



SERVIÇO PÚBLICO FEDERAL · MINISTÉRIO DA EDUCAÇÃO

UNIVERSIDADE FEDERAL DE VIÇOSA · UFV

CAMPUS FLORESTAL

## **Projeto de Desenvolvimento**

# **SISTEMA PARA GERENCIAMENTO DE TRABALHOS DE CONCLUSÃO DE CURSO**

Cleidimar L. dos Passos - **[EF03473]**

João Vitor G. Vieira - **[EF04212]**

Juan Pablo A. Avelar - **[EF04229]**

Leonardo Araujo R. Aguiar - **[EF04702]**

Vinicius Augusto A. Ferreira - **[EF04242]**

Florestal  
2022

# Sumário

<b>1. Introdução</b>	<b>2</b>
<b>2. Desenvolvimento</b>	<b>3</b>
2.1 MVC	3
2.2 Package Model	4
2.3 Package View	4
2.4 Package Controle	5
2.5 Package DAO e Banco de Dados	5
2.6 Como conectar ao banco de dados	7
<b>3. Conclusão</b>	<b>8</b>
<b>4. Referências</b>	<b>9</b>

# 1. Introdução

Este trabalho tem como objetivo principal a criação de um sistema para gerenciamento de trabalho de conclusão de curso e, para a criação deste software foi utilizado a linguagem de programação orientada a objetos JAVA. Diante desse contexto, o sistema deve possuir algumas funcionalidades especificadas na documentação, como o armazenamento desses trabalhos (POC) postados pelos professores, único usuário capaz de cadastrar, pesquisar, editar e remover esses trabalhos, enquanto os alunos cadastrados no sistema podem apenas pesquisá-los. Além disso, também existe o papel de administrador, o único capaz de fazer alterações nos usuários.

## 2. Desenvolvimento

Para o armazenamento do projeto foi utilizado a plataforma GitHub como hospedagem de código-fonte e controle de versionamento dos arquivos, para ajudar no desenvolvimento do projeto. Abaixo vamos falar sobre as especificações do projeto.

### 2.1 MVC

MVC(model-view-controller) para garantir uma organização dinâmica e eficiente. Essa separação de camadas ajuda na redução de acoplamento e promove o aumento de coesão nas classes do projeto.

O **Model** é responsável pelo armazenamento das classes e seus principais métodos como construtores, gets e sets.

A **View** é responsável pela interação com o usuário e tudo aquilo que ele é capaz de visualizar, seja pelo terminal ou por interface gráfica.

O **Controller** é responsável por gerenciar vários elementos de dados e modelar o problema a ser resolvido por meio de métodos.

Ademais, também foi implementado o **DAO** que é a camada de persistência do código responsável por administrar os dados no Banco de Dados.

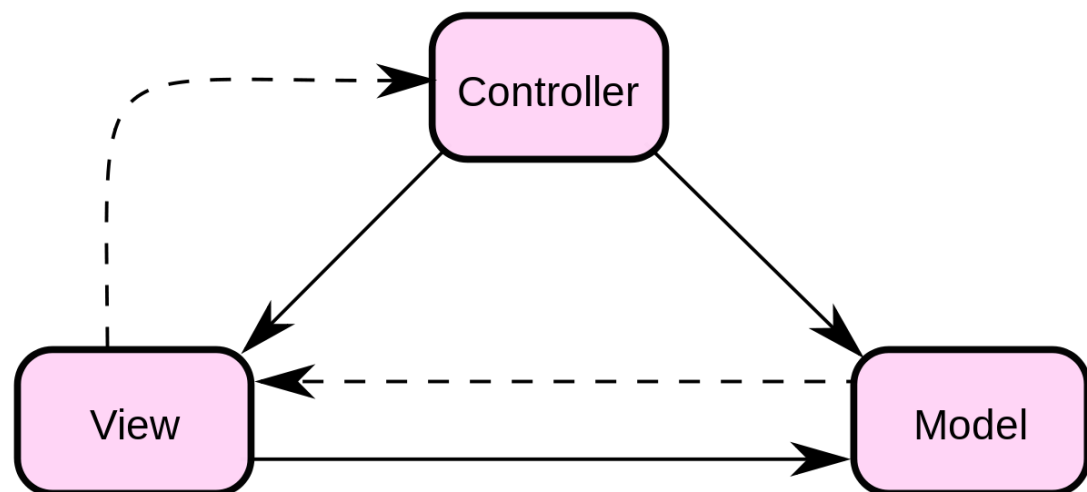


Figura 1: Ilustração do MVC

## 2.2 Package Model

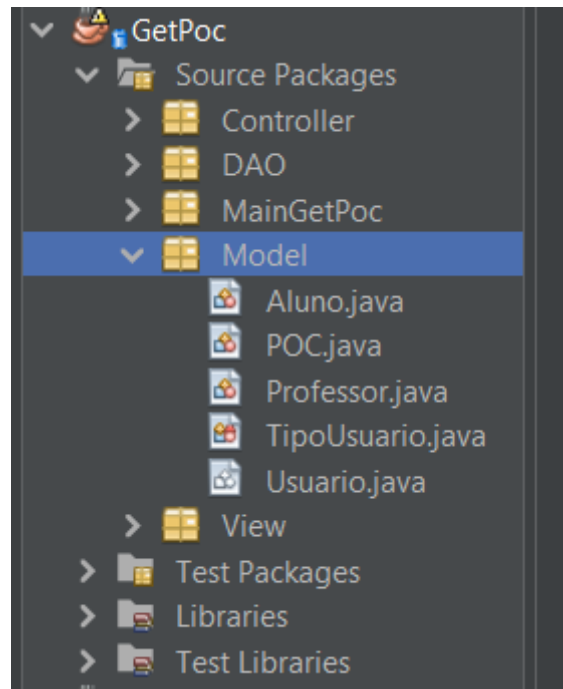


Figura 2: Pacote Model

O modelo do projeto possui **TipoUsuario.java** que é um **ENUM** para qual o tipo de usuário (professor ou aluno). A classe usuário com o nome, matrícula, senha e tipo do usuário. E as classes Professor e Aluno que herdam da classe Usuario. Todas as classes possuem seus respectivos construtores, getters e setters.

## 2.3 Package View

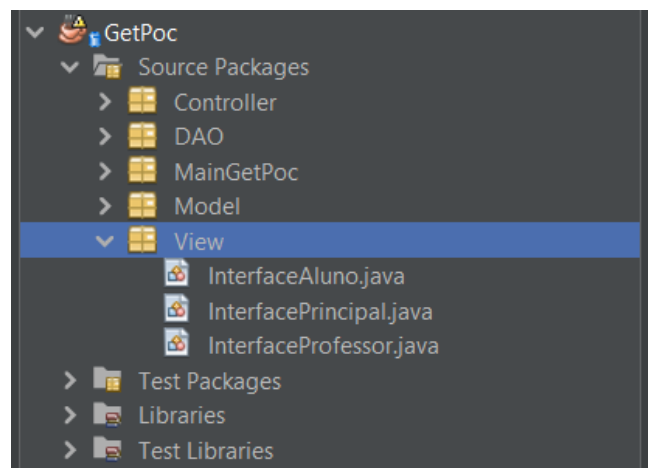


Figura 3: Pacote View

A visão é responsável pela interação com o usuário, neste trabalho foi utilizado o próprio terminal para esta finalidade. A **InterfacePrincipal.java** é o primeiro contato do usuário com o software, no qual é mostrado para ele a interface de login e de cadastro. Após o login do usuário é mostrado a **InterfaceAluno.java** ou **InterfaceProfessor.java** dependendo de quem logou no sistema, em cada uma das interfaces é mostrada as suas respectivas opções para interagir com o sistema.

## 2.4 Package Controle

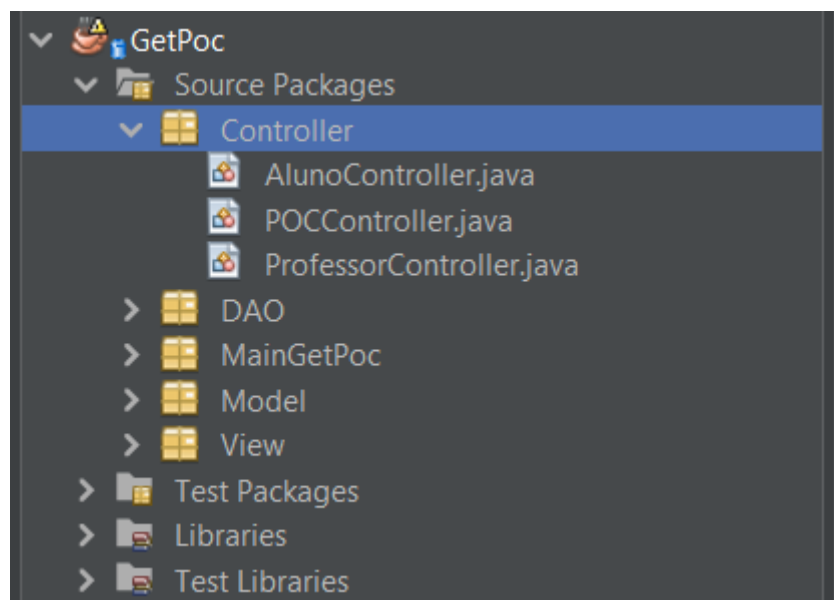


Figura 4: Pacote Controller

O controller é responsável pela manipulação e leitura dos dados, além de suas validações. Dessa forma, cada **Controller.java** possui os métodos que realizam essas manipulações em cada classe.

## 2.5 Package DAO e Banco de Dados

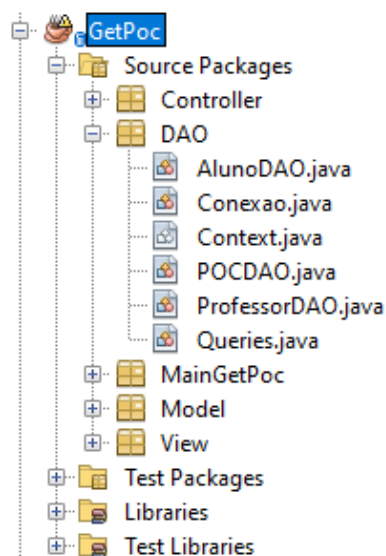
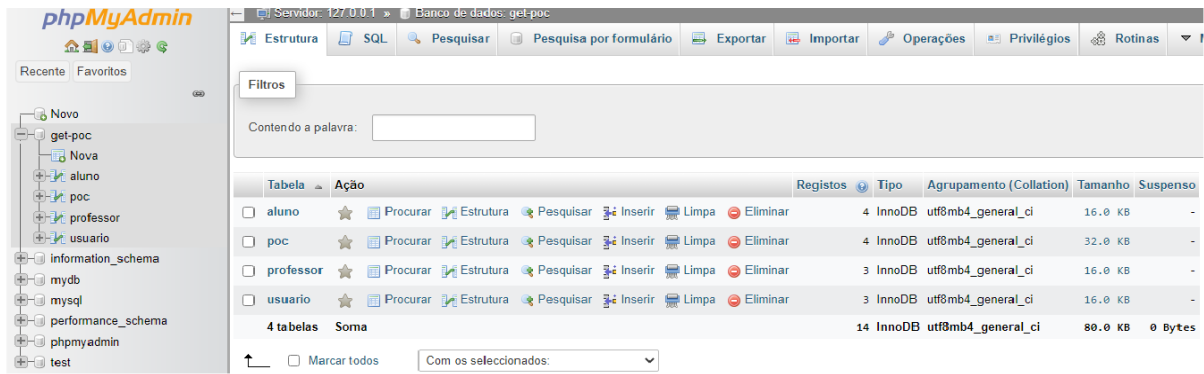


Figura 5: Pacote DAO e Banco de Dados

O pacote **DAO** é responsável pela persistência dos dados e mudanças realizadas durante a execução da aplicação. A classe **AlunoDAO** contém um **ArrayList** que armazena os dados dos alunos cadastrados. A classe **ProfessorDAO** também contém um **ArrayList** que armazena os dados dos professores cadastrados. A classe **POCDAO**, por sua vez, tem a função de armazenar os dados dos POC's. Estas 3 classes tem como função principal armazenar os dados de seus respectivos modelos para agilidade de consulta.

As classes **Context** possuem atributos que são responsáveis por armazenar o status do usuário e da execução da aplicação, como por exemplo, se o usuário está logado se é professor ou aluno e o estado dos POCs no **ArrayList**. As classes **Conexao** e **Queries** são responsáveis pela conexão e persistência dos dados no banco de dados **MYSQL**, a classe **Conexao** possui os atributos e métodos necessários para a conexão com o banco de dados.

A classe **Queries** possui os métodos com os comandos em **SQL** para alterar o banco de dados. No banco de dados existem 3 tabelas essenciais para o funcionamento do projeto: a tabela **poc**, a tabela **professor** e a tabela **aluno**. Existe também uma tabela usuários para testes, mas não é necessária para a execução da aplicação.



## 2.6 Como conectar ao banco de dados

Para a conexão com o banco de dados é necessário que o usuário tenha instalado o xampp e que o apache e o mysql estejam ativos, o usuário deverá criar uma base de dados com o nome “get-poc” e em seguida importar a arquivo sql para a base de dados. Após executar estes passos o usuário poderá executar o projeto e então os dados serão capturados das tabelas do banco para os ArrayLists da aplicação.



### 3. Conclusão

O objetivo principal deste trabalho foi desenvolver um software capaz de gerenciar trabalhos de conclusão de curso, por meio da linguagem orientada a objetos Java. Logo, é possível concluir ao final que foi implementado todos os quesitos solicitados na documentação do projeto.

Foi possível colocar em prática e reforçar todos os conceitos vistos em salas de aula na disciplina CCF 313 ministrada pelo professor Fabrício, como, modularidade, objetos e classes, encapsulamento dos dados, padrão de projeto MVC com DAO, herança, tratamento de exceção, testes unitários, dentre outras.

Além disso, foi possível compreender melhor também as linguagens de programação orientadas a objetos e seus quatro principais pilares: polimorfismo, herança, abstração e encapsulamento.

Ademais, habilidades de soft skills também foram trabalhadas ao longo do desenvolvimento deste trabalho, como comunicação, trabalhar em equipe, organização e planejamento.

## 4. Referências

- 1) <https://stackoverflow.com/>
- 2) <https://www.w3schools.com/sql/>
- 3) [Java ResultSet - javatpoint](#)
- 4) <https://www.apachefriends.org/>
- 5) [Java Progressivo](#)
- 6) [Create a Database Diagram in MySQL Workbench | InMotion Hosting](#)
- 7) [Java Statement - javatpoint](#)
- 8) [Criando uma conexão Java + MySQL Server \(devmedia.com.br\)](#)
- 9) <https://github.com/>