1. INTRODUÇÃO

Neste trabalho apresenta uma análise de dados sobre filmes e suas avaliações, integrando o uso de banco de dados relacional (PostgreSQL), linguagem SQL e visualização de dados no Power BI.
O objetivo é investigar o perfil dos filmes mais bem avaliados e/ou mais populares, considerando principalmente gêneros, notas médias, número de votos e evolução temporal dos lançamentos.

A proposta está alinhada à atividade da disciplina de Análise de Dados com Banco de Dados, que exige a escolha de um conjunto de dados real, a modelagem das tabelas em um banco relacional, a aplicação de consultas SQL de seleção, junção e agregação e, por fim, a construção de dashboards interativos.
Neste contexto, buscamos responder perguntas como:

- Quais gêneros de filmes são mais frequentes e quais apresentam melhores avaliações?
- Quantos filmes são lançados por ano e como isso evolui ao longo do tempo?
- Quais são os filmes com maiores notas e quais são os mais votados?
- Quantos diretores distintos aparecem no conjunto de dados e quais são os que mais produziram filmes?

Nas seções seguintes descrevemos o conjunto de dados utilizado, o modelo de banco de dados adotado, as principais consultas SQL elaboradas e os dashboards desenvolvidos no Power BI, finalizando com os principais insights obtidos.

2. CONJUNTO DE DADOS

O conjunto de dados utilizado tem como base o dataset TMDB 5000 Movies, disponibilizado na plataforma Kaggle. Esse dataset reúne informações de mais de 4.800 filmes, com 12 colunas principais descrevendo características gerais e de avaliação.

Entre os atributos mais relevantes para este trabalho, destacam-se:

`title` e `original_title`: título do filme (local) e título original;
`overview`: breve sinopse do filme;
`release_date`: data de lançamento;
`original_language`: idioma original;
`status`: status do filme (por exemplo, Released);
`vote_average`: nota média atribuída pelos usuários;
`vote_count`: quantidade de votos recebidos;
`genres`: lista de gêneros associados ao filme;
`cast`: principais atores;
`director`: diretor responsável;
`homepage`: página oficial do filme.

O arquivo foi fornecido em formato .xlsx. Antes da carga no banco, realizamos uma padronização na coluna de gêneros, utilizando vírgula e espaço como separador entre gêneros e underline (_) para unir termos compostos (por exemplo, Science_Fiction, TV_Movie). Essa etapa facilitou a posterior separação dos gêneros e a modelagem relacional.

3. MODELAGEM DO BANCO DE DADOS

Para organizar os dados de forma consistente, projetamos um modelo relacional no PostgreSQL com quatro tabelas principais:

`Movies`: armazena os dados de cada filme, com atributos como:

`id_movie` (chave primária);
`title, original_title`;
`overview`;
`release_date`;
`status`;
`vote_average` e `vote_count`;
`actors` (elenco principal);
`name` (nome do diretor).
`original_language`;
`homepage`.

Directors: tabela de dimensão para diretores:

`id_director` (chave primária);
`name` (nome do diretor).

Genres: tabela de dimensão para gêneros:

`id_genre (chave primária);`
`genre` (nome do gênero, como Drama, Action, Science_Fiction etc.).

Relations: tabela associativa que liga filmes, diretores e gêneros:

`id_relations` (chave primária);
`id_movie` (FK para Movies);
`id_director` (FK para Directors);
`id_genre` (FK para Genres).

Esse desenho permite representar adequadamente relações muitos-para-muitos: um filme pode ter vários gêneros e um mesmo diretor pode ter múltiplos filmes.

A Figura 1 apresenta o diagrama entidade-relacionamento (DER) do banco de dados, com as chaves primárias, chaves estrangeiras e relacionamentos entre as quatro tabelas.

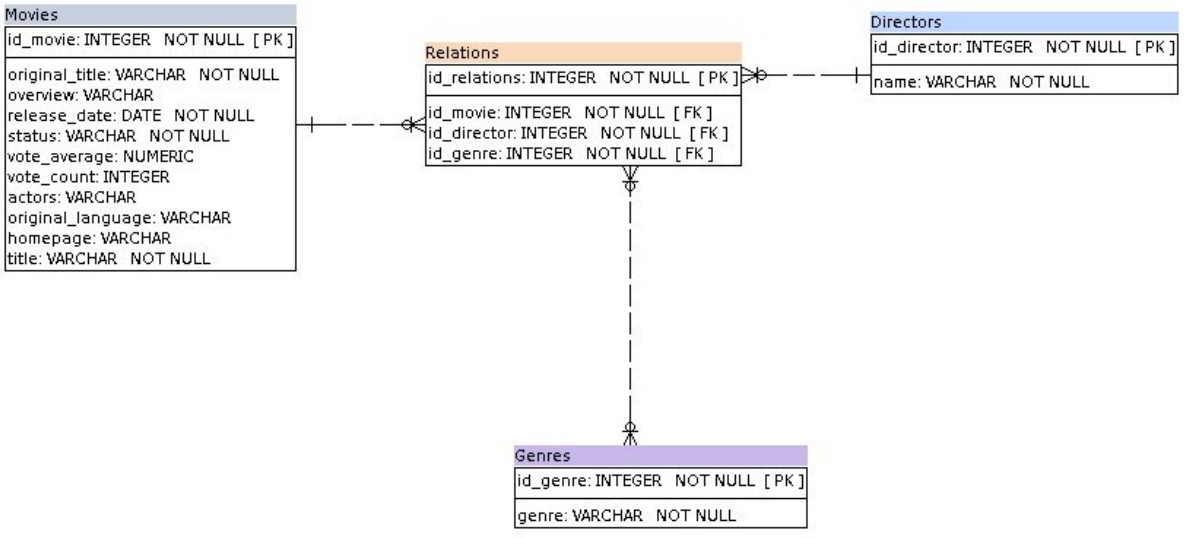

Figura 1 – Diagrama entidade-relacionamento do banco de dados de filmes.

4. PREPARAÇÃO E TRANSFORMAÇÃO DOS DADOS

A carga inicial do dataset para o PostgreSQL foi realizada com um script em Python, utilizando as bibliotecas pandas, sqlalchemy, psycopg2 e openpyxl. O script:

1. Lê o arquivo data.xlsx em um DataFrame do pandas;
2. Separa a coluna de gêneros em valores individuais, respeitando o padrão "Drama, Action, Science_Fiction";
3. Extrai diretores e gêneros distintos para preencher as tabelas Directors e Genres sem duplicidades;
4. Insere os registros de filmes na tabela Movies;
5. Cria os vínculos na tabela Relations, relacionando cada filme ao seu diretor e aos seus gêneros.

Além do script de carga, também trabalhamos diretamente com comandos DML em SQL para testar o modelo e exemplificar operações básicas, como inserção, consulta e limpeza. Por exemplo, os comandos abaixo inserem manualmente o filme Avatar e seus relacionamentos:

```
INSERT INTO movies (
    original_title,
    overview,
    release_date,
    status,
    vote_average,
    vote_count,
    actors,
    original_language,
    homepage,
    title
) VALUES (
    'Avatar',
    'Um ex-fuzileiro naval é enviado ao planeta Pandora e entra em conflito
entre seguir ordens e proteger o novo mundo.',
    '2009-12-10',
    'Released',
    7.8,
```

```
    1200000,
    'Sam Worthington, Zoe Saldana, Sigourney Weaver',
    'en',
    'https://www.avatar.com',
    'Avatar'
);

INSERT INTO directors (name) VALUES ('James Cameron');
INSERT INTO genres (genre) VALUES ('Science_Fiction');
INSERT INTO genres (genre) VALUES ('Adventure');
INSERT INTO genres (genre) VALUES ('Action');

INSERT INTO relations (id_movie, id_director, id_genre)
VALUES (1, 1, 1);
```

Para análise, elaboramos diversas consultas SQL usando seleção, junção, agregação e subconsultas. Alguns exemplos:

Filmes lançados a partir de 2015:

```
SELECT
    title,
    release_date
FROM public.Movies
WHERE release_date >= '2015-01-01';
```

Filmes com seus respectivos diretores (JOIN entre tabelas):

```
SELECT
    m.title AS filme,
    d.name  AS diretor
FROM public.Movies m
JOIN public.Relations r ON m.id_movie = r.id_movie
JOIN public.Directors d ON r.id_director = d.id_director;
```

Total de filmes e média de nota por gênero (agregação + HAVING):

```
SELECT
    g.genre,
    COUNT(r.id_movie)        AS total_filmes,
    AVG(m.vote_average)      AS media_votos
FROM public.Genres g
```

```
JOIN public.Relations r ON g.id_genre = r.id_genre
JOIN public.Movies m    ON r.id_movie = m.id_movie
GROUP BY g.genre
HAVING COUNT(r.id_movie) > 5;
```

Filmes com nota acima da média geral (subconsulta):

```
SELECT
    title,
    vote_average
FROM public.Movies
WHERE vote_average > (
    SELECT AVG(vote_average) FROM public.Movies
);
```

Quantidade de lançamentos por ano:

```
SELECT
    EXTRACT(YEAR FROM release_date) AS ano,
    COUNT(id_movie)                 AS total_lancamentos
FROM public.Movies
GROUP BY EXTRACT(YEAR FROM release_date)
ORDER BY ano DESC;
```

Total de diretores distintos cadastrados:

```
SELECT
    COUNT(DISTINCT id_director) AS total_diretores_unicos
FROM public.Relations;
```

Outras queries foram utilizadas para buscas textuais, como filmes com “Love” no título ou filmes estrelados por um ator específico (actors ILIKE '%Brad Pitt%'). Essas consultas ajudaram a validar o modelo e também inspiraram alguns dos gráficos construídos no Power BI.

5 ANÁLISES E DASHBOARDS NO POWER BI

No Power BI Desktop, conectamos ao banco PostgreSQL em modo de importação, trazendo as tabelas Movies, Directors, Genres e Relations. No modelo de dados do Power BI, mantivemos os mesmos relacionamentos definidos no banco, com Relations fazendo a ponte entre filmes, gêneros e diretores.

A seguir descrevemos os principais dashboards criados.

Dashboard 1 – Diretores e produção

O primeiro painel explora a quantidade de filmes por diretor e o desempenho individual de cada um:

- Um gráfico de barras mostra a quantidade de filmes por diretor, evidenciando, por exemplo, que diretores como Steven Spielberg, Woody Allen, Clint Eastwood e Martin Scorsese estão entre os que mais aparecem no dataset.
- Um slicer (campo de pesquisa) permite filtrar e selecionar um diretor específico. Ao escolher um diretor, são exibidas informações detalhadas, como:
  - total de filmes produzidos por esse diretor (em forma de cartão);
  - melhor filme do diretor (maior vote_average) com sua respectiva nota;
  - pior filme do diretor (menor vote_average) com a respectiva nota.

Esse dashboard responde a perguntas como: “quais diretores mais aparecem no conjunto de dados?” e “qual o melhor e o pior filme de um diretor específico segundo as avaliações?”.

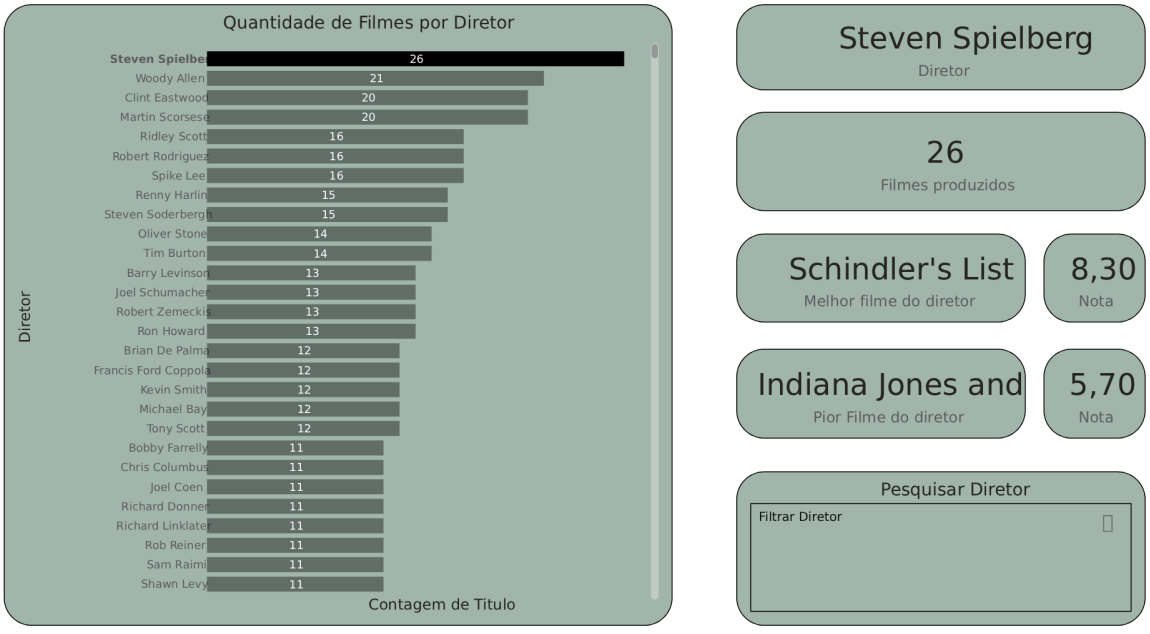

Figura 2 - Dashboard de diretores e produção

Dashboard 2 – Gêneros e avaliação

O primeiro painel é focado na distribuição de filmes por gênero e na avaliação máxima atingida em cada um:

- Um gráfico de pizza exibe a proporção de filmes por gênero. Observa-se, por exemplo, que gêneros como Drama e Comedy concentram fatias importantes do total de títulos, enquanto gêneros como Western, Foreign e TV_Movie aparecem com participação menor.
- Ao lado, um gráfico de barras horizontais apresenta a nota mais alta de cada gênero, mostrando que gêneros como Comedy, Drama e Family alcançam nota 10, enquanto outros gêneros possuem filmes com notas máximas entre 7,5 e 9,3.
- Abaixo dos gráficos, destacamos em texto o gênero com maior nota (Drama) e o gênero selecionado no momento (Action), reforçando as informações mostradas visualmente.

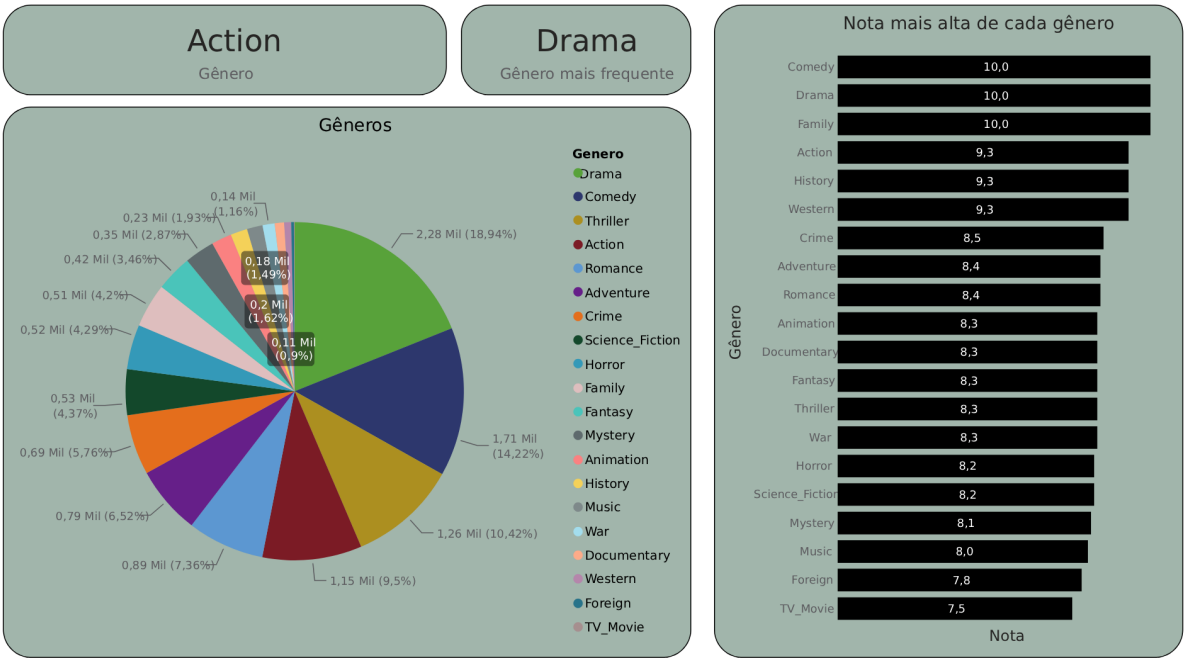

Figura 3 - Dashboard de gêneros e avaliação

Dashboard 3 – Top filmes, votos e linha do tempo

O terceiro painel combina ranking de filmes, popularidade e evolução temporal:

- Uma visualização tipo tabela ou gráfico de barras lista os 100 filmes mais bem

avaliados, ordenados pela coluna vote_average.

- Um indicador de média de votos por filme resume a popularidade geral, mostrando a média de vote_count dentro desse subconjunto.
- Um gráfico de colunas por ano apresenta o número de títulos lançados em cada ano, permitindo observar a concentração de lançamentos em determinados períodos (por exemplo, crescimento mais forte a partir da década de 1990).
- Outros campos, como a data de lançamento do filme selecionado, complementam a análise.

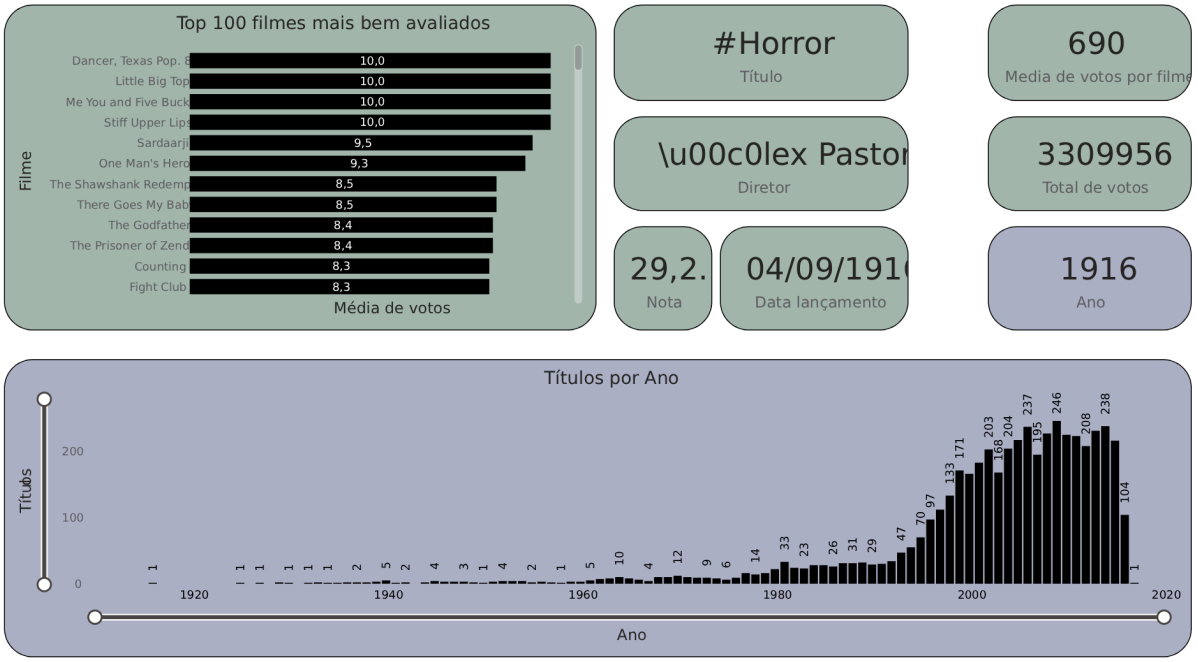

Figura 4 - Dashboard de top filmes, votos e linha do tempo

6. PRINCIPAIS INSIGHTS

A partir das consultas SQL e dos dashboards desenvolvidos, alguns insights se destacam:

- Distribuição de gêneros: o gráfico de pizza revelou que os gêneros Drama, Comedy, Thriller e Action concentram a maior parte dos filmes. Gêneros como Documentary, Western e TV_Movie aparecem em menor quantidade, indicando que o dataset é mais focado em produções voltadas ao grande público.
- Avaliação máxima por gênero: o gráfico “Nota mais alta de cada gênero” mostrou que gêneros como Comedy, Drama e Family possuem filmes com nota máxima (10), enquanto outros, como Foreign e TV_Movie, atingem notas máximas mais baixas

(por volta de 7,5–7,8). Isso sugere que, dentro do conjunto analisado, há pelo menos alguns filmes muito bem avaliados em quase todos os gêneros, mas a "nota máxima" varia.

- Diretores mais produtivos: a análise de "Quantidade de Filmes por Diretor" evidenciou que diretores como Steven Spielberg, Woody Allen, Clint Eastwood e Martin Scorsese estão entre os mais recorrentes, com dezenas de filmes cada. Ao selecionar um diretor específico, é possível ver rapidamente qual é o seu filme melhor avaliado e qual é o pior, o que ajuda a entender a variabilidade dentro da carreira de cada um.

- Top filmes e média de votos: o ranking dos filmes mais bem avaliados permite identificar títulos que combinam alta nota e boa quantidade de votos, bem como filmes menos conhecidos com nota máxima. A comparação com a média de votos reforça que nem sempre os filmes melhor avaliados são os mais populares, indicando que popularidade (número de avaliações) e qualidade percebida (nota média) não são exatamente a mesma coisa.

- Linha do tempo de lançamentos: o gráfico de títulos por ano mostrou como a produção de filmes se concentra em determinados períodos. Em geral, observa-se um aumento significativo da quantidade de lançamentos nas décadas mais recentes, o que é coerente com o crescimento da indústria cinematográfica e da digitalização dos registros.

Esses resultados ilustram como o uso combinado de SQL e Power BI ajuda a transformar uma tabela de filmes em um conjunto de insights sobre padrões de gênero, diretores, avaliações e popularidade.

7. CONCLUSÃO

O projeto atingiu o objetivo de integrar conceitos de banco de dados e análise de dados em um fluxo único: desde a modelagem e carga no PostgreSQL, passando por consultas SQL de exploração, até a construção de dashboards no Power BI.
Trabalhar com o dataset de filmes permitiu visualizar, de forma intuitiva, como decisões de modelagem (tabelas e relacionamentos) impactam diretamente as possibilidades de análise.

A partir dos resultados, foi possível identificar gêneros mais frequentes e melhor

avaliados, diretores com maior número de filmes, diferenças entre filmes mais bem avaliados e mais votados e tendências temporais de lançamentos.
Em trabalhos futuros, seria interessante enriquecer o modelo com variáveis financeiras (orçamento e receita) ou dados de outras fontes, ampliando a análise para a relação entre crítica, público e desempenho comercial.

Ainda assim, mesmo com o escopo limitado a notas, votos e gêneros, o estudo demonstrou, na prática, o papel de SQL e Power BI como ferramentas complementares na construção de análises baseadas em dados.