

Especificação do Projeto - Parte 1

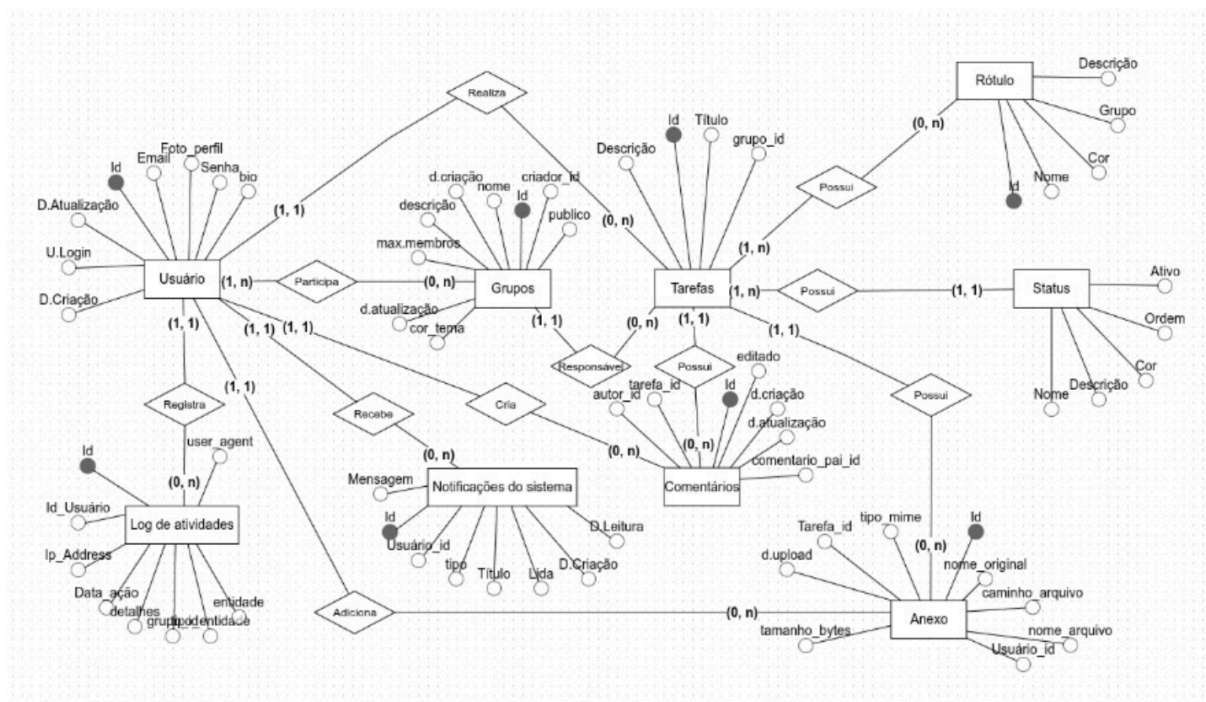
João Vitor Santos de Sena - 190089890
Gustavo Vieira do Nascimento - 222012872
Richard de Carvalho Dorneles Otto - 231006186

Faremos um aplicativo de gestão colaborativa de trabalhos em grupo — algo na linha de um Trello, com recursos para cadastro de usuários, criação de projetos com múltiplos participantes, atribuição de tarefas, anexos reutilizáveis entre projetos e tarefas, e um sistema de conclusão de tarefas.

O usuário poderá informar seu nome, email e senha, e a partir disso poderá criar, ler, atualizar grupos, e dentro desses grupos poderá adicionar membros e tarefas.

Também poderá atribuir outros usuários a tarefas e deixar anotações para eles.

Modelo Entidade Relacionamento:



```
classDiagram
    class usuarios {
        id : UUID
        nome : VARCHAR
        email : VARCHAR
        senha_hash : VARCHAR
        foto_perfil : TEXT
        bio : TEXT
        ativo : BOOLEAN
        data_criacao : TIMESTAMP
        data_atualizacao : TIMESTAMP
        ultimo_login : TIMESTAMP
    }
    class grupos {
        id : UUID
        nome : VARCHAR
        descricao : TEXT
        cor_tema : VARCHAR
        criador_id : UUID
        publico : BOOLEAN
        max_membros : INTEGER
        data_criacao : TIMESTAMP
        data_atualizacao : TIMESTAMP
    }
    class atividades {
        id : UUID
        tipo_entidade : VARCHAR
        entidade_id : UUID
        acao : VARCHAR
        usuario_id : UUID
        grupo_id : UUID
        detalhes : JSONB
        ip_address : TEXT
        user_agent : TEXT
        data_acao : TIMESTAMP
    }
    class rotulos {
        id : UUID
        nome : VARCHAR
        descricao : TEXT
        cor : VARCHAR
        grupo_id : UUID
    }
    class status_tarefa {
        id : SERIAL
        nome : VARCHAR
        descricao : TEXT
        cor : VARCHAR
        ordem : INTEGER
        ativo : BOOLEAN
    }
    class tarefas {
        id : UUID
        titulo : VARCHAR
        descricao : TEXT
        grupo_id : UUID
        criador_id : UUID
        status_id : INTEGER
        prioridade : INTEGER
        data_inicio : DATE
        data_termino : DATE
        estimativa_horas : NUMERIC
        horas_trabalhadas : NUMERIC
        progresso : INTEGER
        data_criacao : TIMESTAMP
        data_atualizacao : TIMESTAMP
        data_conclusao : TIMESTAMP
    }
    class notificacoes {
        id : UUID
        usuario_id : UUID
        tipo : VARCHAR
        titulo : VARCHAR
        mensagem : TEXT
        entidade_tipo : VARCHAR
        entidade_id : UUID
        lida : BOOLEAN
        data_criacao : TIMESTAMP
        data_leitura : TIMESTAMP
    }
    class comentarios {
        id : UUID
        tarefa_id : UUID
        autor_id : UUID
        conteudo : TEXT
        comentario_pai_id : UUID
        data_criacao : TIMESTAMP
        data_atualizacao : TIMESTAMP
        editado : BOOLEAN
    }
    class anexos {
        id : UUID
        tarefa_id : UUID
        usuario_id : UUID
        nome_original : VARCHAR
        nome_arquivo : VARCHAR
        tipo_mime : VARCHAR
        tamanho_bytes : BIGINT
        caminho_arquivo : TEXT
        data_upload : TIMESTAMP
    }
    class atribucoes_tarefa {
        id : UUID
        tarefa_id : UUID
        usuario_id : UUID
        atribuido_por : UUID
        data_atribuicao : TIMESTAMP
        ativo : BOOLEAN
    }
    class usuarios_grupos {
        id : UUID
        usuario_id : UUID
        grupo_id : UUID
        papel : VARCHAR
        data_entrada : TIMESTAMP
        ativo : BOOLEAN
    }
    class tarefas_rotulos {
        id : UUID
        tarefa_id : UUID
        rotulo_id : UUID
    }

    usuarios "1" -- "1" grupos : cria
    usuarios "1" -- "1" grupos : participa
    usuarios "1" -- "0..*" notificacoes : recebe notificacao
    usuarios "1" -- "0..*" tarefas : recebe notificacao
    usuarios "1" -- "0..*" comentarios : recebe
    usuarios "1" -- "0..*" anexos : recebe
    usuarios "1" -- "0..*" atribucoes_tarefa : recebe
    grupos "0..*" -- "0..*" atividades : realiza acao
    grupos "0..*" -- "0..*" tarefas : realiza acao
    grupos "0..*" -- "0..*" comentarios : realiza acao
    grupos "0..*" -- "0..*" anexos : realiza acao
    grupos "0..*" -- "0..*" atribucoes_tarefa : realiza acao
    atividades "0..*" -- "0..*" grupos : relaciona
    atividades "0..*" -- "0..*" tarefas : relaciona
    atividades "0..*" -- "0..*" comentarios : relaciona
    atividades "0..*" -- "0..*" anexos : relaciona
    atividades "0..*" -- "0..*" atribucoes_tarefa : relaciona
    rotulos "1" -- "0..*" grupos : possui
    rotulos "1" -- "0..*" tarefas : possui
    status_tarefa "1" -- "0..*" tarefas : status
    tarefas "0..*" -- "0..*" grupos : cria
    tarefas "0..*" -- "0..*" status_tarefa : cria
    tarefas "0..*" -- "0..*" comentarios : cria
    tarefas "0..*" -- "0..*" anexos : cria
    tarefas "0..*" -- "0..*" atribucoes_tarefa : cria
    notificacoes "0..*" -- "0..*" usuarios : envia anexo
    comentarios "0..*" -- "0..*" usuarios : comenta
    comentarios "0..*" -- "0..*" grupos : comenta
    comentarios "0..*" -- "0..*" tarefas : comenta
    comentarios "0..*" -- "0..*" anexos : comenta
    anexos "0..*" -- "0..*" tarefas : tem anexo
    atribucoes_tarefa "0..*" -- "0..*" tarefas : atribuida
```

```
-- Extensão para gerar UUIDs
CREATE EXTENSION IF NOT EXISTS "uuid-ossf";

-- =====
-- 1. TABELA DE USUÁRIOS
-- =====

CREATE TABLE usuarios (
  id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
  nome VARCHAR(100) NOT NULL,
  email VARCHAR(255) UNIQUE NOT NULL,
  senha_hash VARCHAR(255) NOT NULL, -- bcrypt hash
  foto_perfil TEXT, -- URL ou caminho da foto
  bio TEXT,
  ativo BOOLEAN DEFAULT TRUE,
  data_criacao TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,
  data_atualizacao TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,
  ultimo_login TIMESTAMP WITH TIME ZONE,

  -- Comentários explicativos
```

```

        CONSTRAINT usuarios_email_check CHECK (email ~*
'^[A-Za-z0-9._%~+@][A-Za-z0-9.-]+\.[A-Za-z]{2,4}$'),
        CONSTRAINT usuarios_nome_check CHECK (LENGTH(nome) >= 2)
);

```

```

COMMENT ON TABLE usuarios IS 'Tabela principal para armazenar informações dos
usuários do sistema';
COMMENT ON COLUMN usuarios.id IS 'Identificador único do usuário (UUID)';
COMMENT ON COLUMN usuarios.email IS 'Email único do usuário para autenticação';
COMMENT ON COLUMN usuarios.senha_hash IS 'Hash da senha do usuário usando
bcrypt';
COMMENT ON COLUMN usuarios.ativo IS 'Indica se o usuário está ativo no sistema';

```

```
-- =====
```

```
-- 2. TABELA DE GRUPOS
```

```
-- =====
```

```

CREATE TABLE grupos (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
    nome VARCHAR(100) NOT NULL,
    descricao TEXT,
    cor_tema VARCHAR(7) DEFAULT '#007bff', -- Hex color
    criador_id UUID NOT NULL REFERENCES usuarios(id) ON DELETE RESTRICT,
    publico BOOLEAN DEFAULT FALSE,
    max_membros INTEGER DEFAULT 50,
    data_criacao TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,
    data_atualizacao TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,

```

```

    CONSTRAINT grupos_max_membros_check CHECK (max_membros > 0 AND
max_membros <= 100),
    CONSTRAINT grupos_cor_check CHECK (cor_tema ~* '^#[0-9a-f]{6}$')
);

```

```

COMMENT ON TABLE grupos IS 'Grupos de trabalho onde as tarefas são organizadas';
COMMENT ON COLUMN grupos.criador_id IS 'Usuário que criou o grupo';
COMMENT ON COLUMN grupos.publico IS 'Indica se o grupo é público ou privado';
COMMENT ON COLUMN grupos.max_membros IS 'Número máximo de membros
permitidos no grupo';

```

```
-- =====
```

```
-- 3. TABELA DE RELACIONAMENTO USUÁRIOS-GRUPOS (PIVOT)
```

```
-- =====
```

```

CREATE TABLE usuarios_grupos (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
    usuario_id UUID NOT NULL REFERENCES usuarios(id) ON DELETE CASCADE,
    grupo_id UUID NOT NULL REFERENCES grupos(id) ON DELETE CASCADE,
    papel VARCHAR(20) NOT NULL DEFAULT 'membro',
    data_entrada TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,
    ativo BOOLEAN DEFAULT TRUE,

```

```
    UNIQUE(usuario_id, grupo_id),
    CONSTRAINT usuarios_grupos_papel_check CHECK (papel IN ('admin', 'moderador',
'membro'))
);
```

```
COMMENT ON TABLE usuarios_grupos IS 'Relacionamento muitos-para-muitos entre
usuários e grupos';
COMMENT ON COLUMN usuarios_grupos.papel IS 'Papel do usuário no grupo: admin,
moderador ou membro';
```

```
-- =====
-- 4. TABELA DE STATUS DE TAREFAS
-- =====
```

```
CREATE TABLE status_tarefa (
    id SERIAL PRIMARY KEY,
    nome VARCHAR(50) UNIQUE NOT NULL,
    descricao TEXT,
    cor VARCHAR(7) NOT NULL DEFAULT '#6c757d',
    ordem INTEGER NOT NULL DEFAULT 0,
    ativo BOOLEAN DEFAULT TRUE,

    CONSTRAINT status_cor_check CHECK (cor ~* '^[0-9a-f]{6}$')
);
```

```
COMMENT ON TABLE status_tarefa IS 'Status possíveis para as tarefas (ex: pendente, em
andamento, concluída)';
COMMENT ON COLUMN status_tarefa.ordem IS 'Ordem de exibição dos status';
```

```
-- Inserir status padrão
INSERT INTO status_tarefa (nome, descricao, cor, ordem) VALUES
('Pendente', 'Tarefa criada mas não iniciada', '#6c757d', 1),
('Em Andamento', 'Tarefa sendo executada', '#007bff', 2),
('Em Revisão', 'Tarefa aguardando revisão', '#ffc107', 3),
('Concluída', 'Tarefa finalizada com sucesso', '#28a745', 4),
('Cancelada', 'Tarefa cancelada', '#dc3545', 5);
```

```
-- =====
-- 5. TABELA DE RÓTULOS/TAGS
-- =====
```

```
CREATE TABLE rotulos (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
    nome VARCHAR(50) NOT NULL,
    descricao TEXT,
    cor VARCHAR(7) NOT NULL DEFAULT '#007bff',
    grupo_id UUID REFERENCES grupos(id) ON DELETE CASCADE,

    UNIQUE(nome, grupo_id),
```

```
CONSTRAINT rotulos_cor_check CHECK (cor ~* '^#[0-9a-f]{6}$')
);
```

```
COMMENT ON TABLE rotulos IS 'Rótulos/tags para categorizar tarefas dentro de grupos';
COMMENT ON COLUMN rotulos.grupo_id IS 'Grupo ao qual o rótulo pertence (NULL =
rótulo global)';
```

```
-- =====
-- 6. TABELA DE TAREFAS
-- =====
```

```
CREATE TABLE tarefas (
  id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
  titulo VARCHAR(200) NOT NULL,
  descricao TEXT,
  grupo_id UUID NOT NULL REFERENCES grupos(id) ON DELETE CASCADE,
  criador_id UUID NOT NULL REFERENCES usuarios(id) ON DELETE RESTRICT,
  status_id INTEGER NOT NULL REFERENCES status_tarefa(id) ON DELETE
  RESTRICT,
  prioridade INTEGER DEFAULT 2, -- 1=baixa, 2=normal, 3=alta, 4=urgente
  data_inicio DATE,
  data_vencimento DATE,
  estimativa_horas NUMERIC(5,2),
  horas_trabalhadas NUMERIC(5,2) DEFAULT 0,
  progresso INTEGER DEFAULT 0, -- Percentual de conclusão
  data_criacao TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,
  data_atualizacao TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,
  data_conclusao TIMESTAMP WITH TIME ZONE,
```

```
CONSTRAINT tarefas_prioridade_check CHECK (prioridade BETWEEN 1 AND 4),
CONSTRAINT tarefas_progresso_check CHECK (progresso BETWEEN 0 AND 100),
CONSTRAINT tarefas_datas_check CHECK (data_vencimento >= data_inicio OR
data_vencimento IS NULL),
CONSTRAINT tarefas_horas_check CHECK (estimativa_horas > 0 OR estimativa_horas
IS NULL),
CONSTRAINT tarefas_horas_trabalhadas_check CHECK (horas_trabalhadas >= 0)
);
```

```
COMMENT ON TABLE tarefas IS 'Tarefas principais do sistema organizadas por grupos';
COMMENT ON COLUMN tarefas.prioridade IS 'Prioridade: 1=baixa, 2=normal, 3=alta,
4=urgente';
COMMENT ON COLUMN tarefas.progresso IS 'Percentual de conclusão da tarefa (0-100)';
```

```
-- =====
-- 7. TABELA DE ATRIBUIÇÕES DE TAREFAS
-- =====
```

```
CREATE TABLE atribuicoes_tarefa (
  id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
  tarefa_id UUID NOT NULL REFERENCES tarefas(id) ON DELETE CASCADE,
```

```
usuario_id UUID NOT NULL REFERENCES usuarios(id) ON DELETE CASCADE,  
atribuido_por UUID NOT NULL REFERENCES usuarios(id) ON DELETE RESTRICT,  
data_atribuicao TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,  
ativo BOOLEAN DEFAULT TRUE,
```

```
    UNIQUE(tarefa_id, usuario_id)  
);
```

```
COMMENT ON TABLE atribuicoes_tarefa IS 'Relacionamento entre tarefas e usuários  
responsáveis';  
COMMENT ON COLUMN atribuicoes_tarefa.atribuido_por IS 'Usuário que fez a atribuição';
```

```
-- =====  
-- 8. TABELA DE RELACIONAMENTO TAREFAS-RÓTULOS (PIVOT)  
-- =====
```

```
CREATE TABLE tarefas_rotulos (  
    id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),  
    tarefa_id UUID NOT NULL REFERENCES tarefas(id) ON DELETE CASCADE,  
    rotulo_id UUID NOT NULL REFERENCES rotulos(id) ON DELETE CASCADE,  
  
    UNIQUE(tarefa_id, rotulo_id)  
);
```

```
COMMENT ON TABLE tarefas_rotulos IS 'Relacionamento muitos-para-muitos entre tarefas  
e rótulos';
```

```
-- =====  
-- 9. TABELA DE COMENTÁRIOS  
-- =====
```

```
CREATE TABLE comentarios (  
    id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),  
    tarefa_id UUID NOT NULL REFERENCES tarefas(id) ON DELETE CASCADE,  
    autor_id UUID NOT NULL REFERENCES usuarios(id) ON DELETE RESTRICT,  
    conteudo TEXT NOT NULL,  
    comentario_pai_id UUID REFERENCES comentarios(id) ON DELETE CASCADE, -- Para  
respostas  
    data_criacao TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,  
    data_atualizacao TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,  
    editado BOOLEAN DEFAULT FALSE,  
  
    CONSTRAINT comentarios_conteudo_check CHECK (LENGTH(TRIM(conteudo)) > 0)  
);
```

```
COMMENT ON TABLE comentarios IS 'Comentários nas tarefas com suporte a threading  
(respostas)';  
COMMENT ON COLUMN comentarios.comentario_pai_id IS 'Referência ao comentário pai  
para criar threads';
```

```

-- =====
-- 10. TABELA DE ANEXOS
-- =====
CREATE TABLE anexos (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
    tarefa_id UUID NOT NULL REFERENCES tarefas(id) ON DELETE CASCADE,
    usuario_id UUID NOT NULL REFERENCES usuarios(id) ON DELETE RESTRICT,
    nome_original VARCHAR(255) NOT NULL,
    nome_arquivo VARCHAR(255) NOT NULL, -- Nome no sistema de arquivos
    tipo_mime VARCHAR(100) NOT NULL,
    tamanho_bytes BIGINT NOT NULL,
    caminho_arquivo TEXT NOT NULL,
    data_upload TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,

    CONSTRAINT anexos_tamanho_check CHECK (tamanho_bytes > 0)
);

COMMENT ON TABLE anexos IS 'Arquivos anexados às tarefas';
COMMENT ON COLUMN anexos.nome_original IS 'Nome original do arquivo como enviado pelo usuário';
COMMENT ON COLUMN anexos.nome_arquivo IS 'Nome único gerado para o arquivo no servidor';

-- =====
-- 11. TABELA DE ATIVIDADES/AUDITORIA
-- =====
CREATE TABLE atividades (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
    tipo_entidade VARCHAR(50) NOT NULL, -- 'tarefa', 'grupo', 'usuario', etc.
    entidade_id UUID NOT NULL,
    acao VARCHAR(50) NOT NULL, -- 'criou', 'atualizou', 'deletou', etc.
    usuario_id UUID REFERENCES usuarios(id) ON DELETE SET NULL,
    grupo_id UUID REFERENCES grupos(id) ON DELETE CASCADE,
    detalhes JSONB, -- Dados específicos da ação
    ip_address INET,
    user_agent TEXT,
    data_acao TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,

    CONSTRAINT atividades_tipo_check CHECK (tipo_entidade IN ('tarefa', 'grupo', 'usuario', 'comentario', 'anexo')),
    CONSTRAINT atividades_acao_check CHECK (acao IN ('criou', 'atualizou', 'deletou', 'atribuiu', 'completou', 'comentou', 'anexou'))
);

COMMENT ON TABLE atividades IS 'Log de auditoria de todas as ações importantes do sistema';
COMMENT ON COLUMN atividades.detalhes IS 'Dados em JSON com informações específicas da ação';

```

```

-- =====
-- 12. TABELA DE NOTIFICAÇÕES
-- =====
CREATE TABLE notificacoes (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
    usuario_id UUID NOT NULL REFERENCES usuarios(id) ON DELETE CASCADE,
    tipo VARCHAR(50) NOT NULL,
    titulo VARCHAR(200) NOT NULL,
    mensagem TEXT NOT NULL,
    entidade_tipo VARCHAR(50),
    entidade_id UUID,
    lida BOOLEAN DEFAULT FALSE,
    data_criacao TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,
    data_leitura TIMESTAMP WITH TIME ZONE,

    CONSTRAINT notificacoes_tipo_check CHECK (tipo IN ('tarefa_atribuida',
'tarefa_vencendo', 'comentario_adicionado', 'tarefa_completada', 'convite_grupo'))
);

COMMENT ON TABLE notificacoes IS 'Sistema de notificações para usuários';
COMMENT ON COLUMN notificacoes.entidade_tipo IS 'Tipo da entidade relacionada à notificação';
COMMENT ON COLUMN notificacoes.entidade_id IS 'ID da entidade relacionada à notificação';

-- =====
-- ÍNDICES PARA PERFORMANCE
-- =====

-- Índices para consultas frequentes
CREATE INDEX idx_tarefas_grupo_id ON tarefas(grupo_id);
CREATE INDEX idx_tarefas_criador_id ON tarefas(criador_id);
CREATE INDEX idx_tarefas_status_id ON tarefas(status_id);
CREATE INDEX idx_tarefas_data_vencimento ON tarefas(data_vencimento) WHERE
data_vencimento IS NOT NULL;
CREATE INDEX idx_tarefas_prioridade ON tarefas(prioridade);

CREATE INDEX idx_atribuicoes_tarefa_id ON atribuicoes_tarefa(tarefa_id);
CREATE INDEX idx_atribuicoes_usuario_id ON atribuicoes_tarefa(usuario_id);

CREATE INDEX idx_comentarios_tarefa_id ON comentarios(tarefa_id);
CREATE INDEX idx_comentarios_autor_id ON comentarios(autor_id);

CREATE INDEX idx_anexos_tarefa_id ON anexos(tarefa_id);

CREATE INDEX idx_atividades_entidade ON atividades(tipo_entidade, entidade_id);
CREATE INDEX idx_atividades_usuario_id ON atividades(usuario_id);

```



```

CREATE INDEX idx_atividades_data ON atividades(data_acao);

CREATE INDEX idx_notificacoes_usuario_id ON notificacoes(usuario_id);
CREATE INDEX idx_notificacoes_lida ON notificacoes(lida);

-- Índice composto para busca de tarefas por grupo e status
CREATE INDEX idx_tarefas_grupo_status ON tarefas(grupo_id, status_id);

-- =====
-- TRIGGERS PARA ATUALIZAÇÃO AUTOMÁTICA
-- =====

-- Função para atualizar data_atualizacao
CREATE OR REPLACE FUNCTION atualizar_timestamp()
RETURNS TRIGGER AS $$
BEGIN
    NEW.data_atualizacao = CURRENT_TIMESTAMP;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

-- Triggers para atualizar automaticamente data_atualizacao
CREATE TRIGGER trigger_usuarios_updated_at
    BEFORE UPDATE ON usuarios
    FOR EACH ROW
    EXECUTE FUNCTION atualizar_timestamp();

CREATE TRIGGER trigger_grupos_updated_at
    BEFORE UPDATE ON grupos
    FOR EACH ROW
    EXECUTE FUNCTION atualizar_timestamp();

CREATE TRIGGER trigger_tarefas_updated_at
    BEFORE UPDATE ON tarefas
    FOR EACH ROW
    EXECUTE FUNCTION atualizar_timestamp();

CREATE TRIGGER trigger_comentarios_updated_at
    BEFORE UPDATE ON comentarios
    FOR EACH ROW
    EXECUTE FUNCTION atualizar_timestamp();

-- =====
-- DADOS DE EXEMPLO
-- =====

-- Usuários de exemplo
INSERT INTO usuarios (id, nome, email, senha_hash, bio) VALUES

```

```
(uuid_generate_v4(), 'João Silva', 'joao@empresa.com',
'$2y$10$92IXUNpkjO0rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi', 'Desenvolvedor Full
Stack'),
(uuid_generate_v4(), 'Maria Santos', 'maria@empresa.com',
'$2y$10$92IXUNpkjO0rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi', 'Designer UX/UI'),
(uuid_generate_v4(), 'Pedro Oliveira', 'pedro@empresa.com',
'$2y$10$92IXUNpkjO0rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi', 'Gerente de
Projetos');
```

-- Grupos de exemplo

```
INSERT INTO grupos (id, nome, descricao, criador_id)
SELECT uuid_generate_v4(), 'Desenvolvimento Web', 'Projeto de desenvolvimento do novo
site da empresa', u.id
FROM usuarios u WHERE u.email = 'joao@empresa.com' LIMIT 1;
```

```
INSERT INTO grupos (id, nome, descricao, criador_id)
SELECT uuid_generate_v4(), 'Marketing Digital', 'Campanhas de marketing e redes sociais',
u.id
FROM usuarios u WHERE u.email = 'maria@empresa.com' LIMIT 1;
```

-- Adicionar usuários aos grupos

```
INSERT INTO usuarios_grupos (usuario_id, grupo_id, papel)
SELECT u.id, g.id, 'admin'
FROM usuarios u, grupos g
WHERE u.email = 'joao@empresa.com' AND g.nome = 'Desenvolvimento Web';
```

```
INSERT INTO usuarios_grupos (usuario_id, grupo_id, papel)
SELECT u.id, g.id, 'membro'
FROM usuarios u, grupos g
WHERE u.email = 'maria@empresa.com' AND g.nome = 'Desenvolvimento Web';
```

-- Rótulos de exemplo

```
INSERT INTO rotulos (nome, descricao, cor, grupo_id)
SELECT 'Frontend', 'Desenvolvimento do frontend', '#007bff', g.id
FROM grupos g WHERE g.nome = 'Desenvolvimento Web';
```

```
INSERT INTO rotulos (nome, descricao, cor, grupo_id)
SELECT 'Backend', 'Desenvolvimento do backend', '#28a745', g.id
FROM grupos g WHERE g.nome = 'Desenvolvimento Web';
```

-- Tarefas de exemplo

```
INSERT INTO tarefas (titulo, descricao, grupo_id, criador_id, status_id, prioridade,
data_vencimento)
SELECT
  'Criar wireframes das páginas principais',
  'Desenvolver wireframes para as páginas: home, sobre, contato e produtos',
  g.id,
  u.id,
```

```
s.id,  
3,  
CURRENT_DATE + INTERVAL '7 days'  
FROM grupos g, usuarios u, status_tarefa s  
WHERE g.nome = 'Desenvolvimento Web'  
AND u.email = 'joao@empresa.com'  
AND s.nome = 'Pendente';
```

```
INSERT INTO tarefas (titulo, descricao, grupo_id, criador_id, status_id, prioridade,  
data_vencimento)
```

```
SELECT  
'Implementar sistema de autenticação',  
'Desenvolver sistema completo de login, registro e recuperação de senha',  
g.id,  
u.id,  
s.id,  
4,  
CURRENT_DATE + INTERVAL '14 days'  
FROM grupos g, usuarios u, status_tarefa s  
WHERE g.nome = 'Desenvolvimento Web'  
AND u.email = 'joao@empresa.com'  
AND s.nome = 'Em Andamento';
```