

**UNIDADE CURRICULAR:**

Sistema de Base de Dados I

**DOCENTE:**

Bruno Gaspar

**DATA:**

9 de junho de 2023

**RELATÓRIO: Simulação de uma Máquina Automática de vendas de snacks e bebidas**

**José Carlos nº 2011122**

**Elton Santos nº 2143221**

**Curso Técnico Superior Profissional em Tecnologia de Programação de Sistema de Informação**

**Ìndice**

Introdução

Descrição do Projeto

Modelagem do Banco de Dados

Entidades e Relacionamentos

Diagramas de Entidade-Relacionamento (ER)

Implementação do Banco de Dados

Criação do Banco de Dados e Tabelas

Inserção de Dados

Consultas e Manipulação de Dados

Uso da Notação de Crows Foot

Gestão de Referências e Restrições de Integridade

Considerações de Segurança

Desempenho e Escalabilidade

Conclusão

**Descrição do Projeto**

O projeto envolve a criação de um clone da plataforma Twitter/X, uma rede social onde os utilizadores podem compartilhar mensagens curtas. A plataforma deve permitir funcionalidades essenciais como registo de utilizadores, publicação de tweets, seguimento de outros utilizadores, e interação com conteúdo na forma de gostos e publicações de imagens. Além disso, haverá um backoffice para administração do sistema, permitindo a gestão de utilizadores e tweets.

A base de dados é essencial para armazenar e gerenciar todas as informações e atividades da plataforma. Aqui estão as principais tabelas : Tabela de Tweets (Tweets), Tabela de Seguidores (Seguidires), Tabela de Gostos (Likes), Tabela de Administração (Admin).

A base de dados permite operações como:

Registo e Login: Inserção e validação de utilizadores.

Publicação de Tweets: Inserção e recuperação de tweets para o feed.

Seguimento e Gostos: Gestão das relações de seguimento e gostos através de inserções e consultas.

Backoffice: Operações administrativas para gerenciar utilizadores e tweets.

Com essa estrutura e funcionalidades, a plataforma poderá atender aos requisitos, proporcionando uma boa experiência aos utilizadores e permitindo uma gestão eficiente por parte dos administradores.

1. **Introdução**

O objetivo deste projeto é criar uma base de dados para um clone do Twitter/X, que inclua funcionalidades essenciais como o registo e login de utilizadores, a publicação de tweets, o seguimento de outros utilizadores, gostos em tweets, e um backoffice para a gestão de utilizadores e tweets. Este relatório detalha a criação, implementação e funcionamento da base de dados, utilizando a notação de Crows Foot para a modelagem de entidades e relacionamentos.

**2. Descrição do Projeto**

Este projeto consiste em criar um sistema que permita aos utilizadores:

Registar-se e fazer login.

Publicar tweets com ou sem imagens.

Seguir outros utilizadores.

Gostar de tweets.

Visualizar um feed com os tweets dos utilizadores que seguem.

Utilizar um backoffice para a gestão de utilizadores e tweets, com diferentes níveis de permissão.

**3. Modelagem do Banco de Dados**

**3.1 Entidades e Relacionamentos**

As principais entidades no sistema são:

Users: Armazena informações sobre os utilizadores.

Tweets: Armazena os tweets publicados pelos utilizadores.

Seguidores: Armazena informações sobre quem segue quem.

Likes: Armazena informações sobre quais utilizadores gostaram de quais tweets.

Admin: Armazena informações sobre os administradores e seus níveis de permissão.

**3.2 Diagramas de Entidade-Relacionamento (ER)**

A notação de Crows Foot é usada para representar graficamente os relacionamentos entre as entidades. É amplamente utilizada em diagramas de entidade-relacionamento para ajudar a visualizar a estrutura e as restrições de um banco de dados.

Componentes da Notação de Crows Foot:

**Entidades**

Representadas por retângulos.

Cada entidade corresponde a uma tabela no banco de dados.

O nome da entidade é escrito dentro do retângulo.

**Atributos**

Listados dentro do retângulo da entidade.

Chaves primarias e estrangeiras devidamente identificadss.

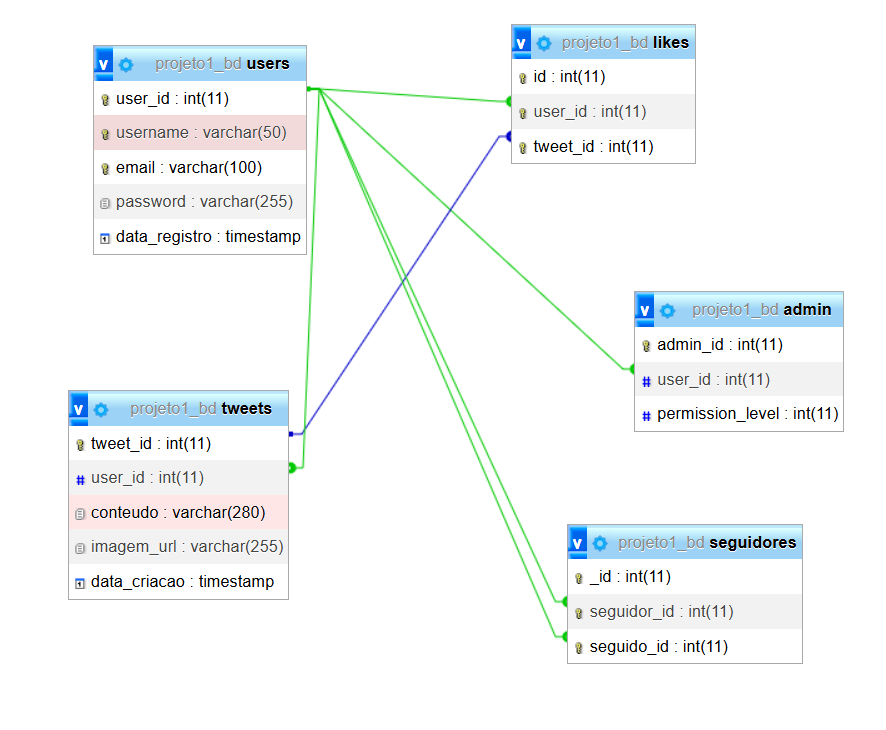
Restrições e tipo de atributos devidamente identificadas.

**Relacionamentos**

Representados por linhas que conectam duas entidades.

As linhas podem ter diferentes símbolos nas extremidades para indicar a cardinalidade e a participação.

1. **Diagrama Antigo**



Explicação do Diagrama

Users

A tabela Users tem user\_id como chave primária.

Relaciona-se com a tabela Tweets em um relacionamento 1

, indicando que um usuário pode criar muitos tweets.

Relaciona-se com a tabela Admin em um relacionamento 1

, indicando que um usuário pode ser administrador.

Tweets

A tabela Tweets tem tweet\_id como chave primária.

Contém user\_id como chave estrangeira, indicando a qual usuário o tweet pertence.

Relaciona-se com a tabela Likes em um relacionamento 1

, indicando que um tweet pode receber muitos likes.

Likes

A tabela Likes tem id como chave primária.

Contém user\_id e tweet\_id como chaves estrangeiras, indicando quais usuários gostaram de quais tweets.

Relaciona-se com a tabela Users em um relacionamento N:1 e com a tabela Tweets também em um relacionamento N:1.

Seguidores

A tabela Seguidores tem \_id como chave primária.

Contém seguidor\_id e seguido\_id como chaves estrangeiras, indicando quais usuários seguem outros usuários.

Relaciona-se com a tabela Users em dois relacionamentos N:1, indicando que um usuário pode seguir muitos usuários e ser seguido por muitos usuários.

Adminn

username: O nome de usuário do administrador, definido como VARCHAR(50) e UNIQUE, garantindo que cada nome de usuário seja único.

email: O endereço de e-mail do administrador, definido como VARCHAR(100) e UNIQUE, garantindo que cada e-mail seja único.

password: A senha do administrador, definida como VARCHAR(255), permitindo o armazenamento de senhas hashed.

permission\_level: Um nível de permissão para o administrador, definido como INT com um valor padrão de 1.

**6. Gestão de Referências e Restrições de Integridade**

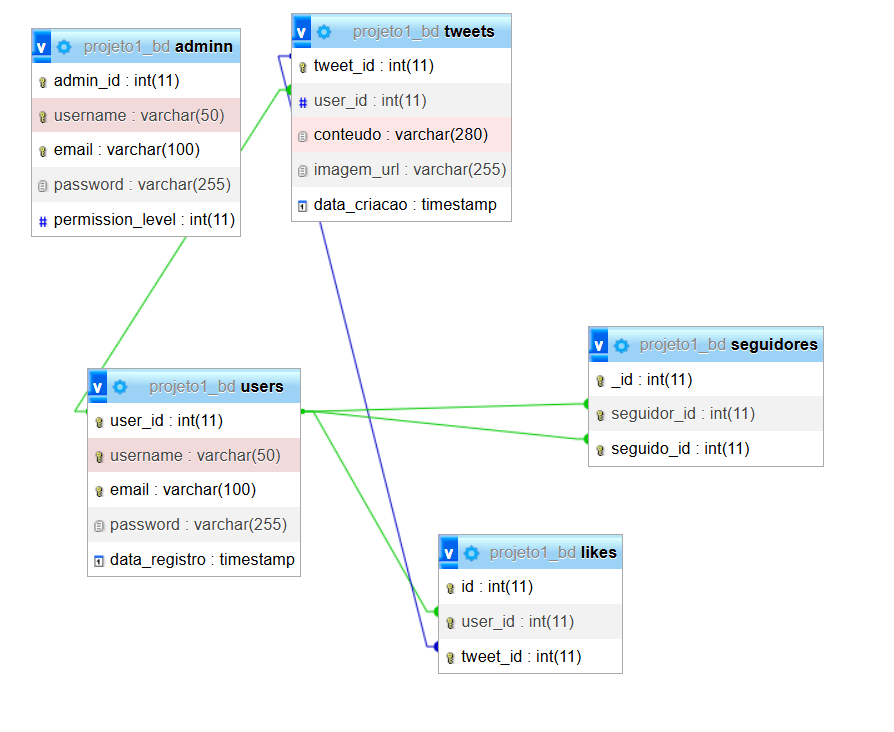
As chaves estrangeiras são usadas para garantir a integridade referencial, assegurando que os dados em uma tabela estejam consistentes com os dados em outra tabela.

As restrições de integridade garantem que:

Não podemos inserir um tweet com um user\_id que não exista na tabela Users.

Não podemos inserir um like com um tweet\_id que não exista na tabela Tweets.

**1.2 Diagrama atualizado**

****

**4. Implementação do Banco de Dados**

**4.1 Criação do Banco de Dados e Tabelas**

O banco de dados foi implementado utilizando o MySQL. Aqui está o código SQL para criar o banco de dados e suas tabelas:

Tabela Users

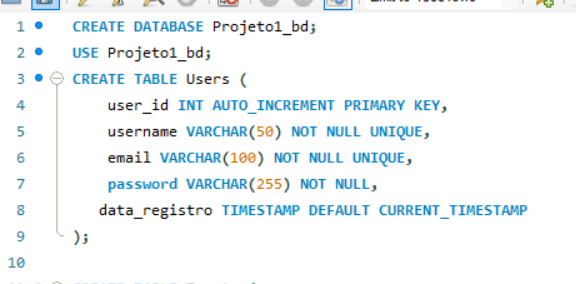


Tabela Tweets

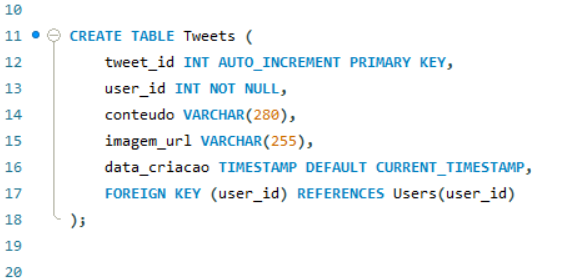


Tabela Seguidores

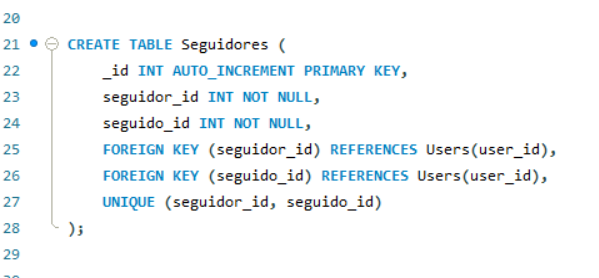


Tabela Likes

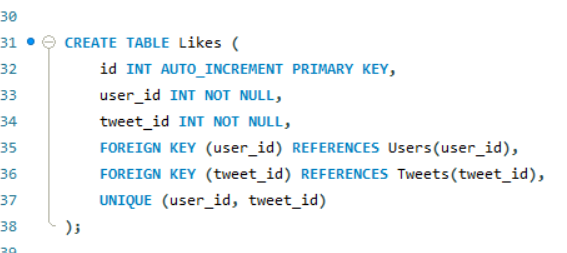


Tabela Admin antiga

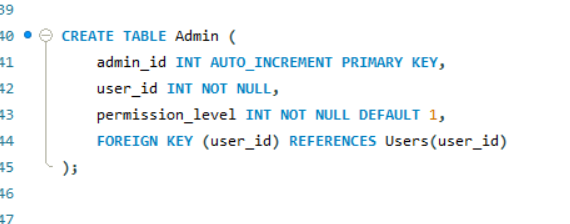
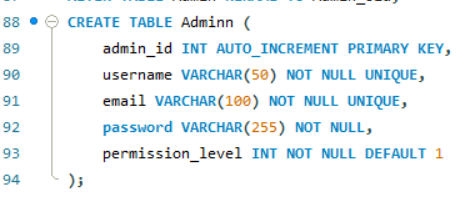


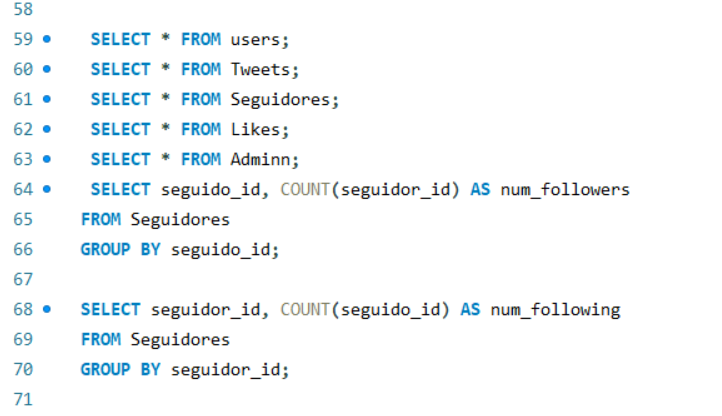
Tabela Adminn Atualizada



4.2 Inserção de Dados



4.3 Consulta e Manipulação de Dados





4.4 Inserção na nova tabela Admin



**7. Considerações de Segurança**

A segurança dos dados é uma consideração crítica. Assegure-se de:

Criptografar senhas antes de armazená-las.

Implementar controles de acesso para diferentes níveis de permissão.

Utilizar consultas parametrizadas para evitar ataques de SQL Injection.

**8. Desempenho e Escalabilidade**

Para garantir o bom desempenho e escalabilidade do sistema:

Indexação: Crie índices em colunas frequentemente utilizadas em consultas, como user\_id e tweet\_id.

Particionamento: Se o volume de dados for muito grande, considere particionar tabelas.

Cache: Utilize técnicas de cache para reduzir a carga no banco de dados.

**9. Conclusão**

Este relatório detalha a criação e operação de uma base de dados para um clone do Twitter/X, abordando desde a modelagem inicial até a implementação prática. A aplicação das práticas de integridade referencial, uso da notação de Crows Foot para modelagem de dados, e considerações de desempenho e segurança são fundamentais para o sucesso deste projeto.Esta documentação deve servir como um guia compreensivo para a criação e gestão de uma base de dados robusta e eficiente, cumprindo todos os requisitos funcionais propostos.

**Comandos**

**CREATE DATABASE Projeto1\_bd;**

**USE Projeto1\_bd;**

**CREATE TABLE Users (**

**user\_id INT AUTO\_INCREMENT PRIMARY KEY,**

**username VARCHAR(50) NOT NULL UNIQUE,**

**email VARCHAR(100) NOT NULL UNIQUE,**

**password VARCHAR(255) NOT NULL,**

**data\_registro TIMESTAMP DEFAULT CURRENT\_TIMESTAMP**

**);**

**CREATE TABLE Tweets (**

**tweet\_id INT AUTO\_INCREMENT PRIMARY KEY,**

**user\_id INT NOT NULL,**

**conteudo VARCHAR(280),**

**imagem\_url VARCHAR(255),**

**data\_criacao TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,**

**FOREIGN KEY (user\_id) REFERENCES Users(user\_id)**

**);**

**CREATE TABLE Seguidores (**

**\_id INT AUTO\_INCREMENT PRIMARY KEY,**

**seguidor\_id INT NOT NULL,**

**seguido\_id INT NOT NULL,**

**FOREIGN KEY (seguidor\_id) REFERENCES Users(user\_id),**

**FOREIGN KEY (seguido\_id) REFERENCES Users(user\_id),**

**UNIQUE (seguidor\_id, seguido\_id)**

**);**

**CREATE TABLE Likes (**

**id INT AUTO\_INCREMENT PRIMARY KEY,**

**user\_id INT NOT NULL,**

**tweet\_id INT NOT NULL,**

**FOREIGN KEY (user\_id) REFERENCES Users(user\_id),**

**FOREIGN KEY (tweet\_id) REFERENCES Tweets(tweet\_id),**

**UNIQUE (user\_id, tweet\_id)**

**);**

**CREATE TABLE Admin (**

**admin\_id INT AUTO\_INCREMENT PRIMARY KEY,**

**user\_id INT NOT NULL,**

**permission\_level INT NOT NULL DEFAULT 1,**

**FOREIGN KEY (user\_id) REFERENCES Users(user\_id)**

**);**

**INSERT INTO Users (username, email, password) VALUES ('Mari', 'mari2@gamil.com', 'password2'),('Carlos', 'carlos1@gmail.com', 'password1');**

**INSERT INTO Tweets (user\_id, conteudo, imagem\_url) VALUES (1, 'Hello, this is my first tweet!', NULL);**

**INSERT INTO Tweets (user\_id, conteudo, imagem\_url) VALUES (2, 'Carro dos sonhos.', '54c02cd28bba57f81d73d11b4fd8af20.jpg'), (3, 'hora de fazer acontecer.', NULL);**

**INSERT INTO Seguidores (seguidor\_id, seguido\_id) VALUES (1, 2), (2, 3),**

**(3, 1), (2, 1), (3,2);**

**INSERT INTO Likes (user\_id, tweet\_id) VALUES (1, 1),(2, 2),**

**(3, 3);**

**INSERT INTO Admin (user\_id, permission\_level) VALUES (1, 1);**

**SELECT \* FROM users;**

**SELECT \* FROM Tweets;**

**SELECT \* FROM Seguidores;**

**SELECT \* FROM Likes;**

**SELECT \* FROM Admin;**

**SELECT seguido\_id, COUNT(seguidor\_id) AS num\_followers**

**FROM Seguidores**

**GROUP BY seguido\_id;**

**SELECT seguidor\_id, COUNT(seguido\_id) AS num\_following**

**FROM Seguidores**

**GROUP BY seguidor\_id;**

**UPDATE Users SET username = 'João Paulo' WHERE user\_id = 1;**

**UPDATE Tweets SET conteudo = 'Vamos atualizar com o tweet' WHERE tweet\_id = 1;**

**UPDATE Seguidores SET seguido\_id = 2 WHERE \_id = 1;**

**UPDATE Likes SET tweet\_id = 2 WHERE id = 1;**

**UPDATE Admin SET permission\_level = 5 WHERE admin\_id = 2;**

**DELETE FROM Admin WHERE user\_id = 3;**

**DELETE FROM Likes WHERE user\_id = 1;**

**SET SQL\_SAFE\_UPDATES = 0;**

**DELETE FROM Likes WHERE tweet\_id IN (SELECT tweet\_id FROM Tweets WHERE user\_id = 1);**

**DELETE FROM Seguidores WHERE seguidor\_id = 1;**

**DELETE FROM Seguidores WHERE seguido\_id = 1;**

**DELETE FROM Tweets WHERE user\_id = 1;**

**SET SQL\_SAFE\_UPDATES = 1;**

**DELETE FROM Users WHERE user\_id = 1;**

**DROP TABLE IF EXISTS Admin;**

**-- Passo 2: Criar a nova tabela Admin**

**CREATE TABLE Admin (**

**admin\_id INT AUTO\_INCREMENT PRIMARY KEY,**

**username VARCHAR(50) NOT NULL UNIQUE,**

**email VARCHAR(100) NOT NULL UNIQUE,**

**password VARCHAR(255) NOT NULL,**

**permission\_level INT NOT NULL DEFAULT 1**

**);**

**INSERT INTO Users (user\_id, username, email, password) VALUES (1, 'admin\_user', 'admin@gmail.com', 'password');**