo.trab1.entity;

import lombok.AllArgsConstructor;
import lombok.Builder;
import lombok.Data;
import lombok.NoArgsConstructor;

import javax.persistence.*;
import javax.validation.constraints.NotBlank;
import java.util.List;

@Entity
@Table(name = "EMPRESA")
@Data
@AllArgsConstructor
@NoArgsConstructor
@Builder
public class Empresa {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    @Column(name = "NOME")
    private String nome;

    @Column(name = "CNPJ")
    private String cnpj;

    @OneToMany(fetch = FetchType.LAZY, mappedBy = "empresa")
    private List<Funcionario> funcionarios;

}
```

Snippets

- Exercício 2

```java
package com.poo.trab1.entity;

import lombok.AllArgsConstructor;
import lombok.Builder;
import lombok.Data;
import lombok.NoArgsConstructor;

import javax.persistence.Entity;
import java.time.LocalDate;

@Entity
@NoArgsConstructor
public class Gerente extends Funcionario {

    @Override
    public Long getId() {
        return super.getId();
    }

    @Override
    public String getNome() {
        return super.getNome() + " | GERENTE";
    }

    @Override
    public Double getSalario() {
        return super.getSalario();
    }

    @Override
    public LocalDate getDtAdmissao() {
        return super.getDtAdmissao();
    }

    @Override
    public Empresa getEmpresa() {
        return super.getEmpresa();
    }

    @Override
    public Departamento getDepartamento() {
        return super.getDepartamento();
    }

    @Override
    public void aumentaSalario() {
        this.setSalario(this.getSalario() + this.getSalario() * 0.15);
    }

}
```

Demonstração

- Execução e demonstração ao vivo dos projetos