

# Entrega Projeto Interdisciplinar

## Lideranças empáticas

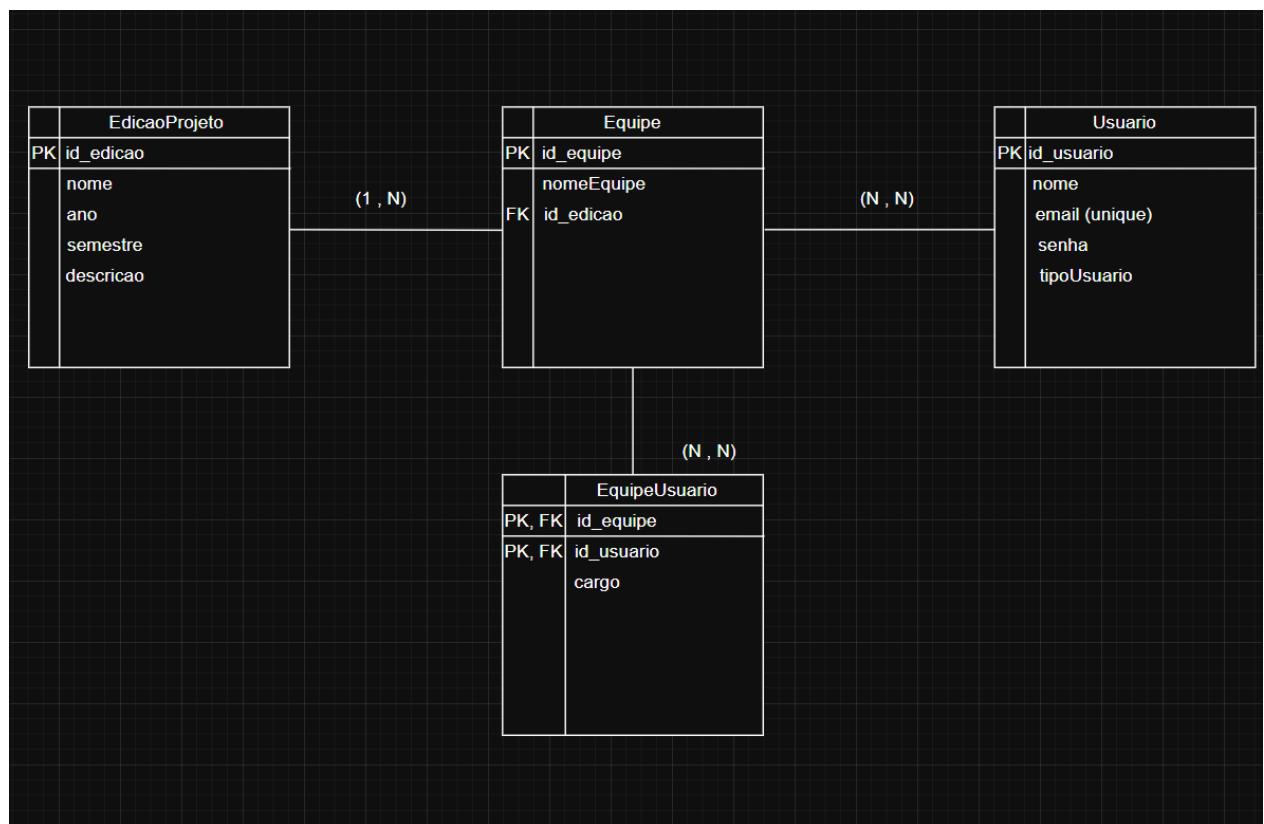
Professor: David de Oliveira Lemes

### Objetivo do Banco de Dados

O objetivo deste banco de dados é gerenciar as edições do site da Lideranças empáticas, permitindo controlar equipes, usuários (como alunos, professores e orientadores) e suas funções dentro das equipes.

### Scripts SQL

### DER do script



### Esquema Relacional

EdicaoProjeto(id\_edicao PK, nome, ano, semestre, descricao)

Usuario(id\_usuario PK, nome, email UNIQUE, senha, tipoUsuario)

Equipe(id\_equipe PK, nomeEquipe, id\_edicao FK → EdicaoProjeto(id\_edicao))

EquipeUsuario(id\_equipe PK FK → Equipe(id\_equipe),  
id\_usuario PK FK → Usuario(id\_usuario),  
cargo)

## Relacionamentos e cardinalidades

Relacionamento	Tabelas Envolvidas	Tipo	Cardinalidade
Edição – Equipe	EdicaoProjeto ↔ Equipe	1:N	Uma edição pode ter várias equipes
Equipe – Usuário	Equipe ↔ Usuario (via EquipeUsuario)	N:M	Um usuário pode participar de várias equipes e uma equipe pode ter vários usuários

## Criação do Banco de Dados

```
CREATE DATABASE IF NOT EXISTS ProjetoAcademico;  
USE ProjetoAcademico;
```

## Criação das Tabelas

```
CREATE TABLE IF NOT EXISTS EdicaoProjeto (  
  id_edicao INT AUTO_INCREMENT PRIMARY KEY,  
  nome VARCHAR(100) NOT NULL,  
  ano INT NOT NULL,  
  semestre INT NOT NULL,  
  descricao TEXT  
);  
  
CREATE TABLE IF NOT EXISTS Usuario (  
  id_usuario INT AUTO_INCREMENT PRIMARY KEY,  
  nome VARCHAR(100) NOT NULL,  
  email VARCHAR(100) NOT NULL UNIQUE,  
  senha VARCHAR(255) NOT NULL,  
  tipoUsuario VARCHAR(50) NOT NULL  
);  
  
CREATE TABLE IF NOT EXISTS Equipe (  
  id_equipe INT AUTO_INCREMENT PRIMARY KEY,  
  id_edicao INT,  
  nomeEquipe VARCHAR(100) NOT NULL,  
  FOREIGN KEY (id_edicao) REFERENCES EdicaoProjeto(id_edicao)  
);  
  
CREATE TABLE IF NOT EXISTS EquipeUsuario (  
  id_equipe INT NOT NULL,  
  id_usuario INT NOT NULL,  
  cargo VARCHAR(100),  
  PRIMARY KEY (id_equipe, id_usuario),  
  FOREIGN KEY (id_equipe) REFERENCES Equipe(id_equipe),  
  FOREIGN KEY (id_usuario) REFERENCES Usuario(id_usuario)  
);
```

## Inserção de Dados

```
INSERT INTO EdicaoProjeto (nome, ano, semestre, descricao) VALUES ('Projeto Integrador I', 2024, 1, 'Primeira edição do projeto integrador.'), ('Projeto Integrador II', 2024, 2, 'Continuação dos projetos.'), ('Hackathon Acadêmico', 2025, 1, 'Edição de inovação e tecnologia.'), ('Desafio de Software', 2025, 1, 'Competição de desenvolvimento.'), ('TCC Sistemas', 2025, 2, 'Trabalhos de conclusão de curso.');
```

```
INSERT INTO Usuario (nome, email, senha, tipoUsuario) VALUES ('Ana Silva', 'ana@email.com', '123456', 'Aluno'), ('Bruno Costa', 'bruno@email.com', 'abc123', 'Aluno'), ('Carlos Lima', 'carlos@email.com', 'senha123', 'Professor'), ('Daniela Souza', 'daniela@email.com', 'senha456', 'Coordenador'), ('Eduardo Ramos', 'eduardo@email.com', 'teste789', 'Aluno');
```

```
INSERT INTO Equipe (id_edicao, nomeEquipe) VALUES (1, 'Equipe Alpha'), (1, 'Equipe Beta'), (2, 'Equipe Gamma'), (3, 'Equipe Delta'), (4, 'Equipe Omega');
```

```
INSERT INTO EquipeUsuario (id_equipe, id_usuario, cargo) VALUES (1, 1, 'Líder'), (1, 2, 'Membro'), (2, 3, 'Orientador'), (3, 4, 'Coordenador'), (4, 5, 'Participante');
```

## Consultas Relevantes e Explicações

### Listar equipes com suas edições

```
SELECT e.nomeEquipe, ep.nome AS edicao, ep.ano, ep.semestre
FROM Equipe e
INNER JOIN EdicaoProjeto ep ON e.id_edicao = ep.id_edicao;
```

Mostra o nome das equipes e a edição à qual pertencem.

### Contar integrantes por equipe

```
SELECT e.nomeEquipe, COUNT(eu.id_usuario) AS
total_integrantes FROM Equipe e
LEFT JOIN EquipeUsuario eu ON e.id_equipe = eu.id_equipe
GROUP BY e.nomeEquipe;
```

Mostra quantos usuários participam de cada equipe.

### Listar usuários e suas equipes

```
SELECT u.nome AS usuario, e.nomeEquipe AS equipe, eu.cargo
FROM Usuario u
LEFT JOIN EquipeUsuario eu ON u.id_usuario = eu.id_usuario
LEFT JOIN Equipe e ON eu.id_equipe = e.id_equipe;
```

Mostra todos os usuários e as equipes em que estão (ou NULL se não participam).

### Contar equipes por edição

```
SELECT ep.nome AS edicao, COUNT(e.id_equipe) AS total_equipes
FROM EdicaoProjeto ep
```

```
LEFT JOIN Equipe e ON ep.id_edicao = e.id_edicao  
GROUP BY ep.nome;
```

Mostra quantas equipes existem em cada edição de projeto.

## Quantidade de usuários por tipo

```
SELECT tipoUsuario, COUNT(*) AS total  
FROM Usuario  
GROUP BY tipoUsuario;
```

Mostra quantos usuários há em cada categoria (alunos, professores, etc.).