

# **Centro Universitário UNISATC**

Engenharia de Software 3a fase – Banco de Dados II – Prof. Luciano B. Fernandes

## **TRABALHO FINAL COM BASE EM METODOLOGIAS ATIVAS DE APRENDIZAGEM**

**Projeto de banco de dados para um sistema de Gestão Eletrônica de Documentos (GED)**

Lucas Perito Lopes - @llucaslopes

Nicolas Andrade de Freitas - @NicolasFreitas1

Vitor Valcanaia Rosendo Martins - @Valcanaia282

Criciúma, 24/06/2024

## URL do projeto no GitHub

<https://github.com/NicolasFreitas1/trabalho-banco-2>

The screenshot shows the GitHub repository page for 'trabalho-banco-2' by user 'NicolasFreitas1'. The repository is public and has 1 watch, 0 forks, and 0 stars. The main branch is 'main' with 2 branches and 0 tags. The repository contains a README.md file and a commit history table.

Commit	Message	Time
d3d1623	Merge branch 'main' of github.com:NicolasFreitas1/trabalho-banco-2	30 minutes ago
	Adicionado documentações atualizadas	40 minutes ago
	Fix prisma	30 minutes ago
	Adicionado documentações atualizadas	40 minutes ago
	Adicionado documentações atualizadas	40 minutes ago
	Adicionado documentações atualizadas	40 minutes ago

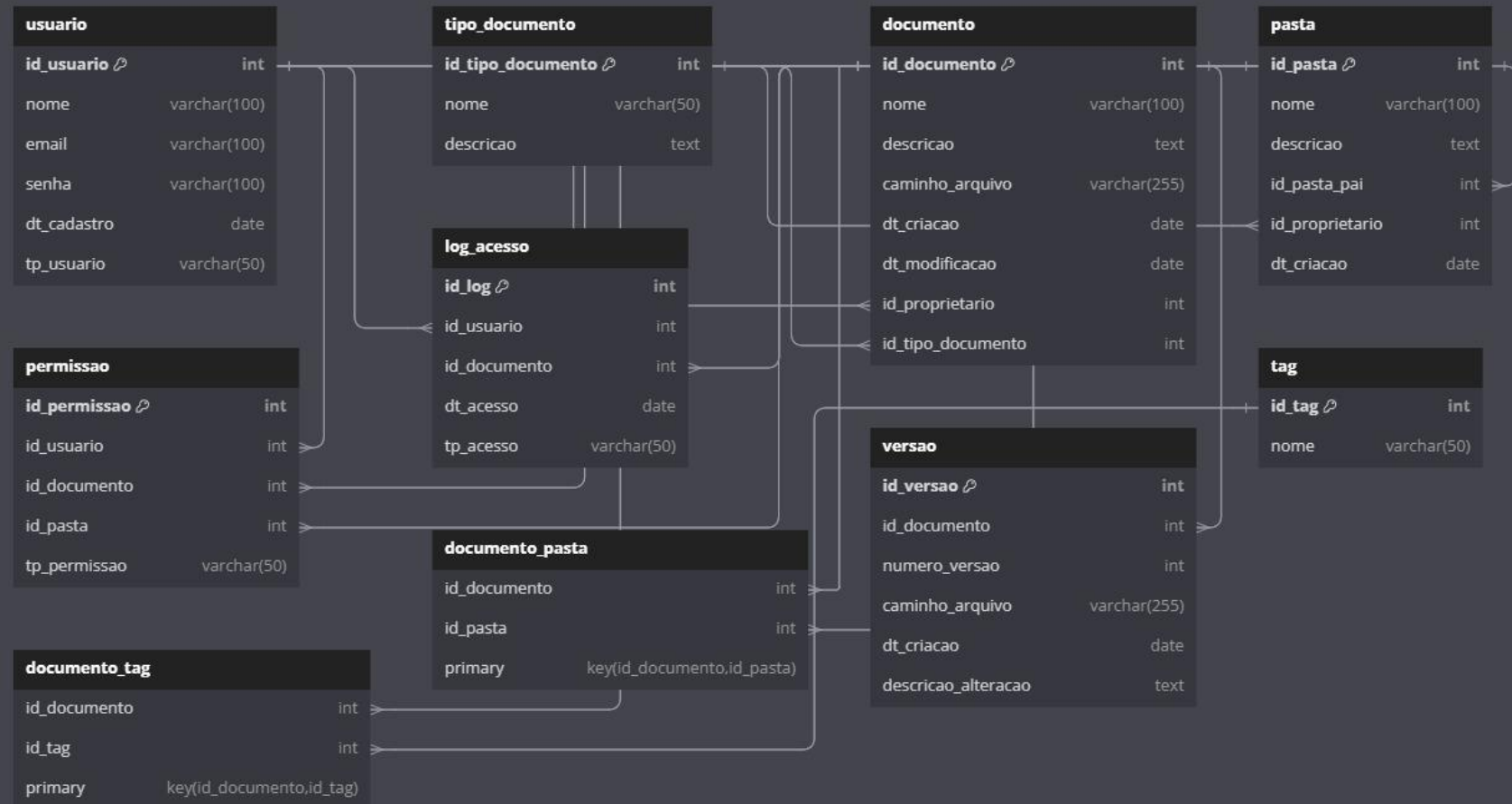
The README file is titled 'Projeto Sistema de Atividades Especiais' and lists the following integrants:

- Lucas Perito Lopes - @lucaslopes
- Nicolas Andrade de Freitas - @NicolasFreitas1
- Vitor Valcanaia Rosendo MartinsVitor Valcanaia - @Valcanaia282
- ...

The README also mentions the use of [dbdiagram.io](https://dbdiagram.io) for creating the physical database model and provides a link to the source file: [Modelo Fisico](#).

The right sidebar shows the repository's activity, including a list of contributors: NicolasFreitas1, Valcanaia282, lllucaslopes, and NicolasFreitas123.

## Modelo ER Físico



## Dicionário de Dados

Tabela	Documento					
Descrição	Tabela responsável pelo registro de documentos					
Atributos						
Nome da Coluna	Tipo do Dado	Valor min e max	Nulidade	PK	FK	Descrição
id_documento	serial	1 – sem limite	NOT NULL	X		Código de identificador da avaliação
nome	varchar(100)	10 - 100	NOT NULL			Nome da avaliação
descricao	text	1 – sem limite	NOT NULL			Descrição do documento
caminho_arquivo	varchar(255)	1 - 255	NOT NULL			Caminho do arquivo
dt_criacao	date		NOT NULL			Data da criação
dt_modificacao	date		NOT NULL			Data da última modificação
id_proprietario	serial	1 – sem limite	NOT NULL		X	Id do proprietário
id_tipo_documento	serial	1 – sem limite	NOT NULL		X	id do tipo do documento
Índice						
Nome do índice	Clustered	NonClustered	Unique	Colunas		
Idx_dt_avaliacao	X		X	cd_avaliacao, dt_avaliacao		
Idx_dt_avaliacao		X		cd_avaliacao, dt_avaliacao		
Idx_dt_avaliacao		X	X	cd_avaliacao, dt_avaliacao		
Idx_dt_avaliacao		X		cd_avaliacao, dt_avaliacao		

<b>Tabela</b>	documento_tag
<b>Descrição</b>	Tabela responsável por vincular a tag a um documento

Atributos						
Nome da Coluna	Tipo do Dado	Valor min e max	Nulidade	PK	FK	Descrição
id_documento	serial	1 – sem limite	NOT NULL	X		Id do documento
id_tag	serial	1 – sem limite	NOT NULL		X	Id da tag
Índice						
Nome do índice	Clustered	NonClustered	Unique	Colunas		
Idx_dt_avaliacao	X		X	cd_avaliacao, dt_avaliacao		
Idx_dt_avaliacao		X		cd_avaliacao, dt_avaliacao		
Idx_dt_avaliacao		X	X	cd_avaliacao, dt_avaliacao		
Idx_dt_avaliacao		X		cd_avaliacao, dt_avaliacao		

Tabela	log_acesso					
Descrição	Coluna responsável por reter as informações assim que ocorre o acesso					
Atributos						
Nome da Coluna	Tipo do Dado	Valor min e max	Nulidade	PK	FK	Descrição
id_log	serial	1 – sem limite	NOT NULL	X		Código de identificador da avaliação
id_usuario	serial	1 – sem limite	NOT NULL		X	Nome da avaliação
id_documento	serial	1 – sem limite	NOT NULL		X	
dt_acesso	date		NOT NULL			
tp_acesso	varchar(50)	1 - 50	NOT NULL			
Índice						
Nome do índice	Clustered	NonClustered	Unique	Colunas		
Idx_dt_avaliacao	X		X	cd_avaliacao, dt_avaliacao		
Idx_dt_avaliacao		X		cd_avaliacao, dt_avaliacao		

Idx_dt_avaliacao		X	X	cd_avaliacao, dt_avaliacao
Idx_dt_avaliacao		X		cd_avaliacao, dt_avaliacao

Tabela	tipo_documento					
Descrição	Tabela responsável por registrar as informações referente ao tipo do documento					
Atributos						
Nome da Coluna	Tipo do Dado	Valor min e max	Nulidade	PK	FK	Descrição
id_tipo_documento	serial	1 – sem limite	NOT NULL	X		Código identificador do tipo do documento
nome	varchar(50)	1 - 50	NOT NULL			Nome do tipo do documento
descricao	text	1 – sem limite	NOT NULL			Descricao do tipo do documento
Índice						
Nome do índice	Clustered	NonClustered	Unique	Colunas		
Idx_dt_avaliacao	X		X	cd_avaliacao, dt_avaliacao		
Idx_dt_avaliacao		X		cd_avaliacao, dt_avaliacao		
Idx_dt_avaliacao		X	X	cd_avaliacao, dt_avaliacao		
Idx_dt_avaliacao		X		cd_avaliacao, dt_avaliacao		

Tabela	pasta					
Descrição	Tabela responsável pelo registro da pasta criada					
Atributos						
Nome da Coluna	Tipo do Dado	Valor min e max	Nulidade	PK	FK	Descrição
id_pasta	serial	1 – sem limite	NOT NULL	X		Código identificador da pasta
nome	varchar(200)	10 - 200	NOT NULL			Nome da pasta

descricao	text	1 – sem limite	NOT NULL			Descrição da pasta
id_pasta_pai	varchar(50)	1 - 50	NOT NULL			Código identificador da pasta pai
id_proprietario	serial	1 – sem limite	NOT NULL		X	Código identificador do proprietário
dt_criacao	int	1 – sem limite	NOT NULL			Data da criação
<b>Índice</b>						
<b>Nome do índice</b>	<b>Clustered</b>	<b>NonClustered</b>	<b>Unique</b>	<b>Colunas</b>		
Idx_dt_avaliacao	X		X	cd_avaliacao, dt_avaliacao		
Idx_dt_avaliacao		X		cd_avaliacao, dt_avaliacao		
Idx_dt_avaliacao		X	X	cd_avaliacao, dt_avaliacao		
Idx_dt_avaliacao		X		cd_avaliacao, dt_avaliacao		

Tabela	permissao					
Descrição	Tabela responsável pela permissão de acesso ao documento					
Atributos						
Nome da Coluna	Tipo do Dado	Valor min e max	Nulidade	PK	FK	Descrição
id_permissao	serial	1 – sem limite	NOT NULL	X		Código identificador da permissão
id_usuario	serial	1 – sem limite	NOT NULL		X	Código identificador do usuário
id_documento	serial	1 – sem limite	NOT NULL		X	Código identificador do documento
id_pasta	serial	1 – sem limite	NOT NULL		X	Código identificador da pasta
tp_permissao	varchar(50)	10 - 50	NOT NULL			Tipo de permissão
Índice						
Nome do índice	Clustered	NonClustered	Unique	Colunas		
Idx_dt_avaliacao	X		X	cd_avaliacao, dt_avaliacao		

Idx_dt_avaliacao		X		cd_avaliacao, dt_avaliacao
Idx_dt_avaliacao		X	X	cd_avaliacao, dt_avaliacao
Idx_dt_avaliacao		X		cd_avaliacao, dt_avaliacao

Tabela	Versão					
Descrição	Tabela responsável pelo registro da versão do documento					
Atributos						
Nome da Coluna	Tipo do Dado	Valor min e max	Nulidade	PK	FK	Descrição
id_versão	serial	1 – sem limite	NOT NULL	X		Código identificador da versão
id_documento	serial	1 – sem limite	NOT NULL		X	Código identificador do documento
numero_versao	int	1 – sem limite	NOT NULL			Número da versão
caminho_arquivo	varchar(255)	1 -255	NOT NULL			Caminho até o arquivo
dt_criacao	date		NOT NULL			Data de criação
descricao_alteracao	text	1 – sem limite	NOT NULL			Descrição da ultima alteração
Índice						
Nome do índice	Clustered	NonClustered	Unique	Colunas		
Idx_dt_avaliacao	X		X	cd_avaliacao, dt_avaliacao		
Idx_dt_avaliacao		X		cd_avaliacao, dt_avaliacao		
Idx_dt_avaliacao		X	X	cd_avaliacao, dt_avaliacao		
Idx_dt_avaliacao		X		cd_avaliacao, dt_avaliacao		

<b>Tabela</b>	Usuário
<b>Descrição</b>	Tabela responsável pelo registro do usuário.



Atributos						
Nome da Coluna	Tipo do Dado	Valor min e max	Nulidade	PK	FK	Descrição
id_usuario	serial	1 – sem limite	NOT NULL	X		Código de identificador do usuário
nome	varchar(100)	10 - 100	NOT NULL			Nome do usuário
email	varchar(100)	10 - 100	NOT NULL			E-mail do usuário
senha	varchar(100)	10 - 100	NOT NULL			Senha do usuário
dt_cadastro	date		NOT NULL			Data de cadastro
tp_usuario	varchar(50)	1 - 50	NOT NULL			Tipo de usuário
Índice						
Nome do índice	Clustered	NonClustered	Unique	Colunas		
Idx_dt_avaliacao	X		X	cd_avaliacao, dt_avaliacao		
Idx_dt_avaliacao		X		cd_avaliacao, dt_avaliacao		
Idx_dt_avaliacao		X	X	cd_avaliacao, dt_avaliacao		
Idx_dt_avaliacao		X		cd_avaliacao, dt_avaliacao		

Tabela	tag					
Descrição	Tabela responsável por registrar as tags					
Atributos						
Nome da Coluna	Tipo do Dado	Valor min e max	Nulidade	PK	FK	Descrição
id_tag	serial	1 – sem limite	NOT NULL	X		Código identificador da tag
nome	varchar(50)	1 - 50	NOT NULL		X	Nome da tag
Índice						
Nome do índice	Clustered	NonClustered	Unique	Colunas		
ldx_dt_avaliacao	X		X	cd_avaliacao, dt_avaliacao		

Idx_dt_avaliacao		X		cd_avaliacao, dt_avaliacao
Idx_dt_avaliacao		X	X	cd_avaliacao, dt_avaliacao
Idx_dt_avaliacao		X		cd_avaliacao, dt_avaliacao

## Script dos comandos DDL para criação do Banco de dados

```
CREATE INDEX idx_document_name ON "documento" (nome);
```

```
CREATE INDEX idx_document_type_id ON "documento" (id_tipo_documento);
```

```
-- CreateEnum  
CREATE TYPE "tp_usuario" AS ENUM ('ADMIN', 'USER');
```

```
-- CreateEnum  
CREATE TYPE "tp_acesso" AS ENUM ('VIEW', 'EDIT');
```

```
-- CreateEnum  
CREATE TYPE "tp_permissao" AS ENUM ('READ', 'WRITE');  
-- public.tag definition
```

```
-- Drop table
```

```
-- DROP TABLE public.tag;
```

```
CREATE TABLE public.tag (  
  id_tag serial4 NOT NULL,  
  nome varchar(50) NOT NULL,  
  CONSTRAINT tag_pkey PRIMARY KEY (id_tag)  
);
```

```
-- public.tipo_documento definition
```

```
-- Drop table
```

```
-- DROP TABLE public.tipo_documento;
```

```
CREATE TABLE public.tipo_documento (  
  id_tipo_documento serial4 NOT NULL,  
  nome varchar(50) NOT NULL,  
  descricao text NOT NULL,  
  CONSTRAINT tipo_documento_pkey PRIMARY KEY (id_tipo_documento)  
);
```

```
-- public.usuario definition
```

```
-- Drop table
```

```
-- DROP TABLE public.usuario;
```

```
CREATE TABLE public.usuario (  
  id_usuario serial4 NOT NULL,  
  nome varchar(100) NOT NULL,  
  email varchar(250) NOT NULL,  
  senha varchar(100) NOT NULL,  
  dt_cadastro timestamp(3) DEFAULT CURRENT_TIMESTAMP NOT NULL,  
  tp_usuario public.tp_usuario DEFAULT 'USER'::tp_usuario NOT NULL,  
  CONSTRAINT usuario_pkey PRIMARY KEY (id_usuario)  
);  
CREATE UNIQUE INDEX usuario_email_key ON public.usuario USING btree (email);
```

```
-- public.documento definition
```

```
-- Drop table
```

```
-- DROP TABLE public.documento;
```

```
CREATE TABLE public.documento (  
  id_documento serial4 NOT NULL,  
  nome varchar(100) NOT NULL,  
  descricao text NOT NULL,  
  caminho_arquivo varchar(255) NOT NULL,  
  dt_criacao timestamp(3) DEFAULT CURRENT_TIMESTAMP NOT NULL,  
  dt_modificacao timestamp(3) NOT NULL,  
  id_proprietario int4 NOT NULL,  
  id_tipo_documento int4 NOT NULL,  
  CONSTRAINT documento_pkey PRIMARY KEY (id_documento),  
  CONSTRAINT documento_id_proprietario_fkey FOREIGN KEY (id_proprietario) REFERENCES public.usuario(id_usuario) ON DELETE RESTRICT ON UPDATE CASCADE,  
  CONSTRAINT documento_id_tipo_documento_fkey FOREIGN KEY (id_tipo_documento) REFERENCES public.tipo_documento(id_tipo_documento) ON DELETE RESTRICT ON UPDATE CASCADE  
);
```

```
-- public.documento_tag definition
```

```
-- Drop table
```

```
-- DROP TABLE public.documento_tag;
```

```
CREATE TABLE public.documento_tag (  
  id_documento int4 NOT NULL,  
  id_tag int4 NOT NULL,  
  CONSTRAINT documento_tag_pkey PRIMARY KEY (id_documento, id_tag),  
  CONSTRAINT documento_tag_id_documento_fkey FOREIGN KEY (id_documento) REFERENCES public.documento(id_documento) ON DELETE RESTRICT ON  
  UPDATE CASCADE,  
  CONSTRAINT documento_tag_id_tag_fkey FOREIGN KEY (id_tag) REFERENCES public.tag(id_tag) ON DELETE RESTRICT ON UPDATE CASCADE  
);
```

```
-- public.log_acesso definition
```

```
-- Drop table
```

```
-- DROP TABLE public.log_acesso;
```

```
CREATE TABLE public.log_acesso (  
  id_log serial4 NOT NULL,  
  id_usuario int4 NOT NULL,  
  id_documento int4 NOT NULL,  
  dt_acesso timestamp(3) NOT NULL,  
  tp_acesso public.tp_acesso NOT NULL,  
  CONSTRAINT log_acesso_pkey PRIMARY KEY (id_log),  
  CONSTRAINT log_acesso_id_documento_fkey FOREIGN KEY (id_documento) REFERENCES public.documento(id_documento) ON DELETE RESTRICT ON  
  UPDATE CASCADE,  
  CONSTRAINT log_acesso_id_usuario_fkey FOREIGN KEY (id_usuario) REFERENCES public.usuario(id_usuario) ON DELETE RESTRICT ON UPDATE  
  CASCADE  
);
```

```
-- public.pasta definition
```

```
-- Drop table
```

```
-- DROP TABLE public.pasta;
```

```

CREATE TABLE public.pasta (
  id_pasta serial4 NOT NULL,
  nome varchar(100) NOT NULL,
  descricao text NOT NULL,
  id_pasta_pai int4,
  id_proprietario int4 NOT NULL,
  dt_criacao timestamp(3) DEFAULT CURRENT_TIMESTAMP NOT NULL,
  CONSTRAINT pasta_pkey PRIMARY KEY (id_pasta),
  CONSTRAINT pasta_id_pasta_pai_fkey FOREIGN KEY (id_pasta_pai) REFERENCES public.pasta(id_pasta) ON DELETE RESTRICT ON UPDATE CASCADE,
  CONSTRAINT pasta_id_proprietario_fkey FOREIGN KEY (id_proprietario) REFERENCES public.usuario(id_usuario) ON DELETE RESTRICT ON UPDATE CASCADE
);

-- public.permissao definition

-- Drop table

-- DROP TABLE public.permissao;

CREATE TABLE public.permissao (
  id_permissao serial4 NOT NULL,
  id_usuario int4 NOT NULL,
  id_documento int4 NULL,
  id_pasta int4 NULL,
  tp_permissao public.tp_permissao NOT NULL,
  CONSTRAINT permissao_pkey PRIMARY KEY (id_permissao),
  CONSTRAINT permissao_id_documento_fkey FOREIGN KEY (id_documento) REFERENCES public.documento(id_documento) ON DELETE SET NULL ON UPDATE CASCADE,
  CONSTRAINT permissao_id_pasta_fkey FOREIGN KEY (id_pasta) REFERENCES public.pasta(id_pasta) ON DELETE SET NULL ON UPDATE CASCADE,
  CONSTRAINT permissao_id_usuario_fkey FOREIGN KEY (id_usuario) REFERENCES public.usuario(id_usuario) ON DELETE RESTRICT ON UPDATE CASCADE
);

-- public.versao definition

-- Drop table

-- DROP TABLE public.versao;

CREATE TABLE public.versao (

```

```

id_versao serial4 NOT NULL,
id_documento int4 NOT NULL,
numero_versao int4 NOT NULL,
"path" varchar(255) NOT NULL,
dt_criacao timestamp(3) DEFAULT CURRENT_TIMESTAMP NOT NULL,tri
descricao_alteracao text NOT NULL,
CONSTRAINT versao_pkey PRIMARY KEY (id_versao),
CONSTRAINT versao_id_documento_fkey FOREIGN KEY (id_documento) REFERENCES public.documento(id_documento) ON DELETE RESTRICT ON UPDATE
CASCADE
);

-- public.documento_pasta definition

-- Drop table

-- DROP TABLE public.documento_pasta;

CREATE TABLE public.documento_pasta (
id_documento int4 NOT NULL,
id_pasta int4 NOT NULL,
CONSTRAINT documento_pasta_pkey PRIMARY KEY (id_documento, id_pasta),
CONSTRAINT documento_pasta_id_documento_fkey FOREIGN KEY (id_documento) REFERENCES public.documento(id_documento) ON DELETE RESTRICT ON
UPDATE CASCADE,
CONSTRAINT documento_pasta_id_pasta_fkey FOREIGN KEY (id_pasta) REFERENCES public.pasta(id_pasta) ON DELETE RESTRICT ON UPDATE CASCADE
);

```

## Script que popula as tabelas do Banco de dados

-- Inserir usuários

```
INSERT INTO usuario (id_usuario, nome, email, senha, dt_cadastro, tp_usuario) VALUES
(1, 'Alice Silva', 'alice@example.com', 'senha123', '2023-01-01', 'ADMIN'),
(2, 'Bruno Souza', 'bruno@example.com', 'senha123', '2023-01-02', 'USER'),
(3, 'Carla Pereira', 'carla@example.com', 'senha123', '2023-01-03', 'USER'),
(4, 'Diego Costa', 'diego@example.com', 'senha123', '2023-01-04', 'USER'),
(5, 'Elena Martins', 'elena@example.com', 'senha123', '2023-01-05', 'USER'),
(6, 'Felipe Lima', 'felipe@example.com', 'senha123', '2023-01-06', 'USER'),
(7, 'Gabriela Fernandes', 'gabriela@example.com', 'senha123', '2023-01-07', 'USER'),
(8, 'Henrique Almeida', 'henrique@example.com', 'senha123', '2023-01-08', 'USER'),
(9, 'Isabela Rodrigues', 'isabela@example.com', 'senha123', '2023-01-09', 'USER'),
(10, 'João Silva', 'joao@example.com', 'senha123', '2023-01-10', 'USER');
```

-- Inserir tipos de documentos

```
INSERT INTO tipo_documento (id_tipo_documento, nome, descricao) VALUES
(1, 'Relatório', 'Documentos de relatórios diversos'),
(2, 'Proposta', 'Propostas comerciais e técnicas'),
(3, 'Contrato', 'Contratos e acordos legais'),
(4, 'Manual', 'Manuais de operação e uso'),
(5, 'Política', 'Políticas internas e procedimentos');
```



(6, 'Memorando', 'Memorandos e comunicados internos'),  
(7, 'Ata', 'Atas de reuniões'),  
(8, 'Plano', 'Planos de projeto e estratégicos'),  
(9, 'Formulário', 'Formulários para preenchimento'),  
(10, 'Outros', 'Outros tipos de documentos');

-- Inserir documentos

```
INSERT INTO documento (id_documento, nome, descricao, caminho_arquivo, dt_criacao, dt_modificacao, id_proprietario, id_tipo_documento) VALUES
```

(1, 'Relatório Financeiro Q1', 'Relatório financeiro do primeiro trimestre', '/caminho/relatorio1.pdf', '2023-01-15', '2023-01-15', 1, 1),  
(2, 'Proposta Comercial XYZ', 'Proposta comercial para o projeto XYZ', '/caminho/proposta1.pdf', '2023-02-01', '2023-02-01', 2, 2),  
(3, 'Contrato de Fornecimento', 'Contrato de fornecimento com a empresa ABC', '/caminho/contrato1.pdf', '2023-02-15', '2023-02-15', 3, 3),  
(4, 'Manual do Usuário', 'Manual do usuário para o software', '/caminho/manual1.pdf', '2023-03-01', '2023-03-01', 4, 4),  
(5, 'Política de Segurança', 'Política de segurança da informação', '/caminho/politica1.pdf', '2023-03-15', '2023-03-15', 5, 5),  
(6, 'Memorando Interno', 'Memorando sobre novas diretrizes', '/caminho/memorando1.pdf', '2023-04-01', '2023-04-01', 6, 6),  
(7, 'Ata de Reunião de Diretoria', 'Ata da reunião da diretoria de março', '/caminho/ata1.pdf', '2023-04-15', '2023-04-15', 7, 7),  
(8, 'Plano de Projeto ABC', 'Plano do projeto ABC para 2023', '/caminho/plano1.pdf', '2023-05-01', '2023-05-01', 8, 8),  
(9, 'Formulário de Cadastro', 'Formulário de cadastro de novos clientes', '/caminho/formulario1.pdf', '2023-05-15', '2023-05-15', 9, 9),  
(10, 'Outros Documentos', 'Documentos diversos não categorizados', '/caminho/outros1.pdf', '2023-06-01', '2023-06-01', 10, 10);

-- Inserir pastas

```
INSERT INTO pasta (id_pasta, nome, descricao, id_pasta_pai, id_proprietario, dt_criacao) VALUES
```

(1, 'Financeiro', 'Documentos financeiros', NULL, 1, '2023-01-01'),

```
(2, 'Comercial', 'Documentos comerciais', NULL, 2, '2023-01-02'),  
(3, 'Jurídico', 'Documentos jurídicos', NULL, 3, '2023-01-03'),  
(4, 'TI', 'Documentos de tecnologia da informação', NULL, 4, '2023-01-04'),  
(5, 'RH', 'Documentos de recursos humanos', NULL, 5, '2023-01-05'),  
(6, 'Administrativo', 'Documentos administrativos', NULL, 6, '2023-01-06'),  
(7, 'Operacional', 'Documentos operacionais', NULL, 7, '2023-01-07'),  
(8, 'Projetos', 'Documentos de projetos', NULL, 8, '2023-01-08'),  
(9, 'Marketing', 'Documentos de marketing', NULL, 9, '2023-01-09'),  
(10, 'Outros', 'Documentos diversos', NULL, 10, '2023-01-10');
```

-- Inserir permissões

```
INSERT INTO permissao (id_permissao, id_usuario, id_documento, id_pasta, tp_permissao) VALUES
```

```
(1, 1, 1, NULL, 'READ'),  
(2, 2, 2, NULL, 'WRITE'),  
(3, 3, 3, NULL, 'READ'),  
(4, 4, 4, NULL, 'READ'),  
(5, 5, 5, NULL, 'WRITE'),  
(6, 6, 6, NULL, 'READ'),  
(7, 7, 7, NULL, 'READ'),  
(8, 8, 8, NULL, 'WRITE'),  
(9, 9, 9, NULL, 'READ'),  
(10, 10, 10, NULL, 'READ');
```

-- Inserir logs de acesso

INSERT INTO log\_acesso (id\_log, id\_usuario, id\_documento, dt\_acesso, tp\_acesso) VALUES

(1, 1, 1, '2023-01-16', 'VIEW'),

(2, 2, 2, '2023-02-02', 'EDIT'),

(3, 3, 3, '2023-02-16', 'VIEW'),

(4, 4, 4, '2023-03-02', 'VIEW'),

(5, 5, 5, '2023-03-16', 'EDIT'),

(6, 6, 6, '2023-04-02', 'VIEW'),

(7, 7, 7, '2023-04-16', 'VIEW'),

(8, 8, 8, '2023-05-02', 'EDIT'),

(9, 9, 9, '2023-05-16', 'VIEW'),

(10, 10, 10, '2023-06-02', 'VIEW');

-- Inserir versões

INSERT INTO versao (id-versao, id\_documento, numero-versao, path, dt\_criacao, descricao\_alteracao) VALUES

(1, 1, 1, '/caminho/versao1\_documento1.pdf', '2023-01-15', 'Versão inicial'),

(2, 2, 1, '/caminho/versao1\_documento2.pdf', '2023-02-01', 'Versão inicial'),

(3, 3, 1, '/caminho/versao1\_documento3.pdf', '2023-02-15', 'Versão inicial'),

(4, 4, 1, '/caminho/versao1\_documento4.pdf', '2023-03-01', 'Versão inicial'),

(5, 5, 1, '/caminho/versao1\_documento5.pdf', '2023-03-15', 'Versão inicial'),

(6, 6, 1, '/caminho/versao1\_documento6.pdf', '2023-04-01', 'Versão inicial'),

```
(7, 7, 1, '/caminho/versao1_documento7.pdf', '2023-04-15', 'Versão inicial'),  
(8, 8, 1, '/caminho/versao1_documento8.pdf', '2023-05-01', 'Versão inicial'),  
(9, 9, 1, '/caminho/versao1_documento9.pdf', '2023-05-15', 'Versão inicial'),  
(10, 10, 1, '/caminho/versao1_documento10.pdf', '2023-06-01', 'Versão inicial');
```

-- Inserir tags

```
INSERT INTO tag (id_tag, nome) VALUES
```

```
(1, 'Financeiro'),  
(2, 'Comercial'),  
(3, 'Jurídico'),  
(4, 'TI'),  
(5, 'RH'),  
(6, 'Administrativo'),  
(7, 'Operacional'),  
(8, 'Projetos'),  
(9, 'Marketing'),  
(10, 'Outros');
```

-- Inserir relacionamento documento\_tag

```
INSERT INTO documento_tag (id_documento, id_tag) VALUES
```

```
(1, 1),  
(2, 2),
```

```
(3, 3),  
(4, 4),  
(5, 5),  
(6, 6),  
(7, 7),  
(8, 8),  
(9, 9),  
(10, 10);
```

```
-- Inserir relacionamento documento_pasta
```

```
INSERT INTO documento_pasta (id_documento, id_pasta) VALUES
```

```
(1, 1),  
(2, 2),  
(3, 3),  
(4, 4),  
(5, 5),  
(6, 6),  
(7, 7),  
(8, 8),  
(9, 9),  
(10, 10);
```



## Triggers

-- Criando a função trigger

```
CREATE OR REPLACE FUNCTION atualiza_data_modificacao()
```

```
RETURNS TRIGGER AS $$
```

```
BEGIN
```

```
    -- Atualiza a data de modificação para o momento atual
```

```
    NEW.dt_modificacao := CURRENT_TIMESTAMP;
```

```
    RETURN NEW;
```

```
END;
```

```
$$ LANGUAGE plpgsql;
```

-- Criando a trigger na tabela documento

```
CREATE TRIGGER documento_atualiza_modificacao
```

```
BEFORE UPDATE ON public.documento
```

```
FOR EACH ROW
```

```
EXECUTE FUNCTION atualiza_data_modificacao();
```

-- Inserindo um documento de teste

```
INSERT INTO public.documento (id_documento, nome, descricao, caminho_arquivo, dt_criacao, dt_modificacao, id_proprietario, id_tipo_documento)
```

```
VALUES (15, 'Documento Teste', 'Descrição do Documento Teste', '/caminho/arquivo/teste.doc', CURRENT_TIMESTAMP, CURRENT_TIMESTAMP, 1, 1);
```

```
UPDATE public.documento  
SET descricao = 'Descrição modificada do Documento Teste'  
WHERE id_documento = 1;
```

```
-- Seleccionando o documento atualizado para verificar a data de modificação  
SELECT * FROM public.documento WHERE id_documento = 1;
```

## **Principais consultas mapeadas baseadas em regras de negócio (mínimo 4)**

```
-- Quantos documentos cada usuário possui na coleção principal (pasta sem pai)?  
CREATE OR REPLACE FUNCTION count_documents_in_main_collection()  
RETURNS TABLE (usuario varchar(100), quantidade_documentos bigint) AS $$  
BEGIN  
    RETURN QUERY  
    SELECT u.nome AS usuario, COUNT(d.id_documento) AS quantidade_documentos  
    FROM usuario u  
    LEFT JOIN documento d ON u.id_usuario = d.id_proprietario  
    LEFT JOIN pasta p ON u.id_usuario = p.id_proprietario AND p.id_pasta_pai IS NULL  
    GROUP BY u.nome;
```



```
END;
```

```
$$ LANGUAGE PLPGSQL;
```

```
-- Qual é o documento mais acessado até o momento?
```

```
CREATE OR REPLACE FUNCTION find_most_accessed_document()
```

```
RETURNS TABLE (documento_mais_acessado varchar(100), quantidade_acessos bigint) AS $$
```

```
BEGIN
```

```
    RETURN QUERY
```

```
    SELECT d.nome AS documento_mais_acessado, COUNT(l.id_log) AS quantidade_acessos
```

```
    FROM log_acesso l
```

```
    JOIN documento d ON l.id_documento = d.id_documento
```

```
    GROUP BY d.nome
```

```
    ORDER BY quantidade_acessos DESC
```

```
    LIMIT 1;
```

```
END;
```

```
$$ LANGUAGE PLPGSQL;
```

```
-- Quais são os usuários que têm permissão de escrita (WRITE) em mais de um documento?
```

```
CREATE OR REPLACE FUNCTION find_users_with_write_permissions()
```

```
RETURNS TABLE (usuario varchar(100), quantidade_permissoes_write bigint) AS $$
```

```
BEGIN
```

```
    RETURN QUERY
```

```
SELECT u.nome AS usuario, COUNT(p.id_permissao) AS quantidade_permissoes_write
FROM usuario u
JOIN permissao p ON u.id_usuario = p.id_usuario AND p.tp_permissao = 'WRITE'
GROUP BY u.nome
HAVING COUNT(p.id_permissao) > 1;
END;
$$ LANGUAGE PLPGSQL;
```

-- Quantos documentos estão associados a cada tipo de documento?

```
CREATE OR REPLACE FUNCTION count_documents_by_document_type()
RETURNS TABLE (tipo_documento varchar(50), quantidade_documentos bigint) AS $$
BEGIN
    RETURN QUERY
    SELECT td.nome AS tipo_documento, COUNT(d.id_documento) AS quantidade_documentos
    FROM tipo_documento td
    LEFT JOIN documento d ON td.id_tipo_documento = d.id_tipo_documento
    GROUP BY td.nome;
END;
$$ LANGUAGE PLPGSQL;
```

-- Exemplo de chamada da função para contar documentos na coleção principal

```
SELECT * FROM count_documents_in_main_collection();
```

-- Exemplo de chamada da função para encontrar o documento mais acessado

```
SELECT * FROM find_most_accessed_document();
```

-- Exemplo de chamada da função para encontrar usuários com permissões de escrita em mais de um documento

```
SELECT * FROM find_users_with_write_permissions();
```

-- Exemplo de chamada da função para contar documentos por tipo de documento

```
SELECT * FROM count_documents_by_document_type();
```