

Projeto Final

Bancos de Dados

Tiago Leão Buson, 20/0034162

¹Dep. Ciência da Computação – Universidade de Brasília (UnB)

tiagobuson@gmail.com

1. Introdução

Este relatório apresenta o projeto de desenvolvimento de um sistema de avaliação de professores e disciplinas, com o objetivo de auxiliar os estudantes na escolha de suas disciplinas. O projeto foi concebido como uma oportunidade para aplicar os conhecimentos adquiridos na disciplina de Bancos de Dados e oferecer uma solução prática e eficiente para a comunidade estudantil.

O sistema desenvolvido permitirá que os estudantes se cadastrem e compartilhem avaliações sobre os professores e turmas das diferentes disciplinas oferecidas nos semestres acadêmicos. Comentários ofensivos ou inadequados poderão ser denunciados pelos usuários, desencadeando uma avaliação por parte dos administradores responsáveis pelo sistema. Os administradores, por sua vez, terão a autoridade para ignorar as denúncias, aceitá-las e remover os comentários ofensivos, ou até mesmo excluir a conta de um estudante infrator.

Além disso, os estudantes também terão a autonomia de editar e excluir seus próprios comentários, permitindo que atualizem suas avaliações à medida que progridem em suas experiências acadêmicas. Esse recurso visa garantir a transparência e a confiabilidade do sistema, bem como fornecer aos usuários a capacidade de corrigir eventuais erros ou equívocos em suas avaliações.

Neste relatório, serão detalhados o Diagrama de Entidade Relacionamento, o Modelo Relacional, e uma avaliação das formas normais de algumas tabelas utilizadas no sistema.

O projeto busca fornecer uma plataforma confiável e útil para os estudantes, permitindo que tomem decisões informadas ao escolherem suas disciplinas e professores. Espera-se que a implementação desse sistema contribua para uma comunidade acadêmica mais informada e colaborativa, promovendo um ambiente de aprendizado enriquecedor e aprimorando a qualidade do ensino.

Foi feito um vídeo ([link](#)) apresentando o programa e o sistema que foi produzido para este projeto e um repositório ([link](#)) do github com todo o projeto.

Para a interface gráfica, foi utilizada a linguagem [Python 2023], com a biblioteca do [Tkinter 2023], e para a implementação da persistência em banco de dados foi utilizado o SGBD do [Postgres 2023], utilizando a biblioteca de Python [Psycopg2 2023] para fazer a conexão com a interface.

2. Diagrama de Entidade Relacionamento

Um diagrama entidade relacionamento (DER) é uma representação gráfica de um modelo de dados que descreve as entidades (objetos ou conceitos) envolvidos em um sistema, seus atributos e os relacionamentos entre elas. Ele é amplamente utilizado na área de bancos de dados para projetar e comunicar a estrutura e as relações dos dados em um sistema.

O diagrama entidade relacionamento ajuda a visualizar e entender a estrutura de um banco de dados, mostrando quais informações são armazenadas, como elas estão relacionadas entre si e quais são as restrições aplicáveis. Ele serve como base para o projeto físico do banco de dados, auxiliando no desenvolvimento de esquemas de tabelas, chaves primárias, chaves estrangeiras e outras estruturas necessárias para armazenar e recuperar os dados de forma eficiente.

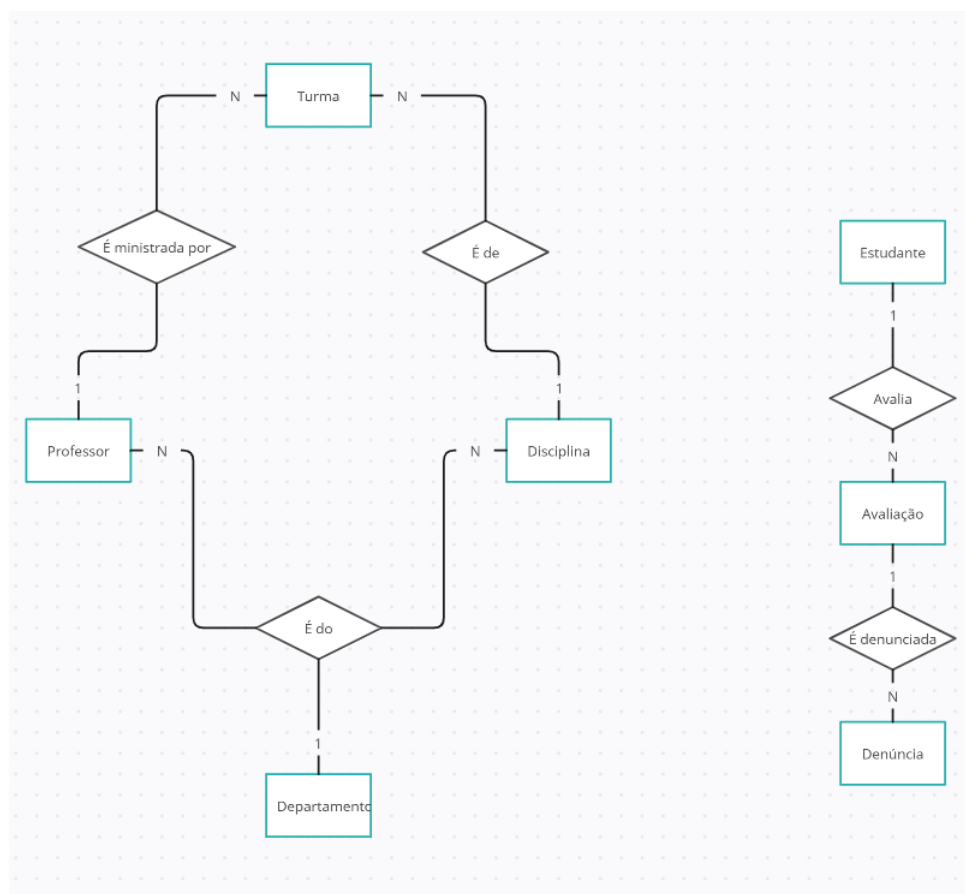


Figure 1. Diagrama de Entidade Relacionamento do Sistema

3. Modelo Relacional

Um modelo lógico relacional é uma representação abstrata de um banco de dados relacional, que descreve a estrutura e as relações entre as tabelas que compõem o banco de dados. Ele define como os dados são organizados e armazenados, fornecendo um plano conceitual para a implementação do banco de dados.

No modelo lógico relacional, os dados são organizados em tabelas, onde cada tabela representa uma entidade ou um relacionamento entre entidades. Cada tabela é com-

posta por colunas, que representam os atributos, e linhas, que representam as instâncias dos dados.

As relações entre as tabelas são estabelecidas por meio de chaves, especificamente as chaves primárias e as chaves estrangeiras. A chave primária identifica de forma única cada registro em uma tabela, enquanto a chave estrangeira estabelece uma relação entre duas tabelas, referenciando a chave primária de outra tabela.

Em resumo, um modelo lógico relacional é uma representação conceitual dos dados em um banco de dados relacional, descrevendo a estrutura das tabelas, os atributos e as relações entre elas. Ele serve como um guia para a implementação do banco de dados, proporcionando uma visão organizada e consistente dos dados a serem armazenados e manipulados.

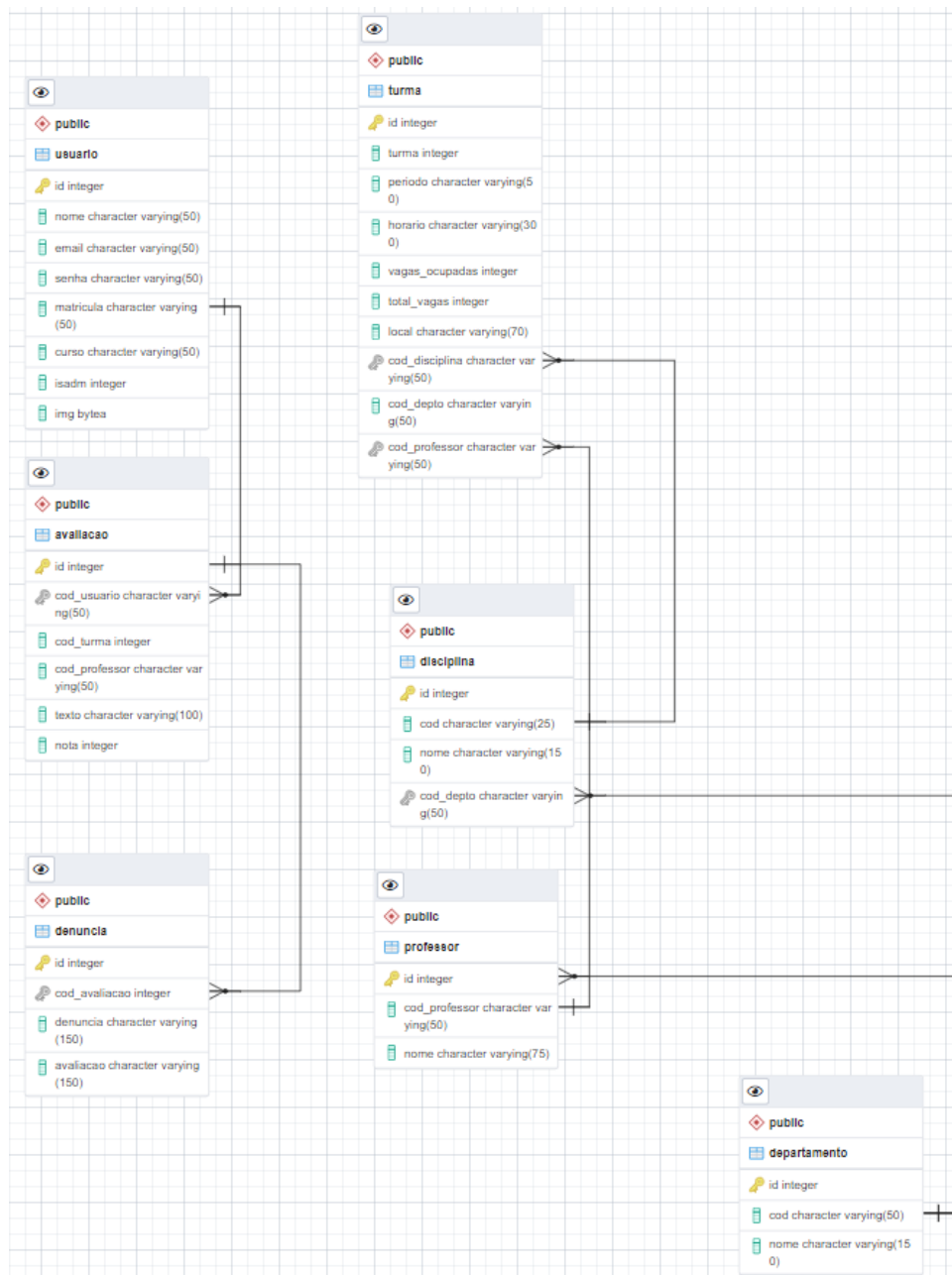


Figure 2. Modelo Relacional do Sistema

4. Avaliação das formas Normais

A normalização de tabelas em bancos de dados é um processo que visa organizar as informações de maneira eficiente, evitando redundâncias e inconsistências. Existem várias formas normais (1NF, 2NF, 3NF, etc.) que estabelecem critérios para garantir a integridade e a consistência dos dados. A seguir são dados os passos que forma utilizados para avaliar se as tabelas estão na terceira forma normal (3NF):

1. Verificar se a tabela está na primeira forma normal (1NF):
 - Cada coluna da tabela deve conter apenas valores atômicos, ou seja, valores indivisíveis.

- Não deve haver repetições de grupos de dados em uma mesma coluna.

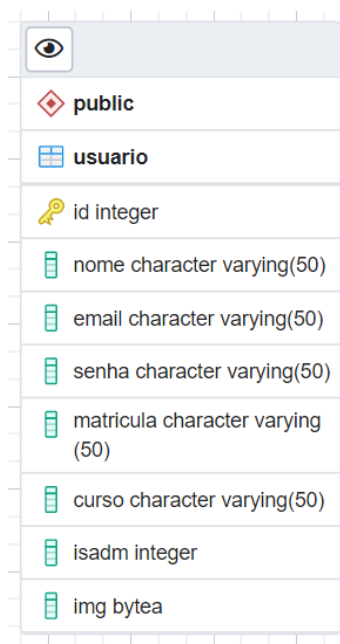
2. Verificar se a tabela está na segunda forma normal (2NF):

- A tabela deve estar na 1NF.
- Todos os atributos não chave devem depender completamente da chave primária.
- Se houver dependências parciais (atributos não chave dependendo de apenas parte da chave), é necessário decompor a tabela em duas ou mais tabelas para remover essas dependências.

3. Verificar se a tabela está na terceira forma normal (3NF):

- A tabela deve estar na 2NF.
- Todos os atributos não chave devem depender diretamente da chave primária.
- Se houver dependências transitivas (atributos não chave dependendo de outros atributos não chave), é necessário decompor a tabela em duas ou mais tabelas para remover essas dependências.

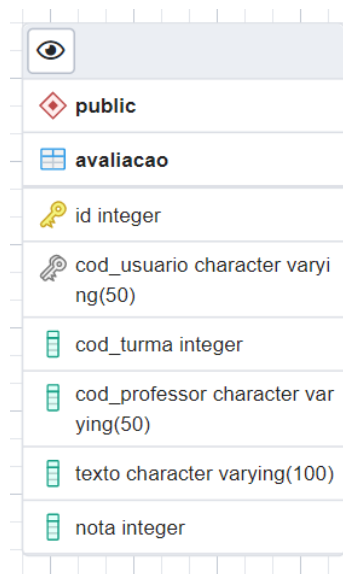
Foram analisadas as 3 seguintes tabelas:



The image shows a screenshot of a database management tool displaying the structure of a table named 'usuario' in the 'public' schema. The table has a primary key 'id' of type 'integer'. Other attributes include 'nome', 'email', 'senha', 'matricula', 'curso', 'isadm', and 'img', all of type 'character varying(50)' or 'bytea'.

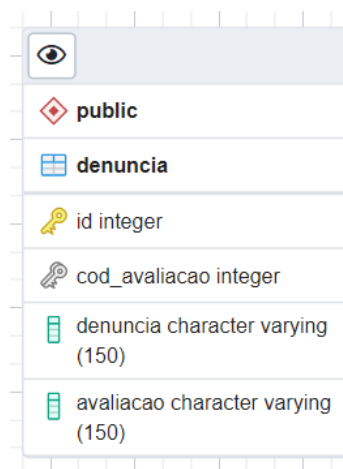
Schema	Table	Attribute	Data Type
public	usuario	id	integer
public	usuario	nome	character varying(50)
public	usuario	email	character varying(50)
public	usuario	senha	character varying(50)
public	usuario	matricula	character varying(50)
public	usuario	curso	character varying(50)
public	usuario	isadm	integer
public	usuario	img	bytea

Figure 3. Tabela de Usuário



public	
avaliacao	
id	integer
cod_usuario	character varying(50)
cod_turma	integer
cod_professor	character varying(50)
texto	character varying(100)
nota	integer

Figure 4. Tabela de Avaliação



public	
denuncia	
id	integer
cod_avaliacao	integer
denuncia	character varying(150)
avaliacao	character varying(150)

Figure 5. Tabela de Denúncia

Analisando as 3 tabelas com os passos citados anteriormente é possível chegar a conclusão de que as 3 tabelas estão adequadas às 3 primeiras formas normais.

5. Requisitos Diversos

5.1. CRUDs

”O CRUD deve ser feito para pelo menos 3 entidades;”

Estudantes

ID / Nome / Email / Senha / Matricula / Curso

1	a	a	a	a	2	{<memory at 0x0000016AD7405780>}
4	Nome	Email	Senha	Matricula	Curso	2 {<memory at 0x0000016AD7405A80>}
7	joao	joao@gmail.com	senha	123	Cic	2 {<memory at 0x0000016AD7405B40>}
9	joao2	joao2@gmail.com	senha	1234	Cic	1 {<memory at 0x0000016AD7405C00>}
10	1	1	1	1	2	{<memory at 0x0000016AD7405CC0>}
11	Nome1	Email1	Senha1	Matricula1	Curso1	2 {<memory at 0x0000016AD7405D80>}
12	Nome2	Email2	Senha2	Matricula2	Curso2	2 {<memory at 0x0000016AD7405E40>}
13	Nome3	Email3	Senha3	Matricula3	Curso3	2 {<memory at 0x0000016AD7405F00>}

Nome

Senha

Email

Matricula

Curso

☐ Usuário

☐ Admin

Criar Estudante

Remover Estudante

Atualizar Estudante

Limpar

Figure 6. CRUD de Estudante

Avaliações

Aluno / Período / Professor / Disciplina / Avaliação / Nota

2 a 2021 (JONATHAS FELIPE AIRES FERREIRA) (TECNOLOGIA SOCIAL E INOVAÇÃO) (n gostei) 1

Avaliação

Nota (1-5)

Criar avaliacao

Remover avaliacao

Atualizar avaliacao

Figure 7. CRUD de Avaliação

Denúncias

ID / Denúncia / Avaliação

9 2 (achei ofensivo) None

Denúncia

Avaliação da Denúncia

Criar denuncia

Atualizar denuncia

Ignorar denuncia

Remover Avaliação

Figure 8. CRUD de Denúncia

5.2. Inserts, Procedures e Views

”Deve ser fornecido código para inserção de pelo menos 3 linhas em cada uma das tabelas criadas;” ”Construção de pelo menos uma View;” ”Construção de pelo menos uma Procedure;”

```

INSERT INTO usuario (nome, email, senha, matricula, curso, isAdm, img) values ('Nome1', 'Email1', 'Senha1', 'Matricula1', 'Curso1', '2', '')
INSERT INTO usuario (nome, email, senha, matricula, curso, isAdm, img) values ('Nome2', 'Email2', 'Senha2', 'Matricula2', 'Curso2', '2', '')
INSERT INTO usuario (nome, email, senha, matricula, curso, isAdm, img) values ('Nome3', 'Email3', 'Senha3', 'Matricula3', 'Curso3', '2', '')

INSERT INTO avaliacao (cod_usuario, cod_turma, cod_professor, texto, nota) VALUES ('1', '1', '1', 'Avaliacao1', 5)
INSERT INTO avaliacao (cod_usuario, cod_turma, cod_professor, texto, nota) VALUES ('1', '1', '1', 'Avaliacao2', 5)
INSERT INTO avaliacao (cod_usuario, cod_turma, cod_professor, texto, nota) VALUES ('1', '1', '1', 'Avaliacao3', 5)

INSERT INTO denuncia (cod_avaliacao, denuncia) VALUES ('1', 'Denuncia1')
INSERT INTO denuncia (cod_avaliacao, denuncia) VALUES ('1', 'Denuncia2')
INSERT INTO denuncia (cod_avaliacao, denuncia) VALUES ('1', 'Denuncia3')

```

Figure 9. Inserts

Todos os inserts estão contidos nos arquivos 'departamentos.sql', 'disciplinas.sql', 'inserts.sql', 'professores.sql' e 'turmas.sql'.

```

CREATE or REPLACE PROCEDURE procedure_insert_user(
    nome, email, senha, matricula, curso, isAdm
) AS $$
begin
    INSERT INTO usuario(nome, email, senha, matricula, curso, isAdm, img) VALUES (nome, email, senha, matricula, curso, isAdm, img);
END;
$$
LANGUAGE PLPGSQL;

```

Figure 10. Procedure

A procedure feita pode ser encontrada no arquivo 'procedures.sql'

```

22 # VIEWS
23
24 self.cur.execute('CREATE or REPLACE VIEW view_turmas as select turma.id, turma.turma, turma.periodo, horario, vagas_ocupadas, total_vagas, local, disciplina.nome as "discipli
25

```

Figure 11. View

A view feita pode ser encontrada no arquivo dbpostgres.sql

6. Conclusão

O projeto de desenvolvimento do sistema de avaliação de professores e disciplinas apresentou uma solução abrangente e eficiente para auxiliar os estudantes na escolha de suas disciplinas, aproveitando os conhecimentos adquiridos na disciplina de Bancos de Dados.

A implementação do sistema permitiu que os estudantes se cadastrassem e compartilhassem avaliações sobre os professores e turmas de diferentes semestres. Além disso, foram incorporadas funcionalidades que garantem a transparência e a confiabilidade das avaliações, como a possibilidade de denunciar comentários ofensivos e a ação dos administradores em relação a essas denúncias.

O sistema também contemplou a capacidade dos estudantes de editar e excluir seus próprios comentários, fornecendo flexibilidade e permitindo a atualização das avaliações à medida que suas experiências acadêmicas evoluem.

O projeto evidenciou a importância do modelo lógico relacional na definição da estrutura das tabelas, atributos e relacionamentos, garantindo a integridade e a consistência dos dados. A aplicação das três primeiras formas normais contribuiu para a organização eficiente das informações, evitando redundâncias e dependências indesejadas.

Em resumo, o projeto do sistema de avaliação de professores e disciplinas proporcionou uma plataforma robusta e confiável para a comunidade estudantil. O desenvolvimento e a implementação desse sistema demonstraram a aplicação prática dos conhecimentos em bancos de dados, bem como a capacidade de projetar e criar soluções eficientes para problemas reais. Espera-se que esse sistema contribua para uma comunidade acadêmica mais informada, colaborativa e promova um ambiente de aprendizado aprimorado.

References

- [Postgres 2023] Postgres (2023). PostgreSQL. <https://www.postgresql.org/>. Acessado em 13 de julho de 2023.
- [Psycopg2 2023] Psycopg2 (2023). Psycopg2. <https://www.psycopg.org/docs/>. Acessado em 13 de julho de 2023.
- [Python 2023] Python (2023). Python. <https://www.python.org>. Acessado em 13 de julho de 2023.
- [Tkinter 2023] Tkinter (2023). Tkinter. <https://docs.python.org/3/library/tkinter.html>. Acessado em 13 de julho de 2023.