

## Modelos de Dados (Entidades)

### 1. Organizadores

Representa os organizadores de eventos.

- **id** (UUID, PK): Identificador único gerado automaticamente.
- **nome** (TEXT, NOT NULL): Nome do organizador.
- **descricao** (TEXT): Descrição opcional.
- **email\_contato** (TEXT): E-mail de contato.
- **telefone\_contato** (TEXT): Telefone de contato.
- **site** (TEXT): Site do organizador.
- **criado\_em** (TIMESTAMPTZ): Data de criação (padrão: now()).
- **atualizado\_em** (TIMESTAMPTZ): Data de atualização.

### 2. Locais

Representa os locais onde os eventos ocorrem.

- **id** (UUID, PK): Identificador único gerado automaticamente.
- **nome** (TEXT, NOT NULL): Nome do local.
- **descricao** (TEXT): Descrição opcional.
- **endereco** (TEXT): Endereço completo.
- **cidade** (TEXT): Cidade.
- **estado** (TEXT): Estado.
- **pais** (TEXT): País.
- **capacidade** (INTEGER): Capacidade máxima.
- **criado\_em** (TIMESTAMPTZ): Data de criação (padrão: now()).
- **atualizado\_em** (TIMESTAMPTZ): Data de atualização.

### 3. Categorias

Representa categorias de eventos (ex.: música, esportes).

- **id** (UUID, PK): Identificador único gerado automaticamente.

- **nome** (TEXT, NOT NULL, UNIQUE): Nome da categoria.
- **slug** (TEXT, NOT NULL, UNIQUE): Slug para URLs amigáveis.

#### 4. Eventos

Representa eventos fixos (ex.: Rock in Rio).

- **id** (UUID, PK): Identificador único gerado automaticamente.
- **id\_organizador** (UUID, FK para organizadores, NOT NULL): Organizador responsável.
- **id\_local** (UUID, FK para locais, NOT NULL): Local do evento.
- **id\_categoria** (UUID, FK para categorias): Categoria opcional.
- **titulo** (TEXT, NOT NULL): Título do evento.
- **slug** (TEXT, NOT NULL, UNIQUE): Slug único.
- **descricao\_curta** (TEXT): Descrição curta.
- **descricao** (TEXT): Descrição completa.
- **status** (TEXT, NOT NULL, DEFAULT 'rascunho'): Status (rascunho, publicado, cancelado, arquivado).
- **capacidade** (INTEGER): Capacidade do evento.
- **metadados** (JSONB): Dados adicionais em JSON.
- **criado\_por** (UUID): Usuário que criou (opcional).
- **criado\_em** (TIMESTAMPTZ): Data de criação (padrão: now()).
- **atualizado\_em** (TIMESTAMPTZ): Data de atualização.

#### 5. Programacoes\_Evento

Representa programações específicas dentro de um evento (ex.: sessões de tempo).

- **id** (UUID, PK): Identificador único gerado automaticamente.
- **id\_evento** (UUID, FK para eventos, CASCADE on delete): Evento associado.
- **inicio\_em** (TIMESTAMPTZ, NOT NULL): Data/hora de início.
- **fim\_em** (TIMESTAMPTZ): Data/hora de fim.
- **titulo** (TEXT, NOT NULL): Título da programação.

- **descricao** (TEXT): Descrição.
- **capacidade** (INTEGER): Capacidade específica.
- **criado\_em** (TIMESTAMPTZ): Data de criação (padrão: now()).
- **atualizado\_em** (TIMESTAMPTZ): Data de atualização.

## 6. Etiquetas

Representa tags para eventos.

- **id** (UUID, PK): Identificador único gerado automaticamente.
- **nome** (TEXT, NOT NULL, UNIQUE): Nome da etiqueta.

## 7. Evento\_Etiquetas

Tabela de junção para relacionamento muitos-para-muitos entre eventos e etiquetas.

- **id\_evento** (UUID, FK para eventos, CASCADE on delete): Evento.
- **id\_etiqueta** (UUID, FK para etiquetas, CASCADE on delete): Etiqueta.
- **PK composta:** (id\_evento, id\_etiqueta).

## 8. Usuarios

Representa usuários do sistema.

- **id** (UUID, PK): Identificador único gerado automaticamente.
- **email** (TEXT, UNIQUE, NOT NULL): E-mail do usuário.
- **nome** (TEXT): Nome completo.
- **senha\_hash** (TEXT): Hash da senha.
- **papel** (TEXT, DEFAULT 'usuario'): Papel (usuario, admin).
- **criado\_em** (TIMESTAMPTZ): Data de criação (padrão: now()).

## 9. Ingressos

Representa tipos de ingressos para eventos.

- **id** (UUID, PK): Identificador único gerado automaticamente.
- **id\_evento** (UUID, FK para eventos, CASCADE on delete): Evento associado.
- **nome** (TEXT, NOT NULL): Nome do ingresso.

- **criado\_em** (TIMESTAMPTZ): Data de criação (padrão: now()).

## 10. Participacoes

Representa inscrições ou compras de ingressos.

- **id** (UUID, PK): Identificador único gerado automaticamente.
- **id\_usuario** (UUID, FK para usuarios, SET NULL on delete): Usuário participante.
- **id\_ingresso** (UUID, FK para ingressos, SET NULL on delete): Ingresso comprado.
- **id\_evento** (UUID, FK para eventos, CASCADE on delete): Evento.
- **criado\_em** (TIMESTAMPTZ): Data de criação (padrão: now()).

## 11. Comentarios

Representa comentários e avaliações de eventos (suporta threads).

- **id** (UUID, PK): Identificador único gerado automaticamente.
- **id\_evento** (UUID, FK para eventos, CASCADE on delete): Evento comentado.
- **id\_usuario** (UUID, FK para usuarios, SET NULL on delete): Usuário que comentou.
- **id\_comentario\_pai** (UUID, FK para comentarios, CASCADE on delete): Comentário pai (para threads).
- **conteudo** (TEXT, NOT NULL): Conteúdo do comentário.
- **avaliacao** (SMALLINT): Avaliação (ex.: 1-5 estrelas).
- **criado\_em** (TIMESTAMPTZ): Data de criação (padrão: now()).

## 12. Imagens\_Evento

Representa imagens associadas a eventos.

- **id** (UUID, PK): Identificador único gerado automaticamente.
- **id\_evento** (UUID, FK para eventos, CASCADE on delete): Evento.
- **url** (TEXT, NOT NULL): URL da imagem.
- **descricao\_alternativa** (TEXT): Descrição alternativa.
- **capa** (BOOLEAN, DEFAULT false): Se é imagem de capa.

- **criado\_em** (TIMESTAMPZ): Data de criação (padrão: now()).

### 13. Favoritos

Representa favoritos de usuários para eventos (relacionamento muitos-para-muitos).

- **id\_usuario** (UUID, FK para usuarios, CASCADE on delete): Usuário.
- **id\_evento** (UUID, FK para eventos, CASCADE on delete): Evento.
- **PK composta:** (id\_usuario, id\_evento).

### Principais Funcionalidades da API

A API segue padrões REST, com endpoints CRUD para cada entidade. Todos os endpoints requerem autenticação JWT (exceto login/registro). Respostas incluem códigos HTTP apropriados (200, 201, 400, 401, 404, 500). Usa DTOs para entrada/saída, validação de dados e paginação/ordenação/filtros em listagens. Documentação completa via Swagger em /api-docs.

### Autenticação

- **POST /auth/login:** Login com email/senha, retorna JWT.
- **POST /auth/register:** Registro de novo usuário.
- **Middleware JWT:** Protege rotas, verifica papéis.

### Endpoints CRUD por Entidade

Cada entidade tem endpoints padrão. Exemplos para "Eventos" (similar para outras):

- **GET /eventos:** Lista eventos com paginação (query params: page, limit), ordenação (sortBy, order) e filtros (ex.: status, categoria).
- **GET /eventos/:id:** Busca um evento por ID.
- **POST /eventos:** Cria novo evento.
- **PUT /eventos/:id:** Atualiza evento.
- **DELETE /eventos/:id:** Deleta evento.

Outras entidades seguem o mesmo padrão: /organizadores, /locais, /categorias, /programacoes, /etiquetas, /usuarios, /ingressos, /participacoes, /comentarios, /imagens, /favoritos.

### Funcionalidades Específicas

- **Avaliações/Comentários:** Suporte a threads em comentários.
- **Favoritos:** POST/DELETE /usuarios/:id/favoritos/:eventoid para gerenciar.
- **Exportação CSV:** GET /entidade/export?format=csv (ex.:  
/eventos/export?format=csv) para download de dados em CSV.
- **Imagens:** Upload e gerenciamento de imagens para eventos.