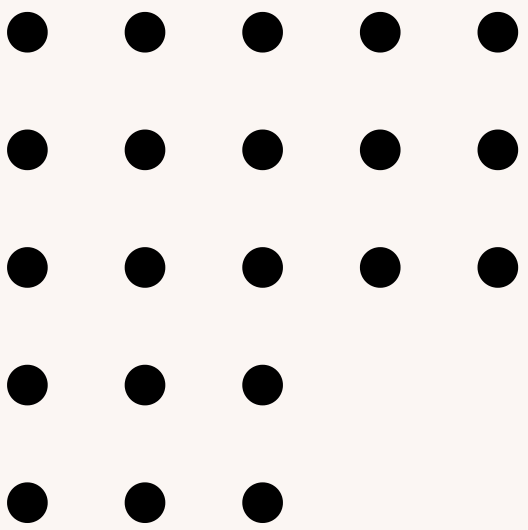


ALV A1 : FASE 1

- CARLOS DANIEL
- ESDRAS CAVALCANTI
- KAUAN KEVEM

- MARCELO ANGELO
- WALLÉRIA SIMÕES
- XIMENA BEATRIZ



MINIMUNDO: ALV

- ALEXIS, LAURA E VALÉRIA
- PLATAFORMA DE STREAMING
- ROYALTIES ANUAIS
- MODELO DE ASSINATURAS
- SISTEMAS DE APOIO À DECISÃO

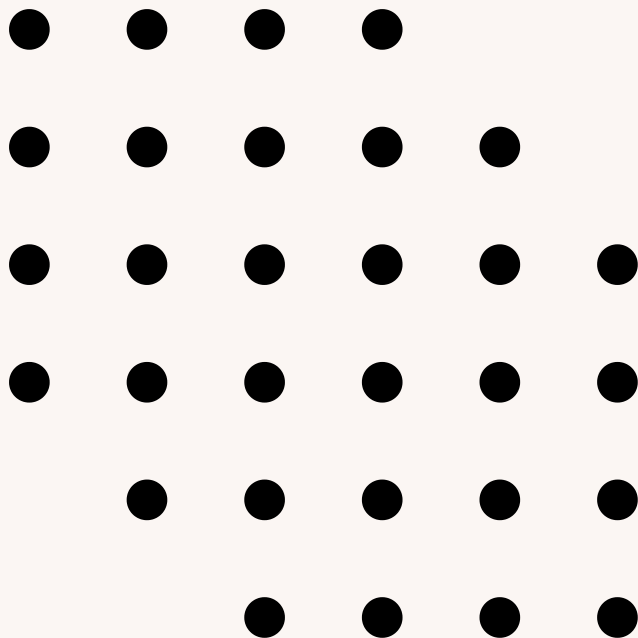


OBJETIVOS INFORMATICAIS

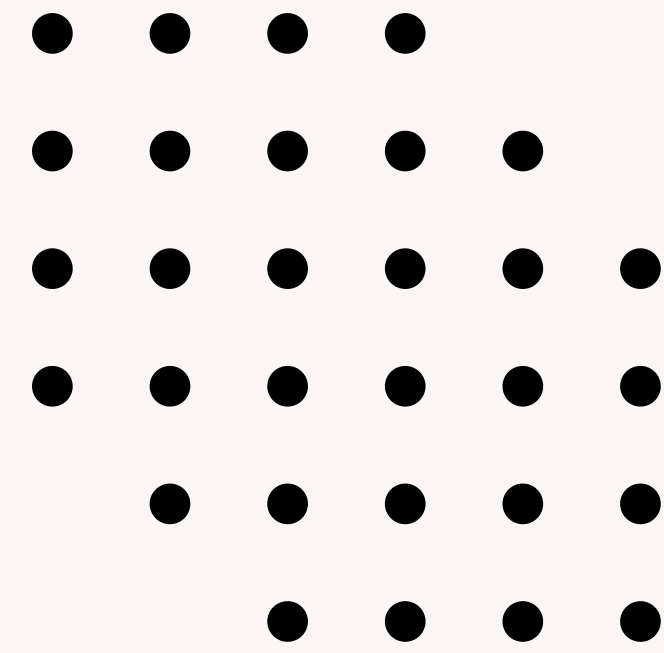
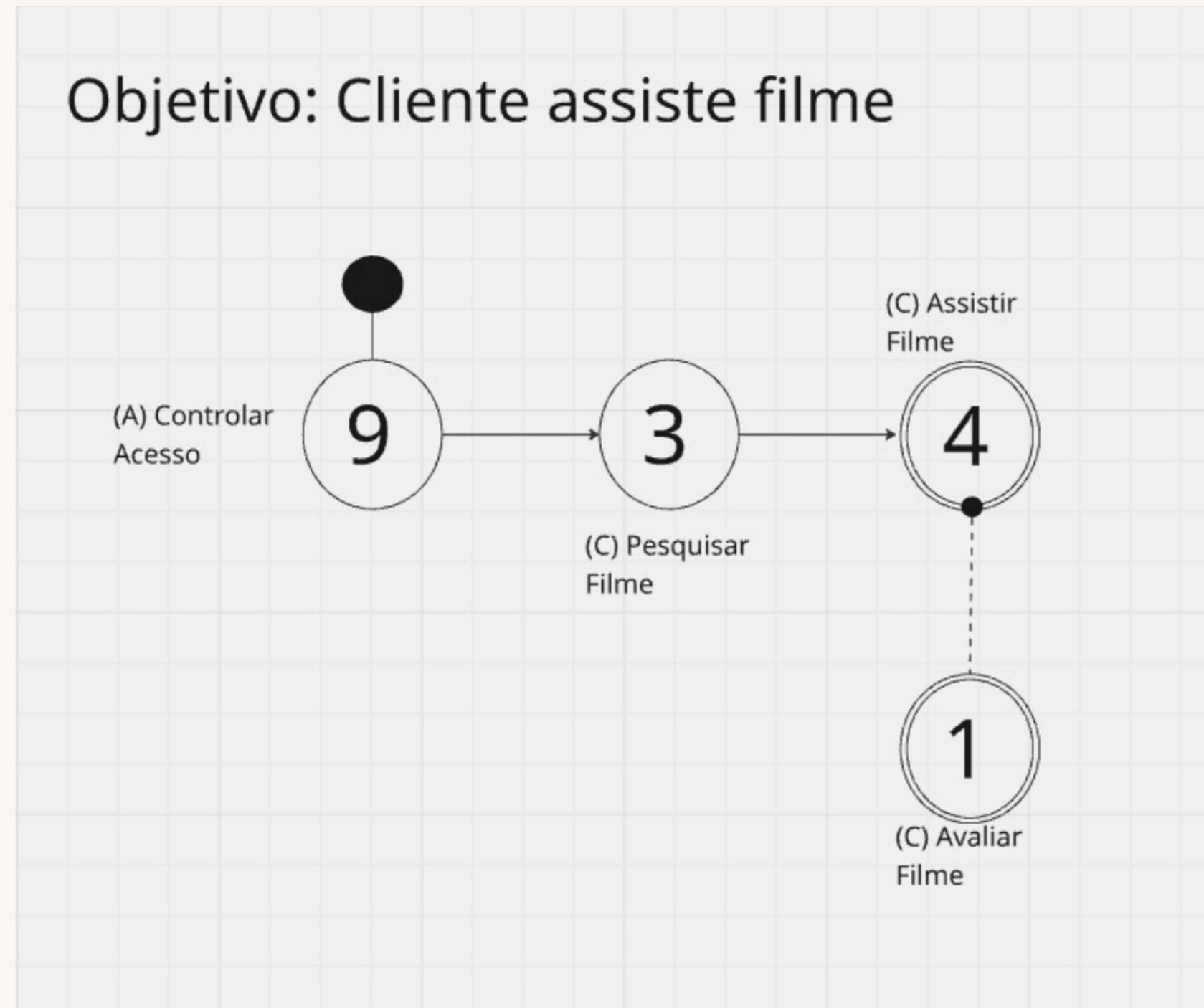
| Cliente | Gerente de Conteúdo | Administração |
|-------------------------|---------------------|-----------------------|
| 1 - Avaliar filme | 8 - Adicionar filme | 10 - Controlar acesso |
| 2 - Efetuar pagamento | 9 - Remover filme | 11 - Pagar royalties |
| 3 - Pesquisar filme | | |
| 4 - Assistir filme | | |
| 5 - Cancelar assinatura | | |
| 6 - Atualizar cadastro | | |
| 7 - Realizar assinatura | | |

INTERFACES INFORMATICAIS

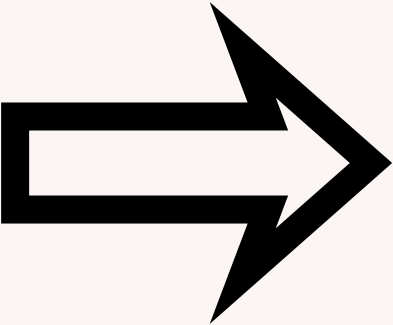
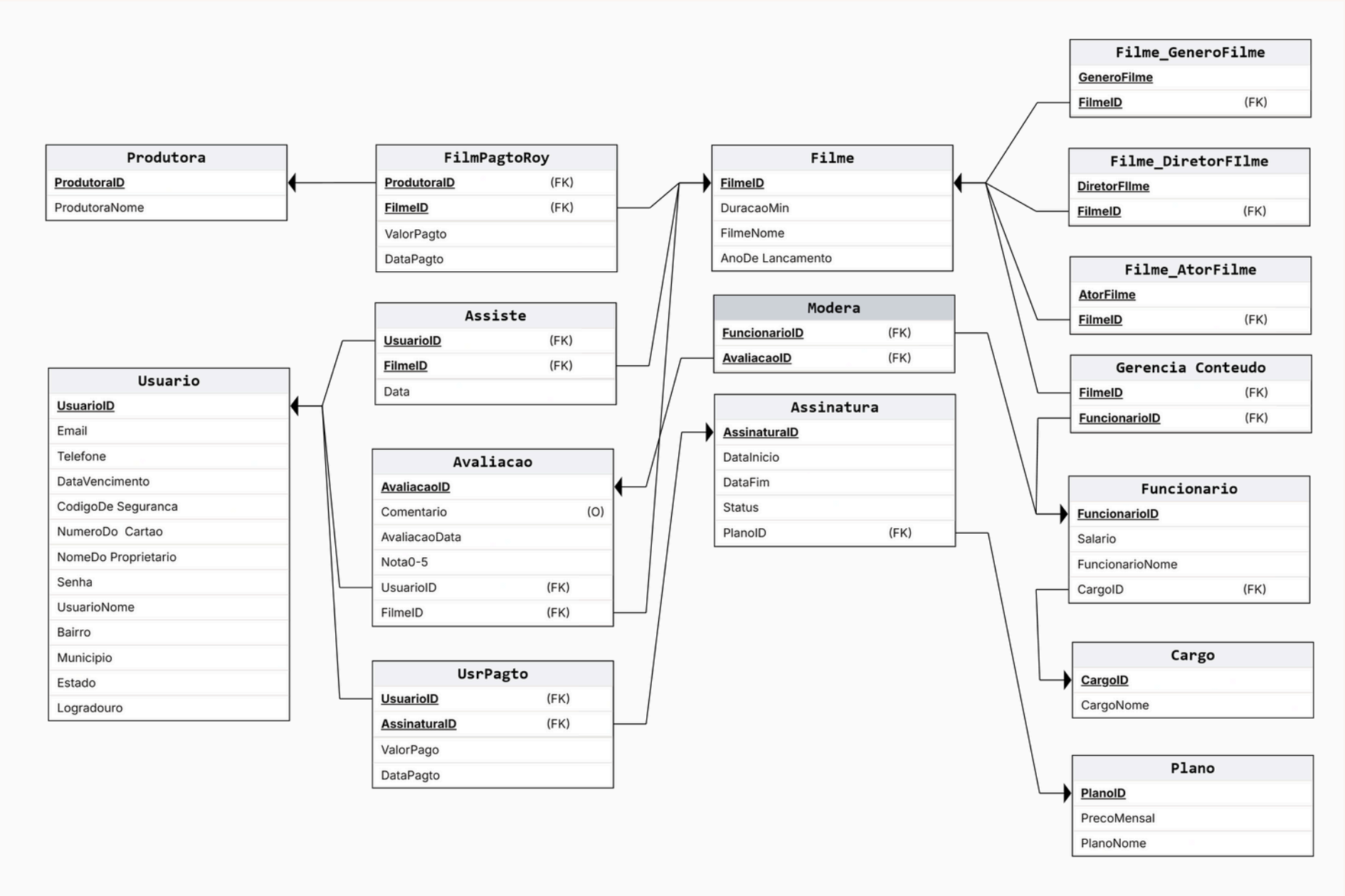
| Ator: Cliente | Objetivo 1: Avaliar filme |
|--|---------------------------|
| → <u>avaliacao</u> = id_cliente + id_filme + <u>nota_cliente</u> + comentario + <u>dt_hr_avaliacao</u> | |
| <u>Descrição:</u> Cliente envia uma avaliação de um filme | |
| <u>Propósito:</u> Captar satisfação e engajamento do cliente | |
| <u>Frequência:</u> Milhares/dia | |



SEQUÊNCIA ADMISSÍVEL



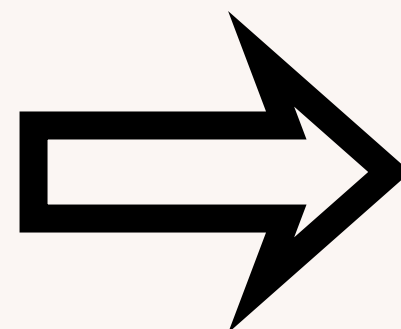
DDL



```
1 Drop schema if exists oper_alv cascade;
2 create schema oper_alv;
3
4 set search_path=oper_alv;
5
6 CREATE TABLE Plano
7 (
8     PlanoID INT NOT NULL,
9     PrecoMensal FLOAT NOT NULL,
10    PlanoNome VARCHAR(128) NOT NULL,
11    PRIMARY KEY (PlanoID)
12 );
13
14 CREATE TABLE Assinatura
15 (
16     AssinaturaID INT NOT NULL,
17     DataInicio DATE NOT NULL,
18     DataFim DATE NOT NULL,
19     Status INT NOT NULL,
20     PlanoID INT NOT NULL,
21     PRIMARY KEY (AssinaturaID),
22     FOREIGN KEY (PlanoID) REFERENCES Plano(PlanoID)
23 );
24
25 CREATE TABLE Usuario
26 (
27     UsuarioID INT NOT NULL,
28     Email VARCHAR(128) NOT NULL,
29     Telefone VARCHAR(128) NOT NULL,
30     DataVencimento DATE NOT NULL,
31     CodigoDe_Seguranca INT NOT NULL,
32     NumeroDo_Cartao VARCHAR(128) NOT NULL,
33     NomeDo_Proprietario VARCHAR(128) NOT NULL,
34     Senha INT NOT NULL,
35     UsuarioNome VARCHAR(128) NOT NULL,
36     Bairro VARCHAR(128) NOT NULL,
37     Municipio VARCHAR(128) NOT NULL,
38     Estado VARCHAR(128) NOT NULL,
39     Logradouro VARCHAR(128) NOT NULL,
40     PRIMARY KEY (UsuarioID)
41 );
42
43 CREATE TABLE Filme
```

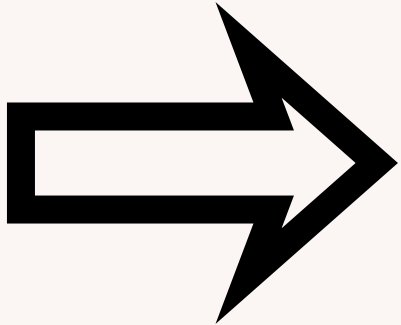
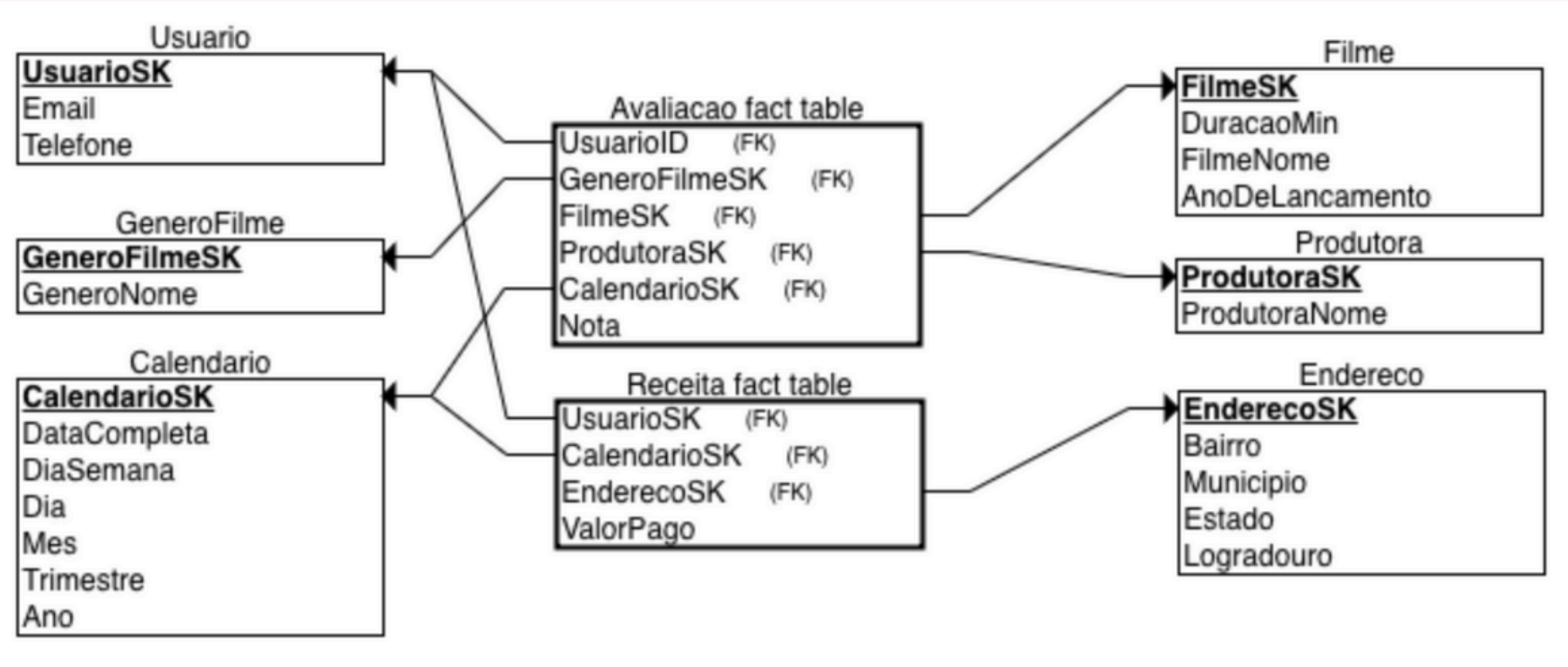
DML

```
332     def create_tables(self):
333         self.create_table_plano()
334         self.create_table_usuario(2000)
335         self.create_table_filme(500)
336         self.create_table_produtores(20)
337         self.create_table_cargo(15)
338         self.create_table_funcionario(1000)
339
340         self.create_table_assinatura(1500)
341         self.create_table_avaliacao(20000)
342         self.create_table_film_pagto_roy(400)
343         self.create_table_usr_pagto(6000)
344         self.create_table_assiste(15000)
345         self.create_table_moderador(20)
346         self.create_table_gerencia_conteudo(50)
347
348         self.create_table_filme_genero_filme(100)
349         self.create_table_filme_diretor_filme(100)
350         self.create_table_filme_ator_filme(100)
351
352 >     def write_dml(self, path: str):...
360
361
362     if __name__ == "__main__":
363         db = DataBase()
364         db.create_tables()
365         db.write_dml("DML_populate_tables_huge_AVL.sql")
```



+60K
LINHAS DE DADOS

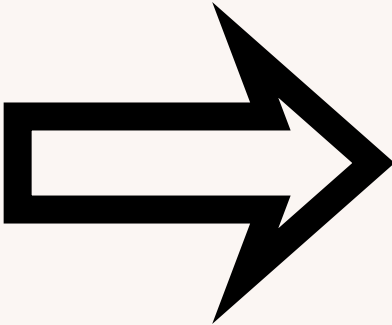
DIMENSIONAL + DW



```
28 CREATE TABLE GeneroFilme
29 (
30     GeneroFilmeSK INT NOT NULL,
31     GeneroNome CHAR NOT NULL,
32     PRIMARY KEY (GeneroFilmeSK)
33 );
34
35 CREATE TABLE Calendario
36 (
37     CalendarioSK INT NOT NULL,
38     DataCompleta DATE NOT NULL,
39     DiaSemana CHAR NOT NULL,
40     Dia INT NOT NULL,
41     Mes INT NOT NULL,
42     Trimestre INT NOT NULL,
43     Ano INT NOT NULL,
44     PRIMARY KEY (CalendarioSK)
45 );
46
47 CREATE TABLE Produtora
48 (
49     ProdutoraSK INT NOT NULL,
50     ProdutoraNome CHAR NOT NULL,
51     PRIMARY KEY (ProdutoraSK)
52 );
53
54 CREATE TABLE FactAvaliacao
55 (
56     Nota INT NOT NULL,
57     UsuarioID INT NOT NULL,
58     GeneroFilmeSK INT NOT NULL,
59     FilmeSK INT NOT NULL,
60     ProdutoraSK INT NOT NULL,
61     CalendarioSK INT NOT NULL,
62     FOREIGN KEY (UsuarioID) REFERENCES Usuario(UsuarioSK),
63     FOREIGN KEY (GeneroFilmeSK) REFERENCES GeneroFilme(GeneroFilmeSK),
64     FOREIGN KEY (FilmeSK) REFERENCES Filme(FilmeSK),
65     FOREIGN KEY (ProdutoraSK) REFERENCES Produtora(ProdutoraSK),
66     FOREIGN KEY (CalendarioSK) REFERENCES Calendario(CalendarioSK)
67 );
68
69 CREATE TABLE Receita
70 (
```


ETL

```
1 ALTER TABLE dw_alv.usuario ALTER COLUMN email TYPE varchar(128);
2 ALTER TABLE dw_alv.usuario ALTER COLUMN telefone TYPE varchar(64);
3 ALTER TABLE dw_alv.filme ALTER COLUMN filnome TYPE varchar(128);
4 ALTER TABLE dw_alv.endereco ALTER COLUMN bairro TYPE varchar(128);
5 ALTER TABLE dw_alv.endereco ALTER COLUMN municipio TYPE varchar(128);
6 ALTER TABLE dw_alv.endereco ALTER COLUMN estado TYPE varchar(64);
7 ALTER TABLE dw_alv.endereco ALTER COLUMN logradouro TYPE varchar(128);
8 ALTER TABLE dw_alv.generofilme ALTER COLUMN generonome TYPE varchar(128);
9 ALTER TABLE dw_alv.calendario ALTER COLUMN diasemana TYPE varchar(9);
10 ALTER TABLE dw_alv.produtora ALTER COLUMN produtoranome TYPE varchar(128);
11
12 WITH src AS (SELECT DISTINCT email, telefone FROM oper_alv.usuario),
13 m AS (SELECT ROW_NUMBER() OVER (ORDER BY email, telefone) AS usuariosk, email, telefone FROM src)
14 INSERT INTO dw_alv.usuario(usuariosk, email, telefone)
15 SELECT usuariosk, email, telefone FROM m;
16
17 DROP TABLE IF EXISTS temp_map_usuario;
18 CREATE TEMP TABLE temp_map_usuario AS
19 SELECT u.usuarioid, d.usuariosk
20 FROM oper_alv.usuario u
21 JOIN dw_alv.usuario d ON d.email=u.email AND d.telefone=u.telefone;
22
23 WITH src AS (SELECT DISTINCT bairro, municipio, estado, logradouro FROM oper_alv.usuario),
24 m AS (SELECT ROW_NUMBER() OVER (ORDER BY bairro,municipio,estado,logradouro) AS enderecosk, bairro,municipio,estado,logradouro FROM src)
25 INSERT INTO dw_alv.endereco(enderecosk,bairro,municipio,estado,logradouro)
26 SELECT enderecosk,bairro,municipio,estado,logradouro FROM m;
27
28 DROP TABLE IF EXISTS temp_map_endereco;
29 CREATE TEMP TABLE temp_map_endereco AS
30 SELECT enderecosk,bairro,municipio,estado,logradouro FROM dw_alv.endereco;
31
32 DROP TABLE IF EXISTS temp_map_filme;
33 CREATE TEMP TABLE temp_map_filme AS
34 SELECT filmeid, ROW_NUMBER() OVER (ORDER BY filmeid) AS filmesk
35 FROM oper_alv.filme;
36
37 INSERT INTO dw_alv.filme(filmesk,duracaomin,filnome,anodelancamento)
38 SELECT m.filmesk, f.duracaomin, f.filnome, f.anode_lancamento
39 FROM oper_alv.filme f
40 JOIN temp_map_filme m USING (filmeid);
41
42 WITH src AS (SELECT DISTINCT generofilme FROM oper_alv.filme_generofilme),
43 m AS (SELECT ROW_NUMBER() OVER (ORDER BY generofilme) AS generofilmesk, generofilme AS generonome FROM src)
44 INSERT INTO dw_alv.generofilme(generofilmesk,generonome)
45 SELECT generofilmesk, generonome FROM m;
46
47 INSERT INTO dw_alv.generofilme(generofilmesk,generonome)
```



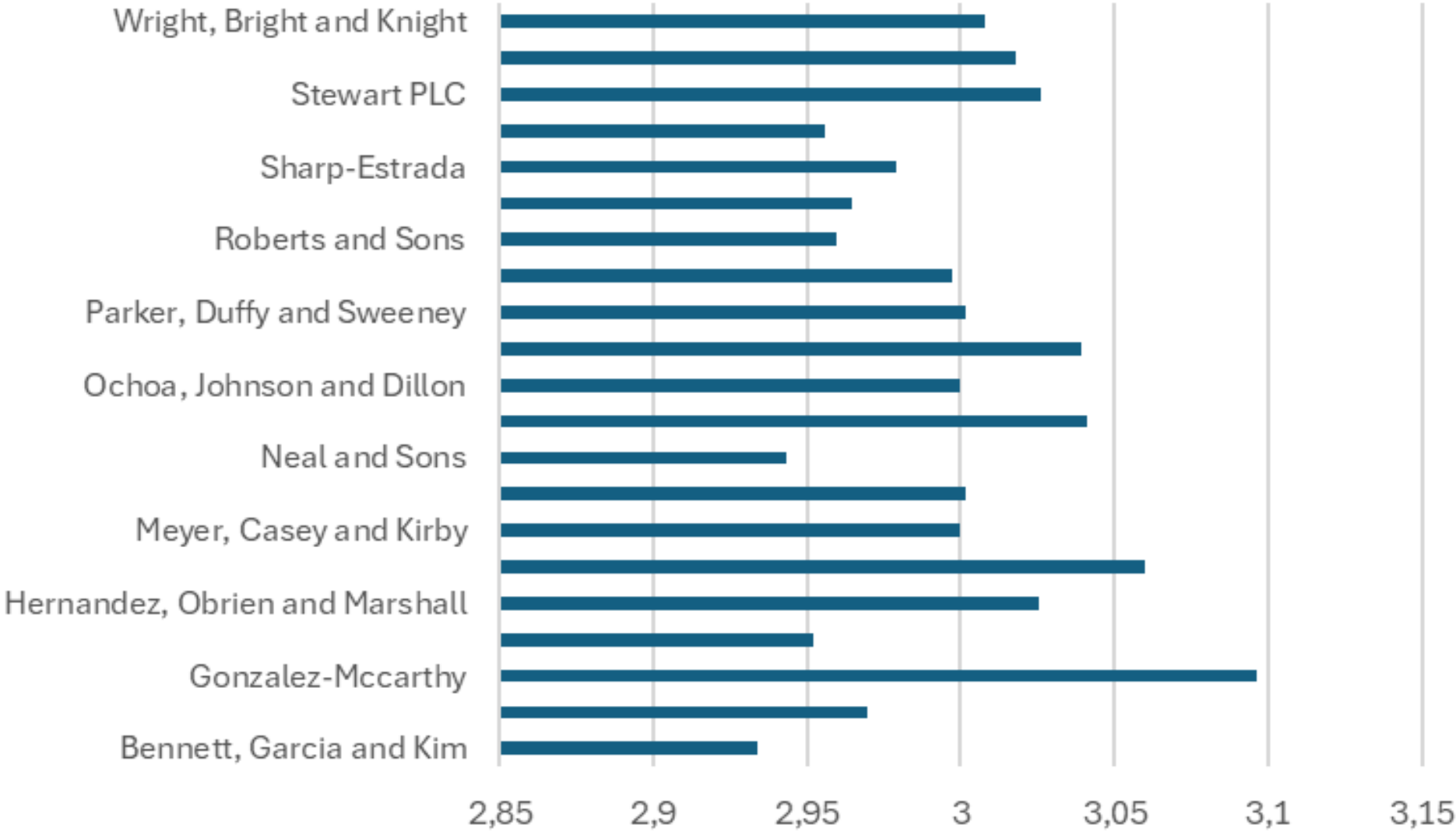
| nota | usuarioid | generofilmesk | filmesk | produtorask | calendariosk |
|------|-----------|---------------|---------|-------------|--------------|
| 2 | 435 | 97 | 250 | 15 | 853 |
| 2 | 40 | 97 | 376 | 21 | 853 |
| 2 | 488 | 97 | 139 | 2 | 853 |
| 1 | 1137 | 97 | 34 | 21 | 853 |
| 3 | 1425 | 76 | 121 | 21 | 853 |
| 4 | 49 | 97 | 98 | 1 | 853 |
| 4 | 1694 | 97 | 93 | 21 | 853 |
| 5 | 289 | 89 | 60 | 14 | 853 |
| 1 | 738 | 97 | 189 | 11 | 853 |
| 1 | 792 | 97 | 161 | 6 | 853 |
| 3 | 1026 | 97 | 161 | 6 | 853 |
| 2 | 502 | 97 | 294 | 5 | 853 |
| 3 | 1739 | 97 | 472 | 21 | 853 |
| 2 | 126 | 97 | 317 | 19 | 853 |
| 2 | 182 | 97 | 500 | 21 | 853 |
| 2 | 62 | 97 | 424 | 21 | 853 |
| 4 | 1729 | 22 | 142 | 21 | 853 |
| 1 | 953 | 97 | 67 | 3 | 853 |
| 5 | 1770 | 97 | 110 | 9 | 853 |
| 4 | 349 | 97 | 378 | 17 | 853 |
| 5 | 690 | 97 | 231 | 3 | 853 |
| 2 | 1631 | 97 | 288 | 21 | 853 |
| 2 | 213 | 5 | 478 | 21 | 853 |
| 4 | 784 | 97 | 299 | 21 | 853 |
| 3 | 36 | 97 | 10 | 2 | 705 |
| 2 | 510 | 97 | 100 | 21 | 705 |

DASHBOARD

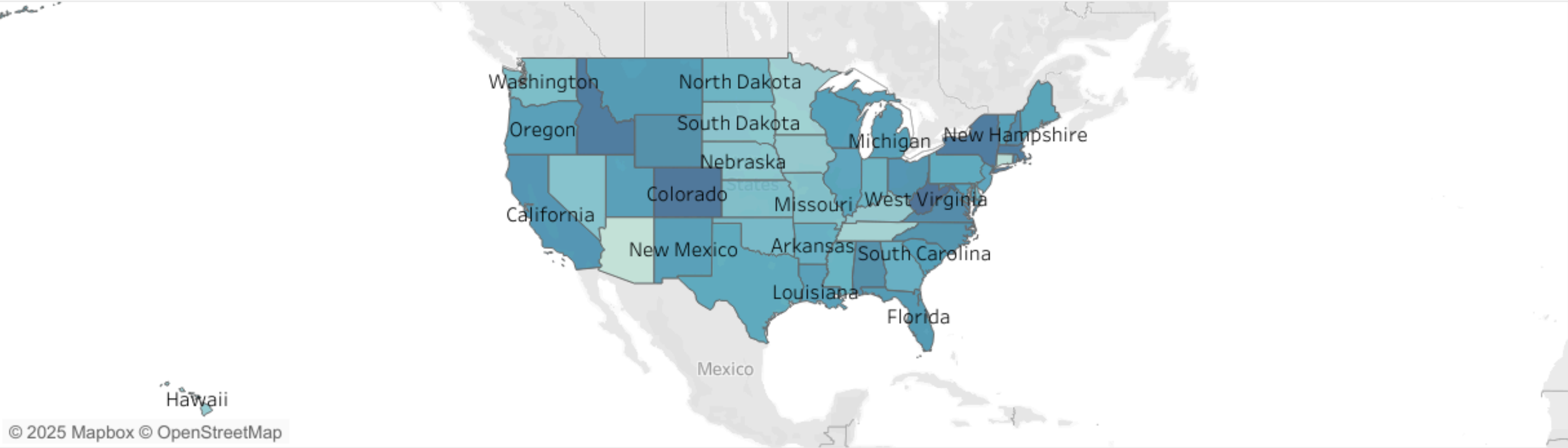
Média de nota

Nota média dos filmes de cada produtora

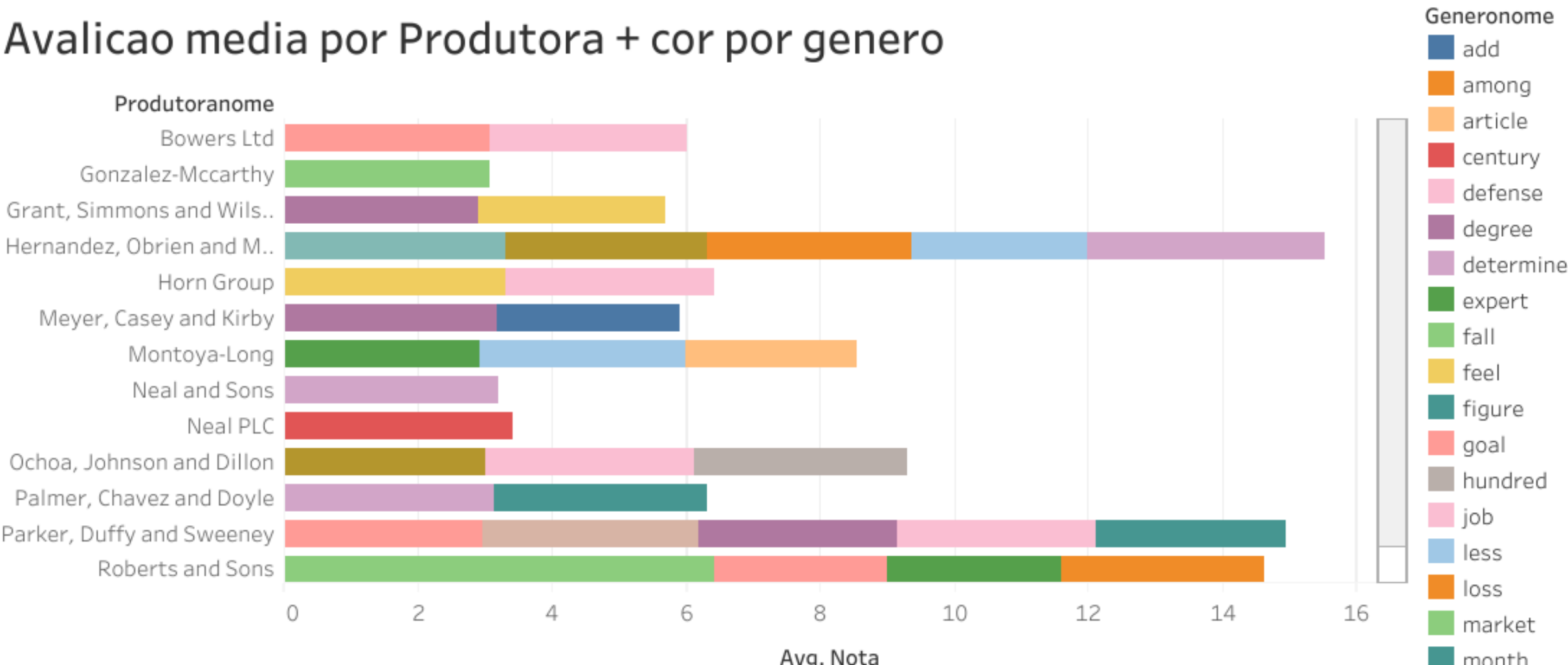
produtoranome ▼

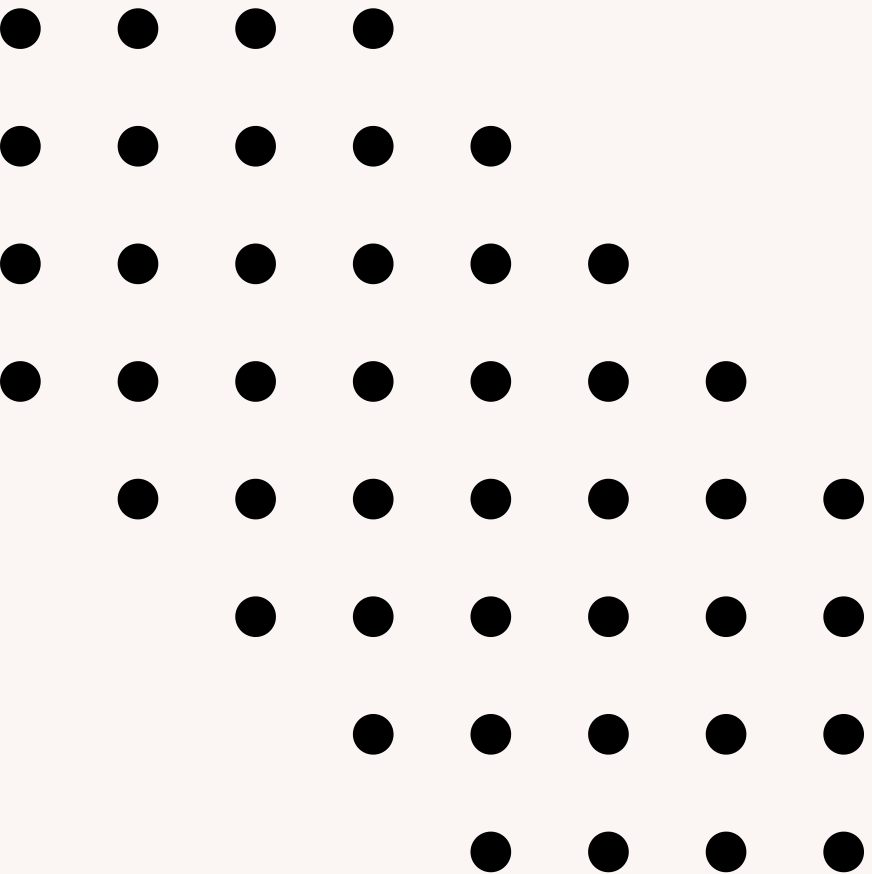


Total recaudado por Estado



Avalicao media por Produtora + cor por genero





Obrigado!

