

Projeto final - Banco de Dados

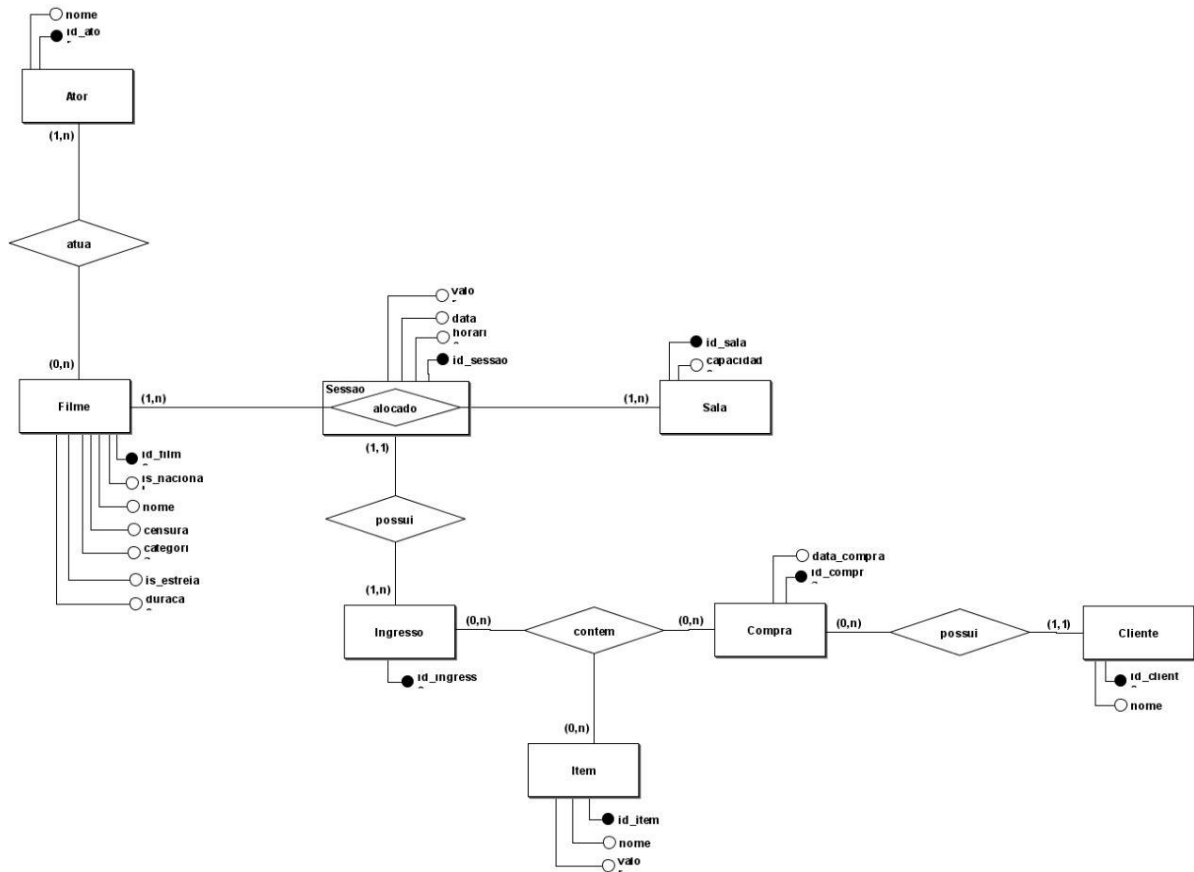
Equipe:

João Cosmo de Brito Filho / 20190024556

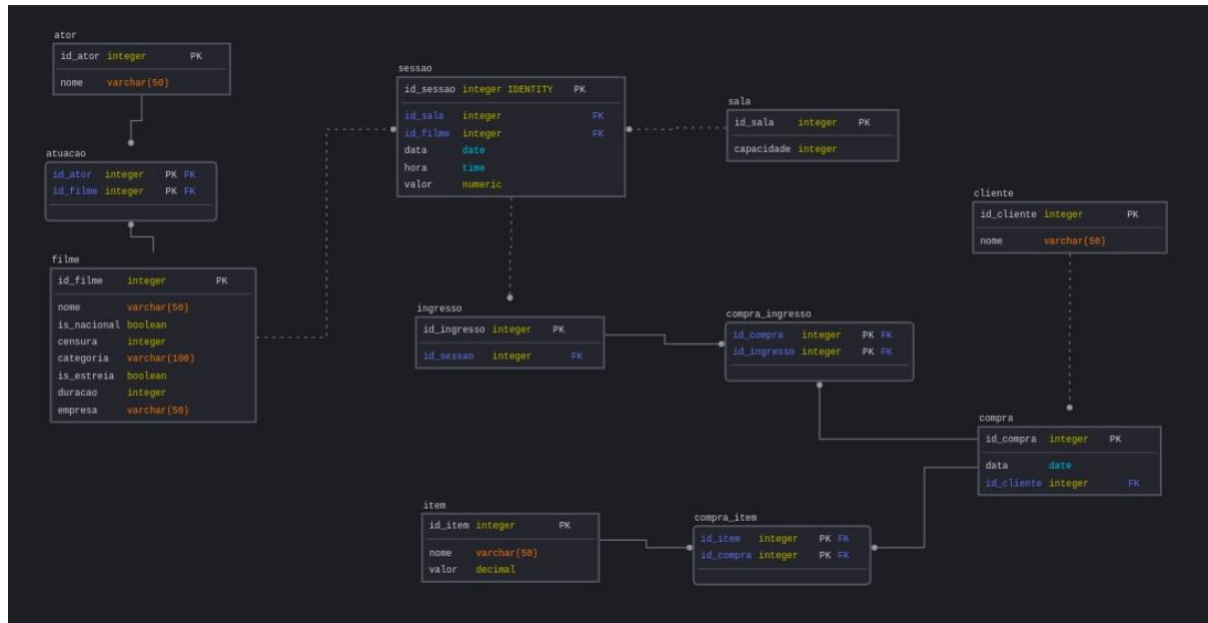
Mirele da Silva Costa / 20190022936

Raissa da Silva Vieira / 20190027870

1. Modelagem Entidade-Relacionamento



2. Modelo relacional



3. Possíveis consultas

- **Filme:**
 - Listagem de todos os filmes (estreias ou não)
 - Listagem de todas as estreias
 - Cadastro de filme
- **Sessão:**
 - Listagem de sessão por filme
 - Listagem de sessão por dia
 - Cadastro de sessão
 -
- **Sala:**
 - Listagem de salas
 - Cadastro de salas
 - Busca da capacidade de uma sala
- **Ingresso:**
 - Contagem de ingressos de uma determinada sessão
 - Cadastro de ingresso
- **Cliente:**
 - Cadastro de cliente
- **Item:**
 - Cadastro de item
 - Listagem de todos os itens disponíveis
- **Atores:**
 - Listagem de todos os atores que atuam de um filme
 - Cadastro de atores
- **Compra:**
 - Cadastro de compra

4. DDL

- **Criação do database**

CREATE DATABASE cinemasauo;

- **Criação das tabelas**

- **Filme:**

```
CREATE TABLE filme (  
  id SERIAL PRIMARY KEY,  
  nome VARCHAR(60) NOT NULL,  
  censura INTEGER NOT NULL,  
  duracao INTEGER NOT NULL,  
  categoria VARCHAR(30) NOT NULL,  
  empresa VARCHAR(20) NOT NULL,  
  isNacional BOOLEAN NOT NULL,  
  isEstreia BOOLEAN DEFAULT false  
);
```

- **Ator:**

```
CREATE TABLE ator (  
  id SERIAL PRIMARY KEY,  
  nome VARCHAR(40) NOT NULL  
);
```

- **Atuação:**

```
CREATE TABLE atuacao (  
  atorId INTEGER NOT NULL,  
  filmeId INTEGER NOT NULL,  
  CONSTRAINT fkAtor  
    FOREIGN KEY(atorId)  
      REFERENCES ator(id),  
  CONSTRAINT fkFilme  
    FOREIGN KEY(filmeId)  
      REFERENCES filme(id)  
);
```

- **Sala:**

```
CREATE TABLE sala(  
  id SERIAL PRIMARY KEY,  
  capacidade INTEGER NOT NULL  
);
```

- **Sessão:**

```
CREATE TABLE sessao(  
  id SERIAL PRIMARY KEY,  
  salaId INTEGER NOT NULL,  
  filmeId INTEGER NOT NULL,  
  CONSTRAINT fkSala  
    FOREIGN KEY(salaId)  
      REFERENCES sala(id),  
  CONSTRAINT fkFilme  
    FOREIGN KEY(filmeId)  
      REFERENCES filme(id),  
  valor NUMERIC(5,2) NOT NULL,  
  data DATE NOT NULL,  
  hora TIME NOT NULL  
);
```

- **Item:**

```
CREATE TABLE item (  
  id SERIAL PRIMARY KEY,  
  valor REAL NOT NULL,
```

```

        nome VARCHAR(30) NOT NULL
    )
-   Cliente:
    CREATE TABLE cliente (
        id SERIAL UNIQUE NOT NULL,
        nome VARCHAR(50)
    )
-   Compra:
    CREATE TABLE compra (
        id SERIAL UNIQUE NOT NULL,
        data_compra TIMESTAMP NOT NULL,
        clienteld INT NOT NULL,

        FOREIGN KEY (clienteld) REFERENCES cliente(id)
    )
-   Ingresso:
    CREATE TABLE ingresso(
        id SERIAL PRIMARY KEY,
        sessaold INTEGER NOT NULL,
        CONSTRAINT fkSessao
            FOREIGN KEY(sessaold)
                REFERENCES sessao(id)
    );
-   Compra item:
    CREATE TABLE compra_item (
        itemld INT NOT NULL,
        comprald INT NOT NULL,

        FOREIGN KEY (itemld ) REFERENCES item(id),
        FOREIGN KEY (comprald) REFERENCES compra(id)
    );
-   Compra ingresso:
    CREATE TABLE compra_ingresso (
        ingressold INT NOT NULL,
        comprald INT NOT NULL,

        FOREIGN KEY (ingressold) REFERENCES ingresso(id),
        FOREIGN KEY (comprald) REFERENCES compra(id)
    );

```

5. DML

- **Cadastro de filme:**
 INSERT INTO filme (nome, censura, duracao, categoria, empresa, isNacional, isEstreia)
 VALUES
 ('Pantera Negra: Wakanda Para Sempre',14,161,'Ação','Marvel Studios',false,true),
 ('Vingadores: Ultimato',12,182,'Ação','Marvel Studios',false,false),
 ('Minha Mãe é Uma Peça',12,84,'Comédia','Telecine Productions',true,false);
- **Cadastro de sala:**
 INSERT INTO sala (capacidade) VALUES (2);
 INSERT INTO sala (capacidade) VALUES (4);
- **Cadastro de item:**
 INSERT INTO item (valor, nome) VALUES (30, 'Pipoca');

```
INSERT INTO item (valor, nome) VALUES (15, 'Coca-cola 1L');
INSERT INTO item (valor, nome) VALUES (7, 'Coca-cola Lata');
INSERT INTO item (valor, nome) VALUES (10, 'Fini Amora');
INSERT INTO item (valor, nome) VALUES (12, 'Fini Banana');
INSERT INTO item (valor, nome) VALUES (6, 'Agua');
```

- **Cadastro de cliente:**

```
INSERT INTO cliente (nome) VALUES ($1) RETURNING id;
```

- **Cadastro de item:**

```
INSERT INTO item (valor, nome) VALUES ($1, $2);
```

- Cadastro de filme:

```
INSERT INTO filme (nome, censura, duracao, categoria, empresa, isNacional, isEstreia)
VALUES ($1, $2, $3, $4, $5, $6, $7) RETURNING id;
```

6. DQL

- **Selecionar todos os itens:**

```
SELECT * FROM item;
```

- **Selecionar todos os filmes de um ator:**

```
SELECT ator.nome FROM ator, atuacao WHERE ator.id=atuacao.atorId AND
atuacao.filmId=$1;
```

- **Listar todas as estreias:**

```
SELECT * FROM filme WHERE isEstreia=true;
```

- **Listagem de todos os filmes:**

```
SELECT * FROM filme;
```

-