

# 1 Projeto Físico do Banco de Dados

O projeto físico do banco de dados do sistema *UserHistory Hub* foi implementado utilizando PostgreSQL, com foco em desempenho, normalização, segurança de dados e manutenibilidade. A modelagem contempla as principais entidades do sistema: usuários, projetos, mensagens e relações entre esses elementos. A seguir, apresenta-se a descrição detalhada de cada tabela, tipos de dados, chaves primárias, chaves estrangeiras e relacionamentos.

## 1.1 Visão Geral das Tabelas

O banco de dados foi estruturado com as seguintes tabelas principais:

- **user**: armazena as credenciais dos usuários.
- **project**: armazena os projetos registrados no sistema.
- **message**: armazena mensagens enviadas pelos usuários.
- **auth**: controla permissões e níveis de acesso dos usuários.
- **userproject**: relacionamento entre usuários e projetos.
- **usermessage**: relacionamento entre usuários e mensagens.
- **messageproject**: relacionamento entre mensagens e projetos.

## 1.2 Descrição das Tabelas

### 1.2.1 Tabela: user

- **id**: *bigserial*, chave primária.
- **username**: *varchar(255)*, único, obrigatório.
- **password**: *varchar(255)*, obrigatório.

Esta tabela armazena os dados de autenticação dos usuários. O índice único em *username* garante que não existam nomes duplicados.

### 1.2.2 Tabela: auth

- **authkeys**: *varchar*, chave primária.
- **level**: *int*, nível de permissão.
- **userid**: *int8*, chave estrangeira implícita referente ao usuário.

Responsável por armazenar tokens ou chaves de autenticação e seus respectivos níveis de acesso. A restrição *auth\_unique* impede que um usuário possua mais de uma chave ativa.

### 1.2.3 Tabela: project

- **id**: *bigserial*, chave primária.
- **description**: *varchar*, opcional.

Tabela destinada ao cadastro de projetos. Possui chave primária autoincrementada.

### 1.2.4 Tabela: message

- **id**: *int8*, chave primária.
- **message**: *varchar*, obrigatório.
- **creator**: *int8*, referência ao usuário.
- **type**: *varchar*, tipo da mensagem.

Armazena mensagens criadas pelos usuários. Possui uma referência lógica para a tabela *user* via campo *creator*.

### 1.2.5 Tabela: userproject

- **userid**: *int8*, referência a **user**.
- **projectid**: *int8*, referência a **project**.

Responsável por representar o relacionamento N:N entre usuários e projetos. Cada linha indica que um usuário específico participa de determinado projeto.

### 1.2.6 Tabela: usermessage

- **user\_id**: *int8*, referência a **user**.
- **message\_id**: *int8*, referência a **message**.

Representa o relacionamento N:N entre usuários e mensagens.

### 1.2.7 Tabela: messageproject

- **messageid**: *int8*, referência a **message**.
- **projectid**: *int8*, referência a **project**.

Tabela que mapeia o relacionamento N:N entre mensagens e projetos.

### 1.3 Relacionamentos entre as Entidades

A modelagem apresenta os seguintes relacionamentos:

- **user 1:N message:** um usuário pode criar várias mensagens.
- **user N:N project:** usuários participam de múltiplos projetos.
- **message N:N project:** mensagens podem estar associadas a mais de um projeto.
- **user N:N message:** usuários podem estar associados a mensagens de acordo com contexto.
- **auth 1:1 user:** cada usuário possui uma chave de autenticação exclusiva.

### 1.4 Observações sobre Normalização

O banco foi projetado seguindo os princípios da **Terceira Forma Normal (3FN)**, garantindo:

- Ausência de redundância de dados.
- Integridade referencial.
- Separação de responsabilidades entre tabelas.

### 1.5 Conclusão do Projeto Físico

A estrutura criada permite alta flexibilidade, possibilitando expansão futura sem violar os princípios de normalização. A presença de tabelas de relacionamento N:N oferece adaptabilidade, essencial para sistemas de interação entre usuários, mensagens e projetos.