

Artefato 7: Projeto físico de banco de dados

Sistema de Gestão de Artefatos Scrum (SGAS)

Descrição Física do Banco de Dados

Arthur Luiz Lima de Araújo
Arthur da Silva Pereira Bispo
João Carlos Gonçalves de Oliveira Filho
Ana Luísa de Souza Paraguassu
Tauã Valentim de A. M. Frade

Sumário

1	Introdução	2
2	Diagrama Completo do Banco de Dados	3
3	Descrição das Tabelas	4
3.1	Tabelas do Framework Django	4
3.1.1	django_session	4
3.1.2	django_migrations	4
3.1.3	django_content_type	4
3.1.4	django_admin_log	5
3.2	Tabelas de Autenticação e Autorização	5
3.2.1	auth_group	5
3.2.2	auth_permission	6
3.2.3	auth_group_permissions	6
3.3	Tabelas da Aplicação	7
3.3.1	api_user	7
3.3.2	api_user_groups	7
3.3.3	api_user_user_permissions	8
3.3.4	api_project	8
3.3.5	api_projectmembership	8
3.3.6	api_userstory	9
3.3.7	api_productbacklogitem	9
3.3.8	api_sprint	10
4	Índices e Otimizações	11
4.1	Índices Implícitos (Primary Keys)	11
4.2	Índices em Foreign Keys	11
4.3	Índices Únicos	11
5	Resumo Estatístico	11

1 Introdução

Este documento apresenta o projeto físico do banco de dados para o Sistema de Gestão de Artefatos Scrum (SGAS). O objetivo principal deste artefato é detalhar a estrutura concreta de armazenamento de dados que dará suporte à aplicação, servindo como um guia para a implementação e manutenção do banco.

O projeto está otimizado para ser gerenciado pelo ORM (Object-Relational Mapper) do framework Django.

Nas seções a seguir, serão apresentados:

- O diagrama físico do banco de dados completo, que ilustra visualmente todas as tabelas e seus respectivos relacionamentos.
- Uma descrição textual detalhada de cada tabela, incluindo seu propósito, a definição de suas colunas (com tipos de dados e restrições), a identificação das chaves primárias e estrangeiras, e os relacionamentos.
- Uma visão geral dos índices de otimização aplicados para garantir o desempenho das consultas.

Este detalhamento é fundamental para garantir a integridade, o desempenho e a consistência dos dados do sistema SGAS.

2 Diagrama Completo do Banco de Dados

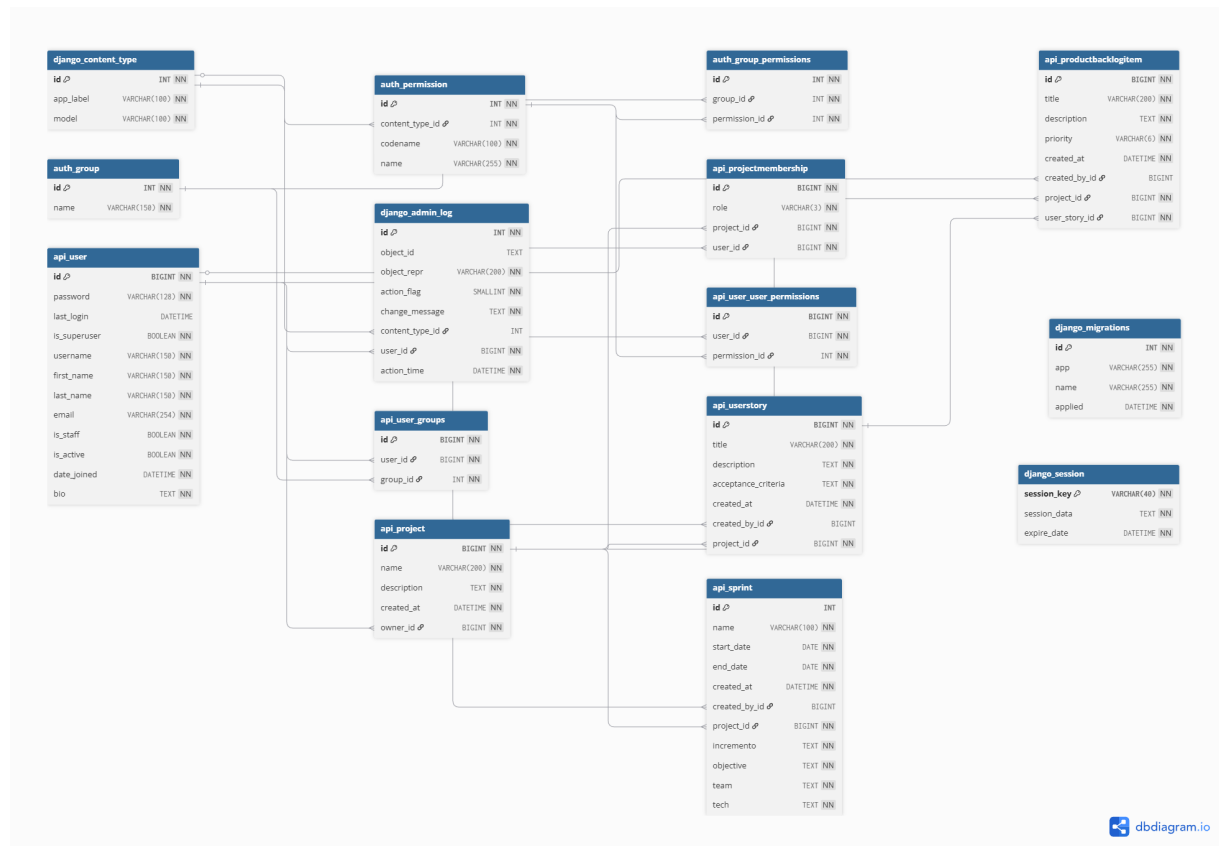


Figura 1: Diagrama Físico do Banco de Dados Completo do Sistema SGAS

3 Descrição das Tabelas

3.1 Tabelas do Framework Django

3.1.1 django_session

Propósito: Armazena sessões de usuários autenticados no sistema Django.

Coluna	Tipo	Restrições	Descrição
session_key	VARCHAR(40)	PK, NOT NULL	Chave única identificadora da sessão
session_data	TEXT	NOT NULL	Dados serializados da sessão
expire_date	DATETIME	NOT NULL	Data e hora de expiração da sessão

Tabela 1: Estrutura da tabela django_session

Chaves:

- Primary Key: session_key

Relacionamentos: Nenhum

3.1.2 django_migrations

Propósito: Registra o histórico de migrações aplicadas ao banco de dados pelo Django ORM.

Coluna	Tipo	Restrições	Descrição
id	INT	PK, AUTO-INC	Identificador único da migração
app	VARCHAR(255)	NOT NULL	Nome da aplicação Django
name	VARCHAR(255)	NOT NULL	Nome do arquivo de migração
applied	DATETIME	NOT NULL	Data e hora da aplicação

Tabela 2: Estrutura da tabela django_migrations

Chaves:

- Primary Key: id

Relacionamentos: Nenhum

3.1.3 django_content_type

Propósito: Armazena tipos de conteúdo (modelos) registrados no framework Django para o sistema de permissões.

Chaves:

- Primary Key: id

Relacionamentos:

- Referenciada por: django_admin_log, auth_permission

Coluna	Tipo	Restrições	Descrição
id	INT	PK, AUTO_INC	Identificador único do tipo
app_label	VARCHAR(100)	NOT NULL	Rótulo da aplicação Django
model	VARCHAR(100)	NOT NULL	Nome do modelo Django

Tabela 3: Estrutura da tabela django_content_type

3.1.4 django_admin_log

Propósito: Registra ações administrativas realizadas através da interface de administração do Django.

Coluna	Tipo	Restrições	Descrição
id	INT	PK, AUTO_INC	Identificador único do log
object_id	TEXT	NULL	ID do objeto afetado
object_repr	VARCHAR(200)	NOT NULL	Representação textual do objeto
action_flag	SMALLINT	NOT NULL	Tipo de ação
change_message	TEXT	NOT NULL	Mensagem descritiva
content_type_id	INT	FK, NULL	Tipo de conteúdo
user_id	BIGINT	FK, NOT NULL	Usuário que realizou a ação
action_time	DATETIME	NOT NULL	Data e hora da ação

Tabela 4: Estrutura da tabela django_admin_log

Chaves:

- Primary Key: id
- Foreign Keys:
 - content_type_id → django_content_type(id)
 - user_id → api_user(id)

Relacionamentos:

- N:1 com django_content_type
- N:1 com api_user

3.2 Tabelas de Autenticação e Autorização

3.2.1 auth_group

Propósito: Define grupos de usuários para gerenciamento de permissões coletivas.

Coluna	Tipo	Restrições	Descrição
id	INT	PK, AUTO_INC	Identificador único do grupo
name	VARCHAR(150)	NOT NULL, UNIQUE	Nome do grupo

Tabela 5: Estrutura da tabela auth_group

Chaves:

- Primary Key: `id`
- Unique: `name`

Relacionamentos:

- Referenciada por: `auth_group_permissions`, `api_user_groups`

3.2.2 auth_permission

Propósito: Define permissões específicas que podem ser atribuídas a usuários ou grupos.

Coluna	Tipo	Restrições	Descrição
<code>id</code>	INT	PK, AUTO_INC	Identificador único
<code>content_type_id</code>	INT	FK, NOT NULL	Tipo de conteúdo
<code>codename</code>	VARCHAR(100)	NOT NULL	Código da permissão
<code>name</code>	VARCHAR(255)	NOT NULL	Nome descritivo

Tabela 6: Estrutura da tabela `auth_permission`

Chaves:

- Primary Key: `id`
- Foreign Key: `content_type_id` → `django_content_type(id)`

Relacionamentos:

- N:1 com `django_content_type`
- Referenciada por: `auth_group_permissions`, `api_user_user_permissions`

3.2.3 auth_group_permissions

Propósito: Tabela de junção que associa permissões a grupos (relacionamento N:N).

Coluna	Tipo	Restrições	Descrição
<code>id</code>	INT	PK, AUTO_INC	Identificador único
<code>group_id</code>	INT	FK, NOT NULL	ID do grupo
<code>permission_id</code>	INT	FK, NOT NULL	ID da permissão

Tabela 7: Estrutura da tabela `auth_group_permissions`

Chaves:

- Primary Key: `id`
- Foreign Keys:
 - `group_id` → `auth_group(id)`
 - `permission_id` → `auth_permission(id)`

Coluna	Tipo	Restrições	Descrição
id	BIGINT	PK, AUTO_INC	Identificador único
password	VARCHAR(128)	NOT NULL	Senha criptografada
last_login	DATETIME	NULL	Data do último login
is_superuser	BOOLEAN	NOT NULL	Indica se é superusuário
username	VARCHAR(150)	NOT NULL, UNIQUE	Nome de usuário único
first_name	VARCHAR(150)	NOT NULL	Primeiro nome
last_name	VARCHAR(150)	NOT NULL	Sobrenome
email	VARCHAR(254)	NOT NULL	Endereço de e-mail
is_staff	BOOLEAN	NOT NULL	Acesso ao admin
is_active	BOOLEAN	NOT NULL	Conta está ativa
date_joined	DATETIME	NOT NULL	Data de cadastro
bio	TEXT	NOT NULL	Biografia do usuário

Tabela 8: Estrutura da tabela api_user

3.3 Tabelas da Aplicação

3.3.1 api_user

Propósito: Armazena informações de usuários do sistema, estendendo o modelo de usuário padrão do Django.

Chaves:

- Primary Key: id
- Unique: username

Relacionamentos:

- Referenciada por: django_admin_log, api_project, api_projectmembership, api_user_group, api_user_user_permissions, api_userstory, api_productbacklogitem, api_sprint

3.3.2 api_user_groups

Propósito: Tabela de junção que associa usuários a grupos (relacionamento N:N).

Coluna	Tipo	Restrições	Descrição
id	BIGINT	PK, AUTO_INC	Identificador único
user_id	BIGINT	FK, NOT NULL	ID do usuário
group_id	INT	FK, NOT NULL	ID do grupo

Tabela 9: Estrutura da tabela api_user_groups

Chaves:

- Primary Key: id
- Foreign Keys:
 - user_id → api_user(id)
 - group_id → auth_group(id)

3.3.3 api_user_user_permissions

Propósito: Tabela de junção que associa permissões individuais diretamente a usuários.

Coluna	Tipo	Restrições	Descrição
id	BIGINT	PK, AUTO_INC	Identificador único
user_id	BIGINT	FK, NOT NULL	ID do usuário
permission_id	INT	FK, NOT NULL	ID da permissão

Tabela 10: Estrutura da tabela api_user_user_permissions

Chaves:

- Primary Key: id
- Foreign Keys:
 - user_id → api_user(id)
 - permission_id → auth_permission(id)

3.3.4 api_project

Propósito: Armazena informações sobre projetos gerenciados no sistema.

Coluna	Tipo	Restrições	Descrição
id	BIGINT	PK, AUTO_INC	Identificador único
name	VARCHAR(200)	NOT NULL	Nome do projeto
description	TEXT	NOT NULL	Descrição detalhada
created_at	DATETIME	NOT NULL	Data de criação
owner_id	BIGINT	FK, NOT NULL	ID do proprietário

Tabela 11: Estrutura da tabela api_project

Chaves:

- Primary Key: id
- Foreign Key: owner_id → api_user(id)

Relacionamentos:

- N:1 com api_user (owner)
- Referenciada por: api_projectmembership, api_userstory, api_productbacklogitem, api_sprint

3.3.5 api_projectmembership

Propósito: Gerencia a associação entre usuários e projetos, definindo o papel de cada membro.

Chaves:

- Primary Key: id

Coluna	Tipo	Restrições	Descrição
id	BIGINT	PK, AUTO_INC	Identificador único
role	VARCHAR(3)	NOT NULL	Papel do membro
project_id	BIGINT	FK, NOT NULL	ID do projeto
user_id	BIGINT	FK, NOT NULL	ID do usuário

Tabela 12: Estrutura da tabela api_projectmembership

- Foreign Keys:
 - project_id → api_project(id)
 - user_id → api_user(id)

3.3.6 api_userstory

Propósito: Armazena histórias de usuário (user stories) associadas aos projetos.

Coluna	Tipo	Restrições	Descrição
id	BIGINT	PK, AUTO_INC	Identificador único
title	VARCHAR(200)	NOT NULL	Título da história
description	TEXT	NOT NULL	Descrição detalhada
acceptance_criteria	TEXT	NOT NULL	Critérios de aceitação
created_at	DATETIME	NOT NULL	Data de criação
created_by_id	BIGINT	FK, NULL	ID do criador
project_id	BIGINT	FK, NOT NULL	ID do projeto

Tabela 13: Estrutura da tabela api_userstory

Chaves:

- Primary Key: id
- Foreign Keys:
 - created_by_id → api_user(id)
 - project_id → api_project(id)

Relacionamentos:

- N:1 com api_user (creator)
- N:1 com api_project
- Referenciada por: api_productbacklogitem

3.3.7 api_productbacklogitem

Propósito: Representa itens do backlog do produto vinculados a histórias de usuário.

Chaves:

- Primary Key: id
- Foreign Keys:

Coluna	Tipo	Restrições	Descrição
id	BIGINT	PK, AUTO_INC	Identificador único
title	VARCHAR(200)	NOT NULL	Título do item
description	TEXT	NOT NULL	Descrição detalhada
priority	VARCHAR(6)	NOT NULL	Prioridade
created_at	DATETIME	NOT NULL	Data de criação
created_by_id	BIGINT	FK, NULL	ID do criador
project_id	BIGINT	FK, NOT NULL	ID do projeto
user_story_id	BIGINT	FK, NOT NULL	ID da user story

Tabela 14: Estrutura da tabela api_productbacklogitem

- created_by_id → api_user(id)
- project_id → api_project(id)
- user_story_id → api_userstory(id)

3.3.8 api_sprint

Propósito: Gerencia sprints do projeto, incluindo datas, objetivos e informações da equipe.

Coluna	Tipo	Restrições	Descrição
id	INT	PK, AUTO_INC	Identificador único
name	VARCHAR(100)	NOT NULL	Nome do sprint
start_date	DATE	NOT NULL	Data de início
end_date	DATE	NOT NULL	Data de término
created_at	DATETIME	NOT NULL	Data de criação
created_by_id	BIGINT	FK, NULL	ID do criador
project_id	BIGINT	FK, NOT NULL	ID do projeto
increment	TEXT	NOT NULL	Incremento do produto
objective	TEXT	NOT NULL	Objetivo do sprint
team	TEXT	NOT NULL	Membros da equipe
tech	TEXT	NOT NULL	Tecnologias utilizadas

Tabela 15: Estrutura da tabela api_sprint

Chaves:

- Primary Key: id
- Foreign Keys:
 - created_by_id → api_user(id) (ON DELETE SET NULL, ON UPDATE CASCADE)
 - project_id → api_project(id) (ON DELETE CASCADE, ON UPDATE CASCADE)

Relacionamentos:

- N:1 com api_user (creator)
- N:1 com api_project

4 Índices e Otimizações

4.1 Índices Implícitos (Primary Keys)

Todas as tabelas possuem índice automático na chave primária, garantindo acesso rápido por ID.

4.2 Índices em Foreign Keys

Para otimizar consultas com JOINS, foi criado índices nas seguintes colunas:

- `api_project.owner_id`
- `api_projectmembership.project_id`
- `api_projectmembership.user_id`
- `api_userstory.project_id`
- `api_userstory.created_by_id`
- `api_productbacklogitem.project_id`
- `api_productbacklogitem.user_story_id`
- `api_sprint.project_id`

4.3 Índices Únicos

- `api_user.username`
- `auth_group.name`

5 Resumo Estatístico

Métrica	Valor
Total de tabelas	15
Tabelas do Django	4
Tabelas de autenticação	3
Tabelas de aplicação	8
Relacionamentos 1:N	12
Relacionamentos N:N	4
Foreign keys	20
Índices únicos	2

Tabela 16: Estatísticas do banco de dados