CCF313 - Programação Orientada a Objetos
Trabalho Prático Final

Germano Barcelos dos Santos (3873)[1], Fábio Trindade Ramos (3869)[2], Dener Vieira Ribeiro (3872)[3]

[1]Instituto de Ciências Exatas e Tecnológicas, Campus UFV-Florestal
Universidade Federal de Viçosa

1. Introdução

Neste trabalho foi proposto desenvolver um sistema utilizando orientação a objetos e a linguagem Java. Foi desenvolvido então um sistema onde é possível cadastrar e contratar aulas, onde essas interações são feitas entre alunos e professores.

2. Implementação

Para implmentar este sistema foi utilizado o padrão de projeto MVC(Model-View-Controller), o padrão DAO(Data Access Object),o padrão DI(Dependency Injection) e o SGBD MySQL para utilizar banco de dados.

2.1. Models

Os models são as classes que realizam as regras de negócio,ou seja, é a implementação das entidades que foram criadas a partir da abstração dos elementos que são necessários para a criação do sistema, sendo eles: Professor, Aluno, Aula, Avaliação, Contrato, ContratoEtapa, Disciplina e Usuário. Professor e aluno são extensões de usuário e ContratoEtapa uma Enumeração que se refere a três estados do contrato: aceito, em negociação e rejeitado.

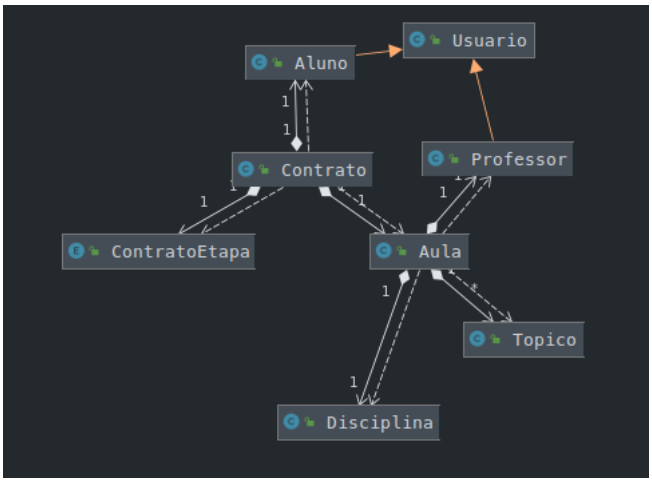

Figura 1. Entidades

Esse diagrama (figura 2) de classes também teve que ser modelado no MySQL.

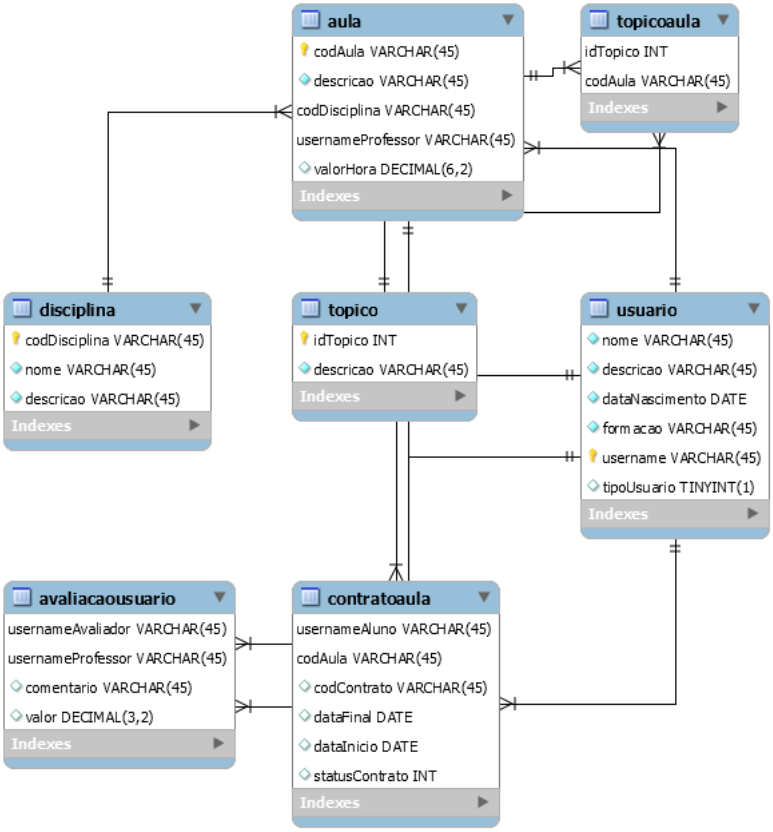

Figura 2. Modelo do Projeto do Banco de Dados

2.2. Views

A view consiste na implementação gráfica do sistema, é onde ocorre a interação direta entre o usuário e o sistema. Foram implementadas as seguintes views: cadastrar usuário, contratos aceitos, contratos pendentes, login, página inicial [1], perfil usuário e main view.

2.3. Controllers

O controller é a parte intermediária entre a view e o model, ou seja, a view passa para o controller dados que foram fornecidos, e o controller utiliza os models para realizar a regra de negócio sobre os dados, retornando para o controller os dados obtidos e o controller retornando para a view os dados recebidos. Sendo assim foi necessário a criação de controllers para 6 models, sendo eles: Aluno, Professor, Disciplina, Feed(que representa..), Professor, Tópico e Usuario.

2.4. Driver MySQL

Para que fosse possível o uso do MySQL como banco de dados, foi necessário o uso de um driver. Esse driver pode ser obtido diretamente da documentação do MySQL, como um executável *.jar*. No entanto, optamos por utilizar um gerenciador de dependências, chamado Maven.

O Maven é um repositório de executáveis *.jar* que gerencia as dependências por meio de códigos.

[1] Tela que contém as aulas já cadastradas

2.5. DAO

O DAO(Data Access Object) é um padrão de projeto que abstrai e encapsula os mecanismos de acesso a dados escondendo os detalhes da execução da origem dos dados.

2.6. DI(Dependency Injection)

É um padrão de projeto que aplica o quinto principio SOLID, que se refere a letra D. Ele diz que: ”Módulos de alto nível não devem depender de módulos de baixo nível, ambos devem depender de abstrações”, ou seja, devemos criar interfaces para que classes DAOs sejam implementações.

Assim, as classes pertencentes ao pacote *controllers* dependem das interfaces. Portanto, podemos alterar as implementações da persistência de dados facilmente.

3. Utilização

Para utilizar o projeto com banco de dados é necessário utilizar o *docker* e executar o comando: *docker-compose up -d*. Esse comando cria a instância MySQL e também executa queries de inserção de dados automaticamente.

Além disso, desenvolvemos uma versão com persistência local, e assim basta executar o .jar do projeto.

4. Conclusão

Com a criação deste sistema foi possível aplicar os fundamentos de programação orientada a objeto, além de adquirir experiência para o trabalho em grupo e uso de ferramentas que auxiliam o trabalho em grupo,como o GitHub e o Trello, além da implementação de padrões de projeto e conexão entre um aplicativo e um banco de dados, sendo assim todos os objetivos propostos foram alcançadas.