Banco de Dados – EP 2/2025 – Fase 2

Projeto: TikEvents – Gestão de Eventos

Modelo do Banco de Dados

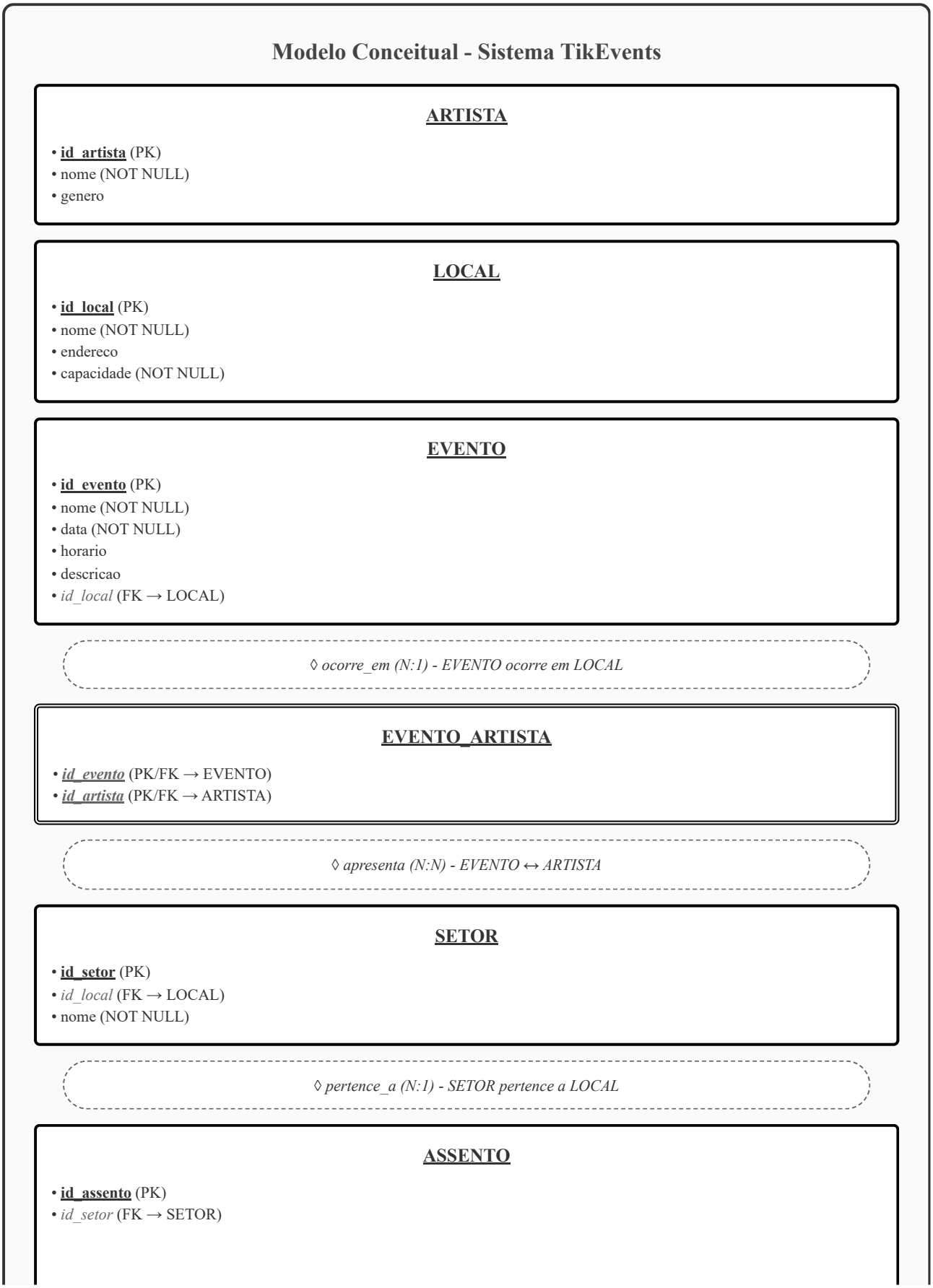
Equipe:

Átila Aroso Soares (14745546)

Kawe da Cruz Gomes (11838839)

Vitor Pazos (14611814)

1. Diagrama Entidade-Relacionamento (DER)



• fileira (NOT NULL)
• numero (NOT NULL)

◊ *pertence_a (N:1) - ASSENTO pertence a SETOR*

INGRESSO

• **id_ingresso** (PK)
• *id_evento* (FK → EVENTO)
• preco (NOT NULL)
• *id_assento* (FK → ASSENTO) [opcional]

◊ *gera (1:N) - EVENTO gera INGRESSO*

◊ *reserva (0:1) - INGRESSO pode reservar ASSENTO*

⊷ Especialização Total e Disjunta

INGRESSO_VIP

• ***id_ingresso*** (PK/FK → INGRESSO)
• beneficios

INGRESSO_PADRAO

• ***id_ingresso*** (PK/FK → INGRESSO)

COMPRADOR

• **id_comprador** (PK)
• nome (NOT NULL)
• email (UNIQUE, NOT NULL)

VENDA

• **id_venda** (PK)
• data (NOT NULL)
• quantidade (NOT NULL)
• *id_ingresso* (FK → INGRESSO)
• *id_comprador* (FK → COMPRADOR)

◊ *refere_se (N:1) - VENDA refere-se a INGRESSO*

◊ *realizada_por (N:1) - VENDA realizada por COMPRADOR*

Legenda: Retângulos = Entidades | Retângulos duplos = Entidades fracas | ◊ = Relacionamentos | ⊷ = Especialização | PK = Chave Primária | FK = Chave Estrangeira

2. Modelo Relacional

Tabela	Esquema Relacional
ARTISTA	ARTISTA(**id_artista**, nome NOT NULL, genero)
LOCAL	LOCAL(**id_local**, nome NOT NULL, endereco, capacidade NOT NULL CHECK(capacidade > 0))
SETOR	SETOR(**id_setor**, *id_local* NOT NULL, nome NOT NULL) FK: id_local → LOCAL(id_local) ON DELETE CASCADE UNIQUE(id_local, nome)
ASSENTO	ASSENTO(**id_assento**, *id_setor* NOT NULL, fileira NOT NULL, numero NOT NULL) FK: id_setor → SETOR(id_setor) ON DELETE CASCADE UNIQUE(id_setor, fileira, numero)
EVENTO	EVENTO(**id_evento**, nome NOT NULL, data NOT NULL, horario, descricao, *id_local* NOT NULL) FK: id_local → LOCAL(id_local) ON DELETE RESTRICT
EVENTO_ARTISTA	EVENTO_ARTISTA(***id_evento***, ***id_artista***) FK: id_evento → EVENTO(id_evento) ON DELETE CASCADE FK: id_artista → ARTISTA(id_artista) ON DELETE CASCADE PK: (id_evento, id_artista)
INGRESSO	INGRESSO(**id_ingresso**, *id_evento* NOT NULL, preco NOT NULL CHECK(preco >= 0), *id_assento*) FK: id_evento → EVENTO(id_evento) ON DELETE CASCADE FK: id_assento → ASSENTO(id_assento) ON DELETE SET NULL UNIQUE(id_evento, id_assento)
INGRESSO_VIP	INGRESSO_VIP(***id_ingresso***, beneficios) FK/PK: id_ingresso → INGRESSO(id_ingresso) ON DELETE CASCADE
INGRESSO_PADRAO	INGRESSO_PADRAO(***id_ingresso***) FK/PK: id_ingresso → INGRESSO(id_ingresso) ON DELETE CASCADE
COMPRADOR	COMPRADOR(**id_comprador**, nome NOT NULL, email VARCHAR(100) UNIQUE NOT NULL)
VENDA	VENDA(**id_venda**, data NOT NULL, quantidade NOT NULL CHECK(quantidade > 0), *id_ingresso* NOT NULL, *id_comprador* NOT NULL) FK: id_ingresso → INGRESSO(id_ingresso) ON DELETE RESTRICT FK: id_comprador → COMPRADOR(id_comprador) ON DELETE RESTRICT

3. Dicionário de Dados

Tabela	Atributo	Tipo	Nulo?	Chave/Regra	Descrição
ARTISTA					
ARTISTA	id_artista	SERIAL	Não	PK	Identificador único do artista
ARTISTA	nome	VARCHAR(100)	Não	NOT NULL	Nome artístico ou banda
ARTISTA	genero	VARCHAR(50)	Sim	-	Gênero musical principal
LOCAL					
LOCAL	id_local	SERIAL	Não	PK	Identificador único do local
LOCAL	nome	VARCHAR(100)	Não	NOT NULL	Nome do estabelecimento
LOCAL	endereco	VARCHAR(150)	Sim	-	Endereço completo
LOCAL	capacidade	INT	Não	CHECK(capacidade>0)	Capacidade máxima
SETOR					
SETOR	id_setor	SERIAL	Não	PK	Identificador do setor
SETOR	id_local	INT	Não	FK→LOCAL	Local ao qual pertence
SETOR	nome	VARCHAR(60)	Não	UNIQUE(id_local,nome)	Nome do setor
ASSENTO					
ASSENTO	id_assento	SERIAL	Não	PK	Identificador do assento
ASSENTO	id_setor	INT	Não	FK→SETOR	Setor ao qual pertence
ASSENTO	fileira	VARCHAR(10)	Não	-	Identificação da fileira
ASSENTO	numero	VARCHAR(10)	Não	UNIQUE(id_setor,fileira,numero)	Número na fileira
EVENTO					
EVENTO	id_evento	SERIAL	Não	PK	Identificador do evento
EVENTO	nome	VARCHAR(100)	Não	NOT NULL	Nome/título do evento
EVENTO	data	DATE	Não	NOT NULL	Data de realização
EVENTO	horario	TIME	Sim	-	Horário de início
EVENTO	descricao	TEXT	Sim	-	Descrição detalhada

Tabela	Atributo	Tipo	Nulo?	Chave/Regra	Descrição
EVENTO	id_local	INT	Não	FK→LOCAL	Local do evento
EVENTO_ARTISTA					
EVENTO_ARTISTA	id_evento	INT	Não	PK/FK→EVENTO	Evento da apresentação
EVENTO_ARTISTA	id_artista	INT	Não	PK/FK→ARTISTA	Artista que se apresenta
INGRESSO					
INGRESSO	id_ingresso	SERIAL	Não	PK	Identificador do ingresso
INGRESSO	id_evento	INT	Não	FK→EVENTO	Evento relacionado
INGRESSO	preco	NUMERIC(10,2)	Não	CHECK(preco>=0)	Valor em reais
INGRESSO	id_assento	INT	Sim	FK→ASSENTO, UNIQUE(id_evento,id_assento)	Assento (se numerado)
INGRESSO_VIP					
INGRESSO_VIP	id_ingresso	INT	Não	PK/FK→INGRESSO	Referência ao ingresso
INGRESSO_VIP	beneficios	TEXT	Sim	-	Benefícios VIP inclusos
INGRESSO_PADRAO					
INGRESSO_PADRAO	id_ingresso	INT	Não	PK/FK→INGRESSO	Referência ao ingresso
COMPRADOR					
COMPRADOR	id_comprador	SERIAL	Não	PK	Identificador do comprador
COMPRADOR	nome	VARCHAR(100)	Não	NOT NULL	Nome completo
COMPRADOR	email	VARCHAR(100)	Não	UNIQUE, NOT NULL	E-mail para contato
VENDA					
VENDA	id_venda	SERIAL	Não	PK	Identificador da venda
VENDA	data	DATE	Não	NOT NULL	Data da transação
VENDA	quantidade	INT	Não	CHECK(quantidade>0)	Qtd de ingressos
VENDA	id_ingresso	INT	Não	FK→INGRESSO	Tipo de ingresso
VENDA	id_comprador	INT	Não	FK→COMPRADOR	Comprador responsável

4. Script SQL para Criação das Tabelas

```
-- ================================================
-- Script de criação do banco de dados TikEvents
-- Sistema de Gestão de Eventos
-- SGBD: PostgreSQL 14+
-- ================================================

-- Tabela de artistas/bandas
CREATE TABLE Artista (
    id_artista SERIAL PRIMARY KEY,
    nome VARCHAR(100) NOT NULL,
    genero VARCHAR(50)
);

-- Tabela de locais de eventos
CREATE TABLE Local (
    id_local SERIAL PRIMARY KEY,
    nome VARCHAR(100) NOT NULL,
    endereco VARCHAR(150),
    capacidade INT NOT NULL CHECK (capacidade > 0)
);

-- Tabela de setores dentro dos locais
CREATE TABLE Setor (
    id_setor SERIAL PRIMARY KEY,
    id_local INT NOT NULL REFERENCES Local(id_local) ON DELETE CASCADE,
    nome VARCHAR(60) NOT NULL,
    UNIQUE (id_local, nome)
);

-- Tabela de assentos
CREATE TABLE Assento (
    id_assento SERIAL PRIMARY KEY,
    id_setor INT NOT NULL REFERENCES Setor(id_setor) ON DELETE CASCADE,
    fileira VARCHAR(10) NOT NULL,
    numero VARCHAR(10) NOT NULL,
    UNIQUE (id_setor, fileira, numero)
);

-- Tabela de eventos
CREATE TABLE Evento (
    id_evento SERIAL PRIMARY KEY,
    nome VARCHAR(100) NOT NULL,
    data DATE NOT NULL,
    horario TIME,
    descricao TEXT,
    id_local INT NOT NULL REFERENCES Local(id_local)
);

-- Tabela associativa Evento-Artista (N:N)
CREATE TABLE Evento_Artista (
    id_evento INT NOT NULL REFERENCES Evento(id_evento) ON DELETE CASCADE,
    id_artista INT NOT NULL REFERENCES Artista(id_artista) ON DELETE CASCADE,
    PRIMARY KEY (id_evento, id_artista)
);

-- Tabela de ingressos (superclasse)
CREATE TABLE Ingresso (
    id_ingresso SERIAL PRIMARY KEY,
    id_evento INT NOT NULL REFERENCES Evento(id_evento) ON DELETE CASCADE,
    preco NUMERIC(10,2) NOT NULL CHECK (preco >= 0),
    id_assento INT NULL REFERENCES Assento(id_assento),
    CONSTRAINT uq_evento_assento UNIQUE (id_evento, id_assento)
);
```

```
-- Especialização: Ingresso VIP
CREATE TABLE Ingresso_VIP (
    id_ingresso INT PRIMARY KEY REFERENCES Ingresso(id_ingresso) ON DELETE CASCADE,
    beneficios TEXT
);

-- Especialização: Ingresso Padrão
CREATE TABLE Ingresso_Padrao (
    id_ingresso INT PRIMARY KEY REFERENCES Ingresso(id_ingresso) ON DELETE CASCADE
);

-- Tabela de compradores
CREATE TABLE Comprador (
    id_comprador SERIAL PRIMARY KEY,
    nome VARCHAR(100) NOT NULL,
    email VARCHAR(100) UNIQUE NOT NULL
);

-- Tabela de vendas
CREATE TABLE Venda (
    id_venda SERIAL PRIMARY KEY,
    data DATE NOT NULL,
    quantidade INT NOT NULL CHECK (quantidade > 0),
    id_ingresso INT NOT NULL REFERENCES Ingresso(id_ingresso),
    id_comprador INT NOT NULL REFERENCES Comprador(id_comprador)
);

-- Índices para otimização
CREATE INDEX idx_evento_data ON Evento(data);
CREATE INDEX idx_evento_local ON Evento(id_local);
CREATE INDEX idx_ingresso_evento ON Ingresso(id_evento);
CREATE INDEX idx_venda_data ON Venda(data);
CREATE INDEX idx_venda_comprador ON Venda(id_comprador);
CREATE INDEX idx_comprador_email ON Comprador(email);
```

Observações e Regras de Negócio

- **Especialização Total e Disjunta:** Todo ingresso deve ser classificado como VIP ou Padrão, garantindo segmentação clara para relatórios e precificação diferenciada.
- **Relacionamento N:N (Evento-Artista):** Permite múltiplos artistas por evento (festivais) e que artistas participem de vários eventos ao longo do tempo.
- **Assento Opcional:** O campo id_assento em INGRESSO pode ser NULL para suportar eventos tipo "pista" ou área livre, mantendo flexibilidade do sistema.
- **Integridade Referencial:** ON DELETE CASCADE aplicado onde apropriado (ex: deletar evento remove ingressos), ON DELETE RESTRICT para prevenir exclusões indevidas.
- **Constraints de Validação:** CHECK constraints garantem dados válidos (capacidade > 0, preco >= 0, quantidade > 0).
- **Unicidade:** Constraints UNIQUE previnem duplicações (email do comprador único, mesmo assento não pode ser vendido duas vezes para o mesmo evento).

- **Índices:** Criados em colunas frequentemente utilizadas em consultas (datas, chaves estrangeiras, email) para otimizar performance.