Cadastro e Gerenciamento de Alunos

Baseado nos requisitos de boas práticas da programação orientada a objetos, preparação para conexão com um banco de dados central no futuro e robustez para suportar um sistema maior, modular e expansível, o projeto foi desenvolvido baseado no padrão de arquitetura MVC (Model-View-Controller), utilizando Java 24 pela sua robustez, juntamente do software IntelliJ IDEA. Para a estruturação do projeto foram criados os seguintes pacotes para cada camada:

Model

- Entity: representação das entidades no banco de dados.
- Repository: gerenciamento do banco de dados (queries).
- Service: regra de negócio, conversão entity/dto e vice-versa.
- Dto (Data Transfer Object): transporte de dados entre camadas.
- Type: tipos auxiliares como enums.
- Util: classes utilitárias.

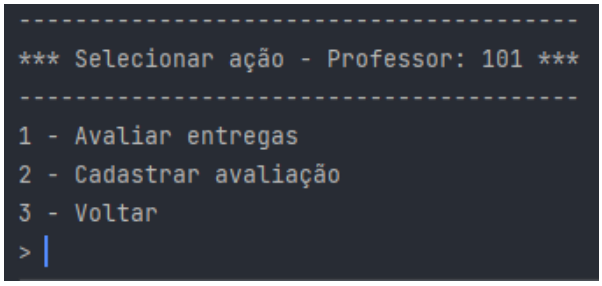
Controller

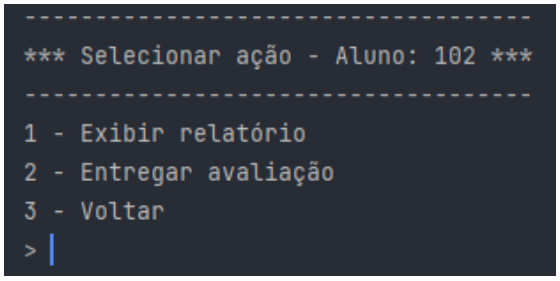
- Controller: responsável por receber requisições e retornar respostas.

View

- Ui: responsável por requisições para a controller e apresentação dos dados.

A partir da estrutura criada, foi desenvolvida a seguinte lógica da aplicação: existem alunos e funcionários nas instituições. Os funcionários foram definidos inicialmente como professor e administrador, onde o professor pode criar avaliações para as turmas em que está inserido de forma individual, bem como avaliar todas as avaliações da turma. O aluno só poderá ser avaliado caso tenha feito a entrega da avaliação, que será disponibilizada assim que o professor criar a avaliação ou, em caso de entrada tardia, as avaliações já criadas anteriormente também serão disponibilizadas para a entrega. Segue abaixo a visão de professor e aluno respectivamente:

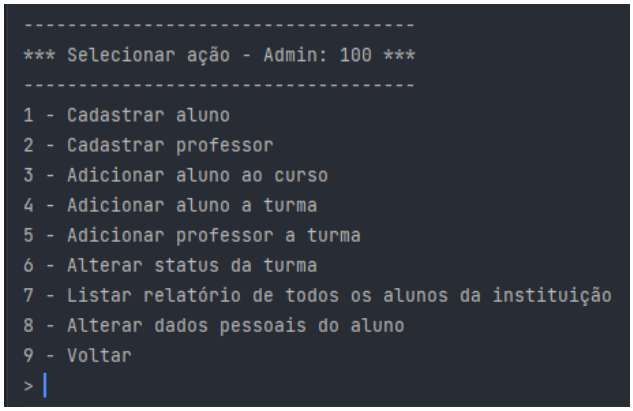




O aluno poderá ser inserido em cursos (da sua instituição), que possuem sua grade de disciplinas (as disciplinas podem ser compartilhadas entre cursos ou não) e só poderá ser inserido em uma turma caso algum dos cursos do aluno possua a disciplina da turma.

A entrega só será disponibilizada para o aluno caso ele esteja inserido na mesma turma que foi criada a avaliação pelo professor. O professor também só poderá ministrar disciplinas da sua instituição e, da mesma forma, o administrador pode fazer o gerenciamento da sua instituição.

Dentre as coisas que o administrador pode fazer, é permitido cadastrar alunos e professores, alterar dados pessoais dos alunos, adicionar alunos em cursos e turmas, adicionar professores para ministrarem múltiplas turmas, alterar o status da turma e gerar o relatório de todos os alunos da sua instituição. Segue abaixo a visão do administrador:



Para versões futuras, é possível a criação do funcionário coordenador, que poderá criar cursos, adicionar disciplinas aos cursos e criar disciplinas e turmas de forma fácil, sendo necessário apenas chamar a *controller* específica já implementada. Além da facilidade de utilizar as *controllers* já desenvolvidas para fazer o gerenciamento, é possível vinculá-las a *endpoints*, sendo possível fazer futuramente solicitações web com um *front-end* desenvolvido à parte. O banco de dados também pode ser integrado facilmente já que foi feita a divisão de responsabilidades para o pacote *repository*, que pode ser refatorado isoladamente, afetando minimamente o funcionamento dos outros pacotes.