

Projeto final: Bancos de Dados

Bernardo Manciola Lobo
211026486

Introdução

O objetivo desse trabalho é a elaboração de um sistema onde estudantes possam avaliar professores e disciplinas. O programa foi criado utilizando o framework Flask e o PostgreSQL como SGBD.

Planejamento

No processo de planejamento do trabalho, foi necessário pensar na melhor forma de construir o banco de dados que seria utilizado no trabalho. Para isso, foi criado um diagrama entidade relacionamento e de um modelo entidade relacionamento. A visualização desses dois modelos de bancos de dados facilita a modelagem e construção do sistema, tendo em vista as entidades, relações e atributos contemplados pelo modelo.

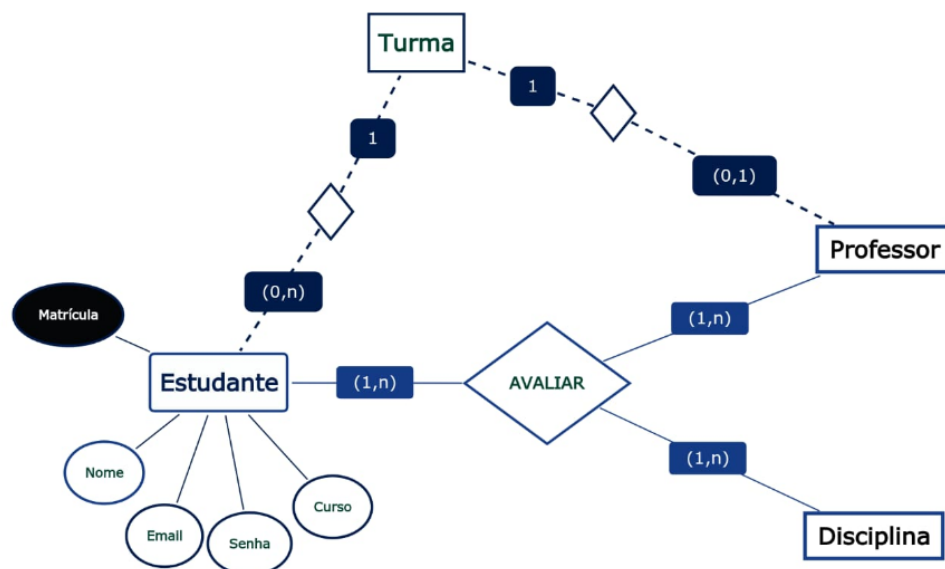


Figura 1: Modelo ER do Banco de Dados do sistema

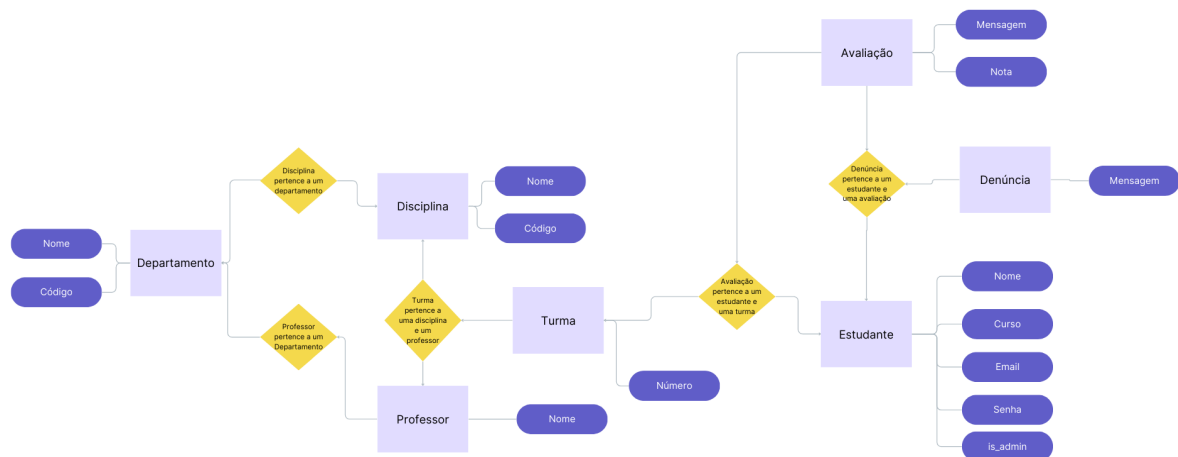


Figura 2: Diagrama ER do Banco de Dados do sistema

A elaboração desses dois modelos permitiu a construção de um banco de dados mais robusto. Nota-se que o diagrama ER construído possui adaptações ao modelo ER, tornando-o mais eficiente e menos redundante em relação à análise original do problema.

Análise das tabelas

A normalização das tabelas consiste na análise que minimize a redundância em um banco de dados relacionais. Para verificar que nossos modelos foram bem construídos, analisaremos a forma normal de 3 das nossas tabelas principais: Estudantes, Turmas e Avaliações.

A forma normal deve garantir que:

- Todos os atributos do modelo sejam atômicos;
- Atributos que não são chave dependam unicamente da chave primária e devem ser independentes entre si;
- Não existam dependências multivaloradas;
- Não possua dependências de junção;

A tabela de estudantes possui 6 atributos: id (chave primária), nome, curso, email, senha e is_admin (booleano que indica se o usuário é administrador do sistema). Nota-se que todos os atributos são indivisíveis, não dependem um dos outros (dependendo unicamente da chave primária) e não são multivalorados. É possível construir a tabela pela junção de quaisquer subtabelas, garantindo então a normalização da tabela Estudantes.

A tabela de Turmas possui os atributos: id (chave primária), número, id_disciplina (chave-estrangeira de disciplinas) e id_professor (chave estrangeira de professores). Mais uma vez, é reconhecível a independência e atomicidade dos atributos, e a possibilidade de construção por meio das subtabelas, garantindo então a normalização da tabela Turmas.

A tabela de Avaliações possui os atributos: id (chave primária), mensagem, nota, id_turma (chave estrangeira de turma) e id_estudante (chave estrangeira de estudantes). Por fim, é possível afirmar que os atributos são independentes e indivisíveis, dependem unicamente da chave primária e podem compor subtabelas que formam a tabela original. Dessa forma, podemos garantir a normalização da tabela Avaliações.

Implementação

Com o banco de dados devidamente planejado e avaliado, a construção do banco de dados se dá por meio do seguinte diretório:

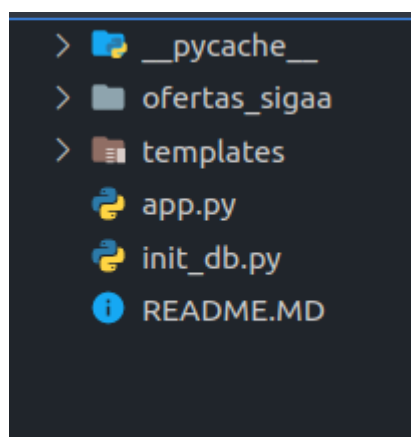


Figura 3: representação do diretório do sistema

A inicialização do banco de dados é feita pelo arquivo “init_db.py”. Esse arquivo contém os parâmetros necessários para a comunicação com o banco de dados PostgreSQL, e os scripts SQL para a criação e povoamento das tabelas relacionais necessárias para o funcionamento do sistema, tal qual foi projetado nos diagramas.

O arquivo “app.py” contém a lógica de negócio da aplicação em suas respectivas funções. Cada função desse arquivo pode ser abstraída como uma funcionalidade implementada em sua devida página.

A pasta “templates” possui os arquivos html que geram a interface com usuário do projeto, a pasta “ofertas_sigaa” contém os arquivos csv disponibilizados para a população do banco de dados, e a pasta “__pycache__” possui arquivos necessários para o funcionamento do framework.

O projeto possui as funcionalidades de:

- Visualizar professores, departamentos, disciplinas e turmas
- Registrar um novo usuário
- Visualizar avaliações de turmas
- Enviar avaliação de turma com login e senha de usuário
- Denunciar avaliações
- Visualizar denúncias e excluir os reports (caso logado como administrador)

O projeto pode ser visualizado pelo vídeo disponível no youtube:
<https://youtu.be/Tvy0AIS7GdQ>

O código está disponível em: https://github.com/tlamen/Trabalho_BD

Conclusão

Infelizmente, por questões pessoais de organização e gestão de tempo, não foi possível realizar um trabalho tão bom quanto eu gostaria. Apesar disso, considero satisfatórias as funcionalidades implementadas no sistema construído.

Foi possível aplicar os conhecimentos de modelamento de banco de dados, construção de bancos de dados, SGBDs, SQL e lógica de negócio, além de aprofundar meus conhecimentos em python e no framework flask para a construção de aplicações web.