UNIVERSIDADE CATÓLICA DE BRASÍLIA

DISCIPLINA: LABORATÓRIO DE BANCO DE DADOS

PROFESSOR: JOÃO ROBSON

ALUNOS: LETÍCIA CARVALHO RIBEIRO, VICTOR ALEXANDRE PRAÇA, JOÃO ANTHONY, BRUNO RAMIRO

PROJETO FINAL – REDE SOCIAL

1. INTRODUÇÃO

Este relatório apresenta o desenvolvimento de um banco de dados relacional para uma rede social, atendendo aos requisitos estabelecidos na disciplina Laboratório de Banco de Dados. O objetivo é demonstrar as decisões de modelagem, a estruturação lógica e física do banco, bem como a implementação de dados e consultas essenciais.

2. ANÁLISE DOS REQUISITOS

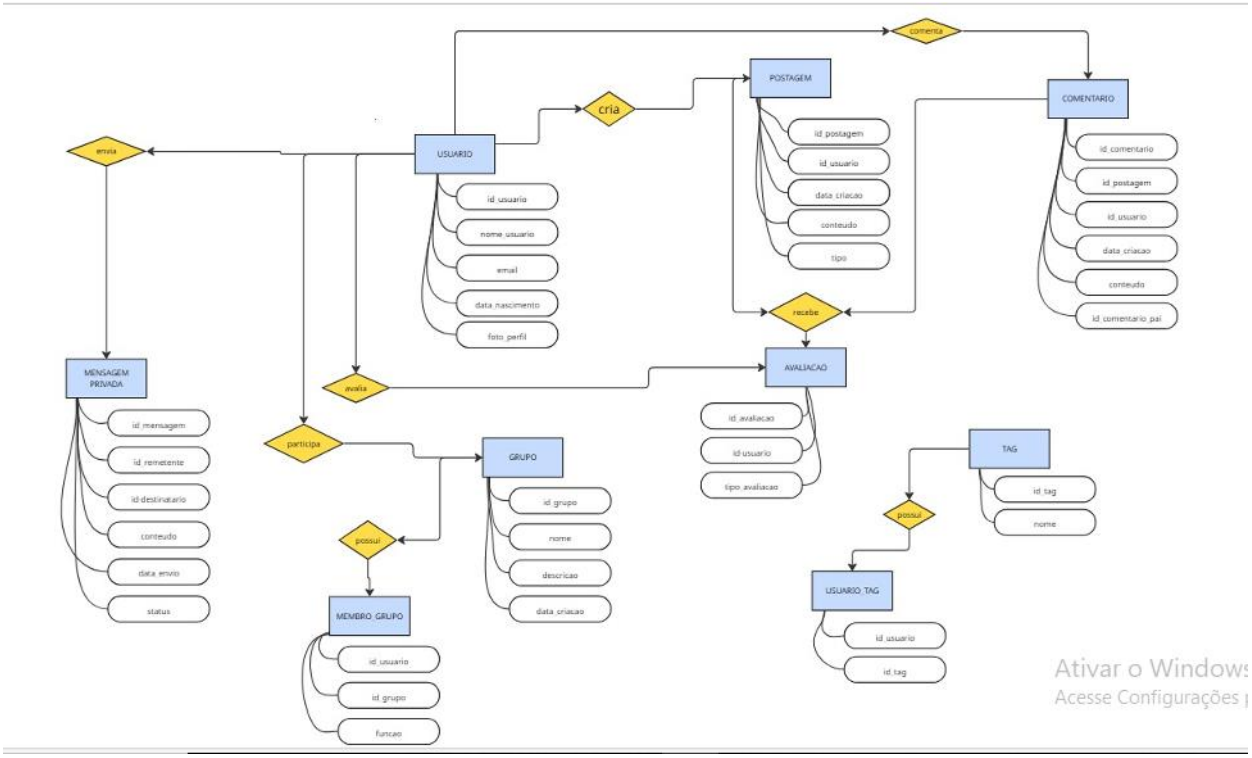
O projeto propôs a modelagem e implementação de um banco de dados para uma rede social, com funcionalidades típicas como cadastro de usuários, postagens, comentários, avaliações, grupos, mensagens privadas e tags.

A seguir, destacamos como cada uma dessas funcionalidades foi interpretada e modelada:

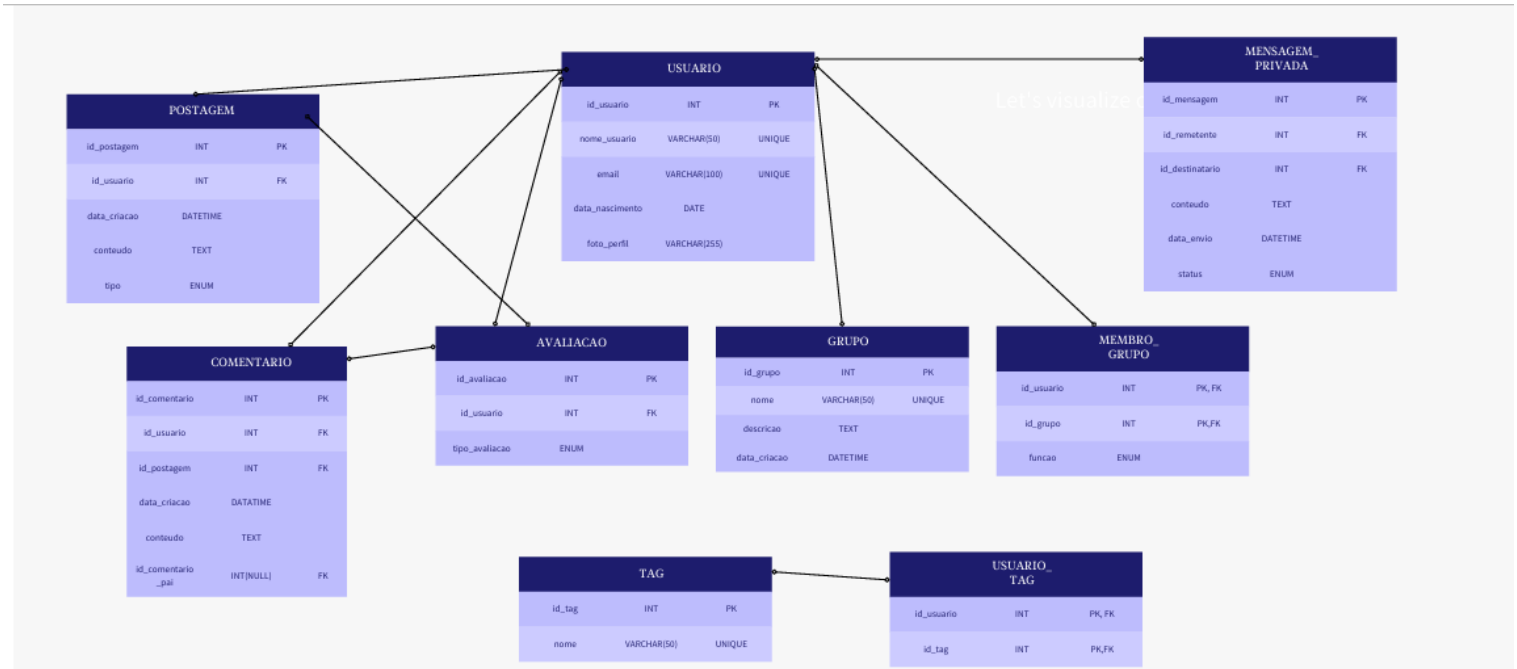
- **Usuários:** Foi criada uma tabela `usuário` com dados essenciais (nome, e-mail, data de nascimento e foto de perfil). O campo `nome_usuario` e o `e-mail` foram definidos como únicos para evitar duplicidade.
- **Postagens:** A tabela `postagem` relaciona-se com o usuário (1:N) e contém data, conteúdo e tipo (texto, imagem, vídeo).
- **Comentários:** Os comentários foram modelados com a possibilidade de responder a outros comentários, através do campo `id_comentario_pai`, criando um relacionamento recursivo.
- **Avaliações:** A tabela `avaliação` permite ao usuário avaliar postagens ou comentários, com uma flag de tipo (`positivo` ou `negativo`). Avaliações podem apontar para postagem ou comentário, usando FK's opcionais.
- **Grupos:** Modelamos os grupos com `nome`, `descrição` e `data_criacao`. O relacionamento N:N entre usuários e grupos foi resolvido pela tabela `membro_grupo`, com a coluna `função` indicando se é membro ou administrador.
- **Mensagens privadas:** Foram modeladas com remetente e destinatário, além de status (`enviada`, `recebida`, `lida`) e histórico.
- **Tags:** A relação N:N entre usuários e tags foi modelada com a tabela `usuario_tag`. Também foi criada uma trigger para limitar cada usuário a no máximo 5 tags, conforme exigência do enunciado.

3. MODELAGEM

3.1 MODELAGEM CONCEITUAL



3.2 MODELO LÓGICO



3.3 MODELO FÍSICO – PARTE INICIAL

```
CREATE TABLE usuario (Add commentMore actions
    id_usuario INT AUTO_INCREMENT PRIMARY KEY,
    nome_usuario VARCHAR(50) UNIQUE NOT NULL,
    email VARCHAR(100) UNIQUE NOT NULL,
    data_nascimento DATE NOT NULL,
    foto_perfil VARCHAR(255)
);
CREATE TABLE postagem (
    id_postagem INT AUTO_INCREMENT PRIMARY KEY,
    id_usuario INT NOT NULL,
    data_criacao DATETIME DEFAULT CURRENT_TIMESTAMP,
    conteudo TEXT NOT NULL,
    tipo ENUM('texto', 'imagem', 'video') NOT NULL,
    FOREIGN KEY (id_usuario) REFERENCES usuario(id_usuario)
);
CREATE TABLE comentario (
    id_comentario INT AUTO_INCREMENT PRIMARY KEY,
    id_postagem INT NOT NULL,
    id_usuario INT NOT NULL,
    data_criacao DATETIME DEFAULT CURRENT_TIMESTAMP,
    conteudo TEXT NOT NULL,
    id_comentario_pai INT NULL,
    FOREIGN KEY (id_postagem) REFERENCES postagem(id_postagem),
    FOREIGN KEY (id_usuario) REFERENCES usuario(id_usuario),
    FOREIGN KEY (id_comentario_pai) REFERENCES comentario(id_comentario)
);
CREATE TABLE avaliacao (
    id_avaliacao INT AUTO_INCREMENT PRIMARY KEY,
    id_usuario INT NOT NULL,
```

4. IMPLEMENTAÇÃO

A implementação do banco de dados foi realizada utilizando a linguagem SQL no MySQL Workbench. O projeto foi organizado em diferentes arquivos, conforme os critérios estabelecidos no enunciado da atividade. Inicialmente, foi criado o script criar_tabelas.sql, responsável por definir a estrutura do banco de dados, incluindo a criação das tabelas, definição de tipos de dados, chaves primárias, chaves estrangeiras e restrições. Todas as tabelas foram elaboradas conforme o modelo lógico apresentado na seção anterior.

Em seguida, foi desenvolvido o script inserir_dados.sql, que contém instruções de inserção (INSERT) para alimentar o banco de dados com dados fictícios. Foram adicionados registros coerentes, com atenção à integridade referencial e à consistência entre as tabelas. Cada tabela recebeu pelo menos dez registros para permitir a execução de testes e simulações de uso do sistema.

Para ilustrar a inserção de dados, segue um exemplo:

```
INSERT INTO usuario (nome_usuario, email, data_nascimento,
foto_perfil)
VALUES ('ana123', 'ana@gmail.com', '2000-01-15', 'ana.jpg');
```

Além disso, o arquivo `consultas.sql` reúne cinco consultas relevantes ao funcionamento da rede social. Essas consultas foram projetadas para responder a necessidades reais do sistema, como: listar postagens de um usuário, buscar comentários de uma postagem específica, contar o número de postagens por usuário, entre outras. Como exemplo, segue uma das consultas implementadas:

```
SELECT u.nome_usuario, COUNT(p.id_postagem) AS
total_postagens
FROM usuario u
LEFT JOIN postagem p ON u.id_usuario = p.id_usuario
GROUP BY u.id_usuario;
```

Por fim, foi criada uma `trigger`, incluída no script `trigger.sql`, com o objetivo de garantir a aplicação de uma regra de negócio: limitar cada usuário a um máximo de cinco tags. Essa trigger é acionada antes da inserção em `usuario_tag` e impede que sejam atribuídas mais de cinco tags a um mesmo usuário, garantindo a conformidade com os requisitos da aplicação.

5. CONCLUSÃO

O desenvolvimento deste projeto proporcionou a aplicação prática dos conhecimentos adquiridos ao longo da disciplina de Laboratório de Banco de Dados, envolvendo todas as etapas do ciclo de modelagem e implementação de um banco relacional. Através da análise dos requisitos de uma rede social, foi possível estruturar um modelo conceitual claro e coerente, seguido por um modelo lógico bem normalizado e um modelo físico funcional. A modelagem contemplou aspectos essenciais como relacionamentos N:N, relacionamentos recursivos, integridade referencial e uso de `ENUM` para campos com valores

controlados. A criação das tabelas e os dados inseridos permitiram simular o funcionamento real da aplicação, com interações entre usuários, postagens, comentários, grupos, mensagens e tags. As consultas elaboradas evidenciam a viabilidade do modelo para atender a diferentes tipos de demandas informacionais da aplicação. A implementação de uma trigger também demonstrou a possibilidade de aplicar regras de negócio diretamente no banco, garantindo controle sobre a lógica do sistema. Com isso, o projeto atingiu os objetivos propostos, validando as decisões tomadas em cada etapa da modelagem e demonstrando a viabilidade técnica do banco de dados desenvolvido para uma rede social funcional e escalável.