

**CENTRO UNIVERSITÁRIO FEI**  
**CIÊNCIA DA COMPUTAÇÃO**

**RELATÓRIO 2º ETAPA DE BANCO DE DADOS**

**ANDY SILVA BARBOSA – RA: 22.218.025-9**

**RAFAEL ZACARIAS PALIERINI - RA: 22.218.030-9**

**RUBENS DE ARAÚJO RODRIGUES MENDES - RA: 22.218.009-3**

**VITOR ACOSTA DA ROSA – RA: 22.218.006-9**

**SÃO BERNARDO – SP**

**2020**

## SUMÁRIO

|  |   |
|--|---|
| 1. ALTERAÇÕES DO DIAGRAMA ER.....                                | 2 |
| 1.1 Entidade “Perfil” .....                                      | 2 |
| 1.2 Entidade “Produção Audiovisual” .....                        | 3 |
| 1.3 Entidade “Produtora” .....                                   | 4 |
| 1.4 Entidade “Diretor” .....                                     | 6 |
| 2. MAPEAMENTO DO MER PARA O MODELO RELACIONAL.....               | 6 |
| 2.1 Tratamento do Overlap na Entidade Produção Audiovisual ..... | 7 |
| 2.2 Relações 1 para Muitos 1..N .....                            | 7 |
| 2.3 Relações Muitos para Muitos M..N .....                       | 7 |
| 3. CONSULTA.....   | 8 |

## 1. ALTERAÇÕES DO DIAGRAMA ER

Para a correta criação do modelo relacional utilizando a ferramenta **StarUML**, foi necessário alterar alguns atributos, entidades e relacionamentos previamente idealizados na modelagem conceitual, visando mapear, de forma mais concisa e objetiva, toda a estrutura desenvolvida.

### 1.1 Entidade “Perfil”

A entidade perfil sofreu alterações em alguns atributos:

- Foi adicionado um ID único para todo usuário, a fim de manter as tuplas do banco de dados normalizadas.
- Além disso, o atributo "foto" foi alterado para "url\_foto", isso porque é inviável armazenar a imagem de cada perfil do serviço no banco de dados. Nesse contexto será armazenado somente um varchar, a url, a localização e o caminho do servidor até a imagem.
- A grande mudança foi referente ao atributo composto "histórico", a sua idealização, na primeira etapa, consistia em um atributo composto por três atributos multivalorados: a produção audiovisual assistida, o último acesso à essa produção e a quantidade de vezes assistidas; Algo que, além de não normalizado, mostrava-se complexo para armazenar no banco. Nessa perspectiva, o histórico será dado através da **própria relação** entre a entidade perfil e produção audiovisual, haja vista que, essa relação gerará um atributo, multivalorado, "último acesso". Através dele é possível visualizar a última vez que o usuário acessou uma certa produção audiovisual, bem como os dados dessa produção e, por ser um atributo que irá armazenar todos os acessos do usuário àquela produção, é possível contar a quantidade de entradas a um determinado filme/série/documentário.

Na figura abaixo, podemos ver as mudanças descritas anteriormente.

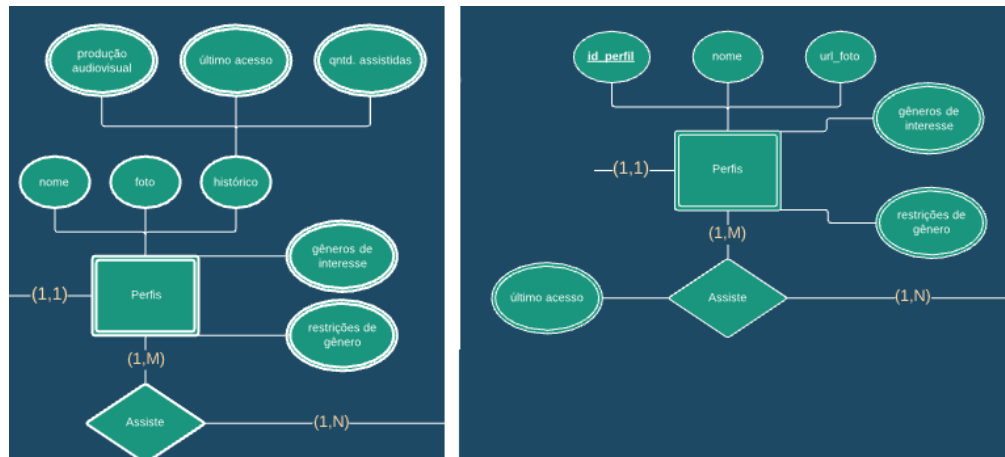


Figura 1 À esquerda a entidade idealizada na etapa 1. À direita, a mesma entidade após alterações na etapa 2

## 1.2 Entidade “Produção Audiovisual”

A entidade da produção audiovisual, a entidade central construída pelo grupo sofreu alterações em seus atributos e em algumas relações:

- O atributo código que era uma chave primária única para cada produção teve o nome alterado para ID\_Produção.
- O atributo "franquia" foi alterado para avaliação no imdb. Na primeira etapa o atributo franquias teria como objetivo mostrar o "universo" do filme. Entretanto, como a entidade passa pelo processo de "overlapping", esse atributo teria diversos valores nulos (caso o tipo da obra fosse um documentário ou até mesmo uma série). No lugar, foi adicionado a avaliação no imdb do filme, que tem a função de retornar um valor: a nota do filme.
- Além disso, a relação entre produção audiovisual e produtora, "Produz", na primeira etapa tinha um atributo que seria gerado, a "quantidade de produções audiovisuais", entretanto, esse atributo mostrou-se descartável já que, para contar quantas obras foram criadas por uma determinada produtora basta contar quantas vezes uma determinada produtora aparece nos registros:

```
SELECT produtora.nome,
       COUNT(produtora.nome) as producoes_realizadas
FROM produtora
INNER JOIN prod_audio_produtores
ON produtora.cnpj = prod_audio_produtores.cnpj
GROUP BY (produtora.nome)
ORDER BY producoes_realizadas DESC
```

- As mudanças podem ser vistas nas figuras abaixo:

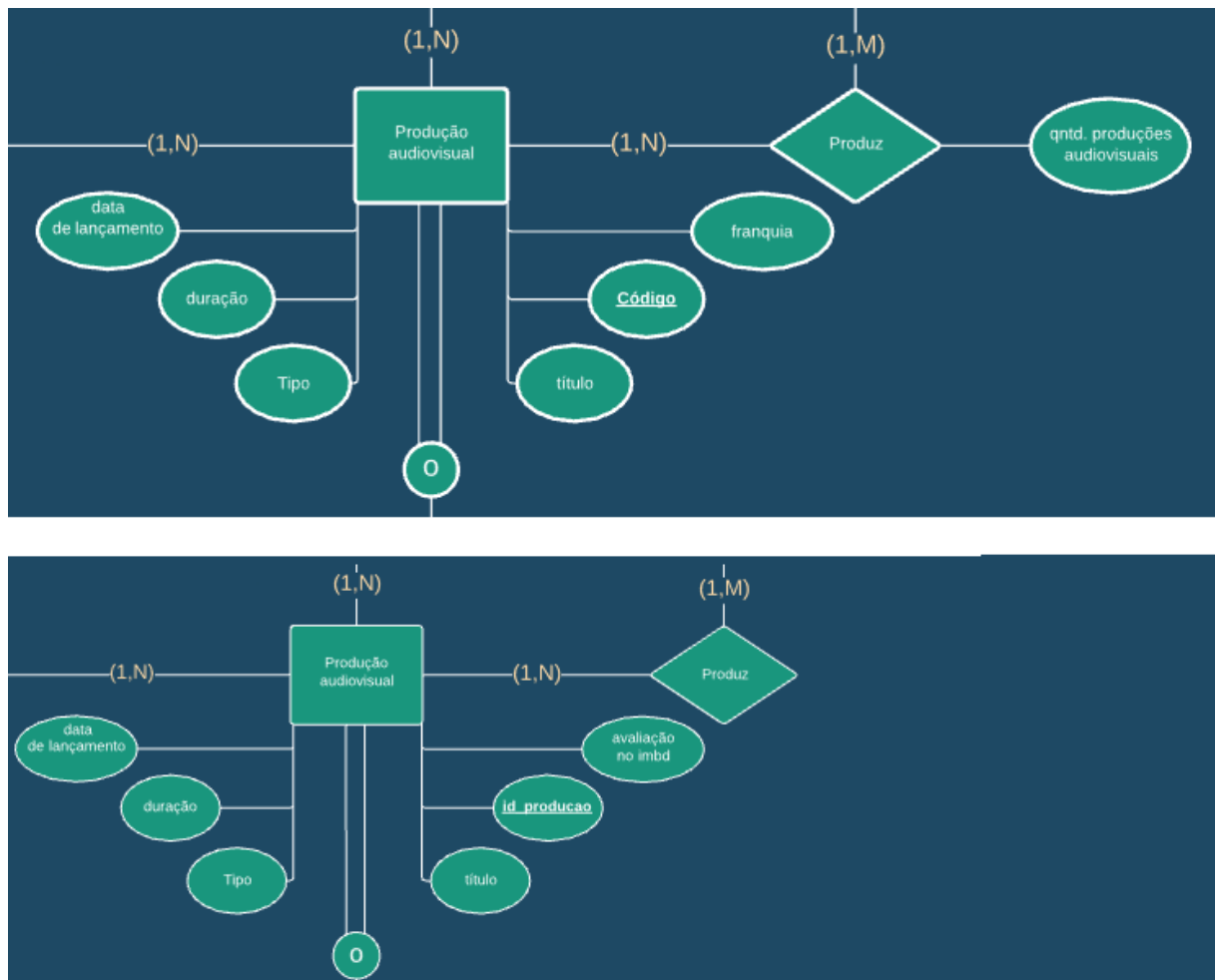


Figura 2 A imagem acima é referente à etapa 1, abaixo, a mesma entidade porém com as alterações realizadas na segunda etapa.

### 1.3 Entidade “Produtora”

A entidade referente aos dados da produtora de uma determinada obra teve mudanças principalmente em seus atributos.

- O atributo endereço, idealizado na primeira etapa, foi substituído por atributos atômicos, a saber: país, estado, cidade, CEP, rua e número. Alteração feita visando normalizar o banco.
- As mudanças são visíveis na figura 3.

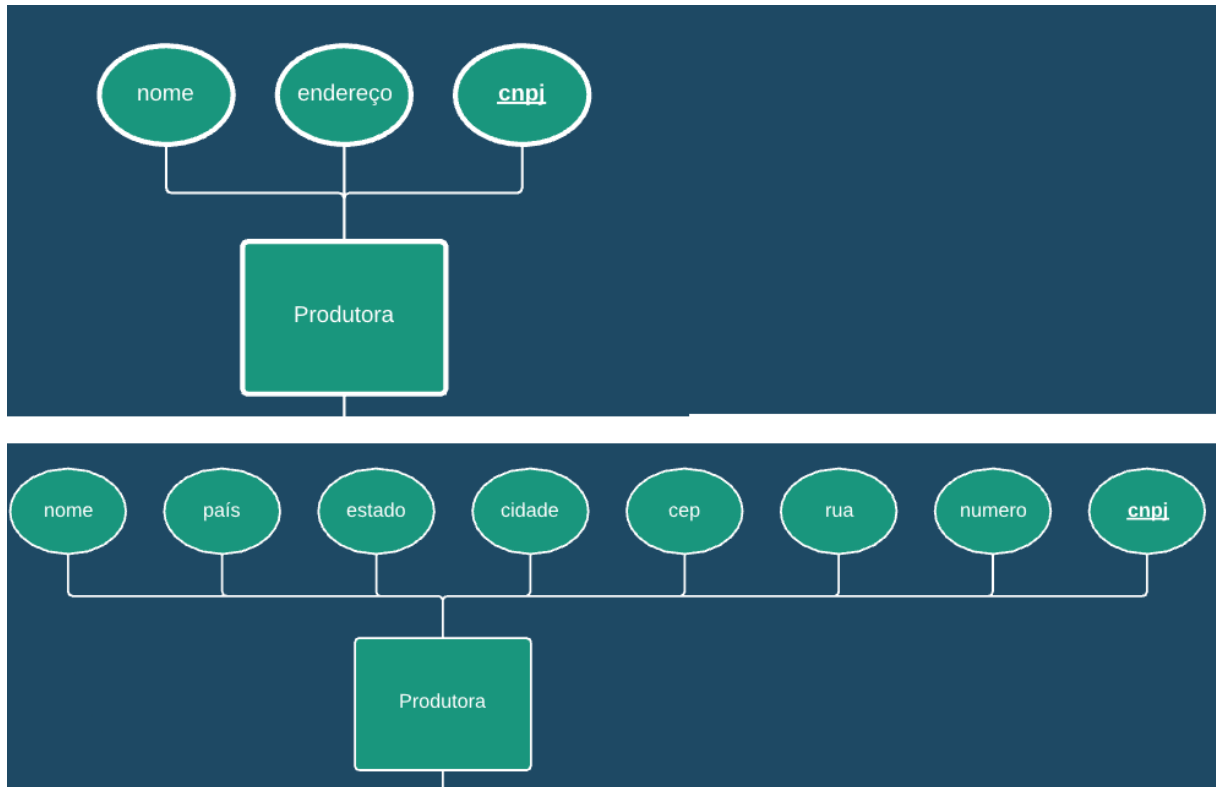


Figura 3 A imagem acima é referente à Produtora idealizada na etapa 1, já a imagem abaixo é referente à entidade envolvendo somente atributos atômicos (etapa 2).

## 1.4 Entidade “Diretor”

A entidade diretor, sofreu alterações somente em seus atributos.

- O atributo "avaliações" que era um atributo composto formado por outros dois atributos multivalorados, mostrou-se inviável em sua implementação, pois geraria diversos dados em uma única coluna, sobrecarregando as tuplas dessa entidade. Nessa perspectiva, foi removido tal atributo e inserido o atributo "melhor obra". Perceba que, como a melhor obra do diretor estará marcada, é possível pela entidade de produção audiovisual, acessar sua classificação (através do atributo já citado "avaliação do imbd"). Além disso, caso o usuário deseje ver todas as obras de um determinado diretor, é possível observar a avaliação de todas as obras.
- Essa alteração pode ser vista abaixo:

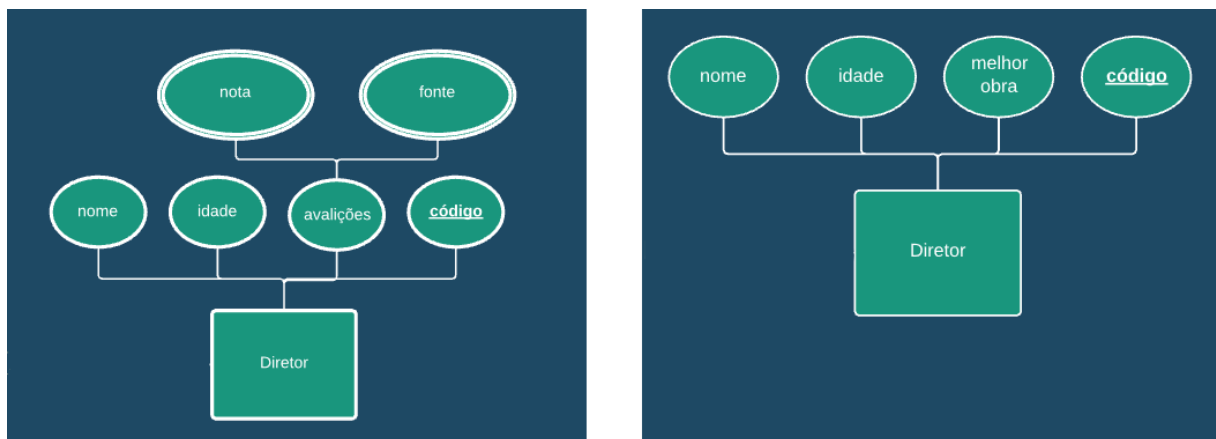


Figura 4 A imagem à esquerda faz jus a entidade idealizada na etapa 1, já a imagem à direita mostra as alterações realizadas na etapa 2 do projeto.

## 2. MAPEAMENTO DO MER PARA O MODELO RELACIONAL

Após todas as alterações na base do modelo de entidade de relacionamento (MER), foi necessário mapear as relações e interligar os dados de diferentes tabelas. Após o mapeamento houve como resultado a criação de 15 tabelas, um número esperado, já que 83.3% (5) das relações eram de muitos para muitos (M..N), e também, nosso MER possuía um Overlap.

## 2.1 Tratamento do Overlap na Entidade Produção Audiovisual

Cada entidade que passava pelo processo de "overlapping" virou uma tabela, contendo os atributos originais idealizados na etapa 1 mais uma chave estrangeira com o código da produção responsável por ligar as tabelas. Nesse contexto, as entidades filme, série e documentário foram transformadas em tabelas.

## 2.2 Relações 1 para Muitos 1..N

A única relação de todo o banco idealizado que possuía o tipo 1..N era a relação "Possui", que ligava as entidades "Usuário" e "Perfil".

Para mapear essa relação, bastou incluir como chave estrangeira na entidade participante (no caso a entidade "Perfil") a chave primária da entidade "Usuário". Dessa forma, a entidade "Perfil", além de todos seus atributos, recebeu uma chave estrangeira "id\_usuario", para referenciar sua relação com a entidade "Usuário".

| Usuario |            |              | Perfil |                  |                |
|---------|------------|--------------|--------|------------------|----------------|
| PK      | id_usuario | INTEGER(100) | PK     | id_perfil        | INTEGER(100)   |
|         | email      | VARCHAR(50)  | FK     | id_usuario       | INTEGER(100)   |
|         | senha      | VARCHAR(25)  |        | nome             | VARCHAR(50)    |
|         | plano      | VARCHAR(50)  |        | url_foto         | VARCHAR(50)    |
|         |            |              |        | restricao_genero | VARCHAR(10000) |
|         |            |              |        | genero_interesse | VARCHAR(10000) |

Figura 5 Atribuição de chave estrangeira para a entidade "Perfil".

## 2.3 Relações Muitos para Muitos M..N

Correspondente à maior parte das relações de todo o banco, tem-se as relações que ligam as entidades:

- Perfil e Produção Audiovisual
- Produção Audiovisual e Produtora
- Produção Audiovisual e Ator
- Produção Audiovisual e Diretor
- Produção Audiovisual e Roteirista



Todas elas obedeceram o seguinte: Para cada relação M.. N, foi criada uma nova tabela com os atributos sendo as chaves primárias de cada entidade participante desse relacionamento, bem como, se existir, os atributos gerados por aquela relação.

Logo, todas relações citadas acima, foram transformadas em tabelas.

Portanto, após transformar as relações em novas tabelas, o modelo relacional feito através da ferramenta StarUml resultou em:

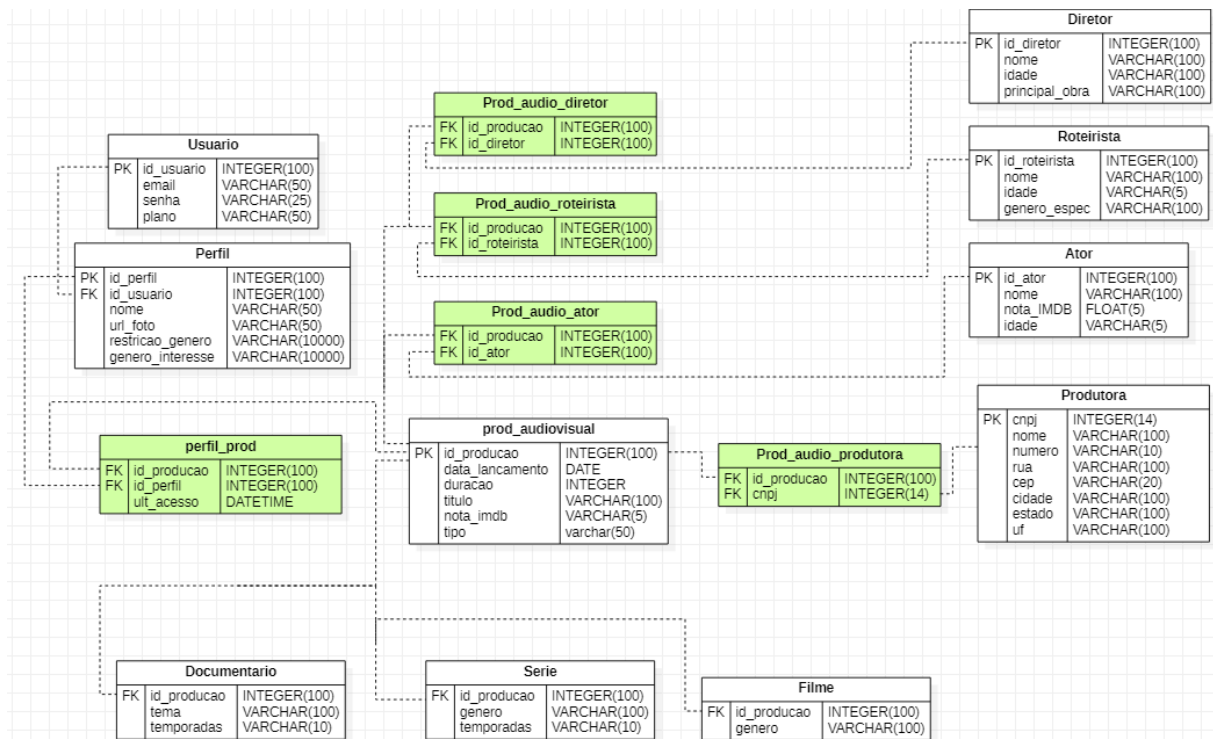


Figura 6 Modelo relacional após mapear todo o MER.

Perceba que, as linhas pontilhadas evidenciam como as entidades se comunicam entre si, além disso, em verde é possível visualizar todas as relações N..M que se tornaram novas tabelas e garantiram a interligação entre todos os dados.

### 3. CONSULTA

A query pedida pela professora foi: "Quais foram os top 5 itens mais vistos no ano de 2020. Indicar o título, tipo da obra e avaliação média no imdb dos atores das respectivas obras." O grupo decidiu separar em partes toda essa query, uma vez que são utilizadas quatro tabelas para retornar o resultado esperado.

Essa consulta ocasionou em diversos problemas, sendo o principal deles a multiplicação das funções de agregação (no caso, o COUNT do top 5) ao realizar o JOIN com as demais tabelas, algo que já era esperado, uma vez que, ao realizar o JOIN é feito o produto cartesiano entre as tuplas.

Para contornar esse problema, o grupo decidiu fazer a query proposta partindo das notas médias do IMDB dos atores (pois envolveria somente o AVG, uma função de agregação não afetada pelos JOINS) e avançando para os dados sobre a produção e por fim estruturar o top 5.

O primeiro passo foi realizar a ligação entre a tabela de atores com a tabela intermediária prod\_audio\_ator, na qual o id\_ator deveria ser igual em ambas. Esse primeiro "salto" foi feito através de um INNER JOIN.

```
SELECT
    ROUND(AVG(act.nota_imdb_ator)::numeric,2) as "Média de notas IMDB dos atores", --Média das notas
    prod_auvi.titulo as "Título da produção", --Título da obra
    prod_auvi.tipo as "Tipo da produção", --Tipo da obra
    prod_auvi.nota_imdb as "Nota IMDB da produção", --Opcional do grupo: nota da obra
    ppqnt.Vezes_Assistidas --Quantidade de vezes que a produção foi assistida (necessário para o top 5)
FROM ator act

INNER JOIN prod_audio_ator prod_auvi_act    -- Primeira ligação: Tabela ator com a
ON act.id_ator = prod_auvi_act.id_ator    -- tabela de relacionamento prod_audio_ator
```

O segundo passo foi interligar a tabela de relação prod\_audio\_ator com a tabela principal das produções audiovisuais, adquirindo assim dados como o título, tipo e nota IMDB dessas obras, esse processo foi simples, necessitando somente a construção de um outro INNER JOIN.

```
INNER JOIN prod_audiovisual prod_auvi          -- Após adquirir os IDs de cada ator,
on prod_auvi_act.id_producao = prod_auvi.id_producao -- liga os atores a produção audiovisual
```

Após isso, seria necessário ligar a tabela de produção audiovisual com "perfil\_prod" para criar então o top 5.

Nesse ponto, tivemos o desafio. Como agrupar as produções que mais foram vistas sem ocorrer a multiplicação das tuplas?

Era conhecido que para adquirir os top 5 itens mais vistos no ano 2020, bastava contar os dados existentes na tabela "perfil\_prod" e agrupar pelo próprio id

da produção, limitando, é claro a data de acessos para somente o ano de 2020, nessa perspectiva já tínhamos o seguinte:

```
(SELECT COUNT(perf_prod.id_producao) as Vezes_Assistidas,
      perf_prod.id_producao
FROM public.perfil_prod perf_prod
WHERE EXTRACT(year FROM perf_prod.ult_acesso)='2020' --Limita que a data de acesso seja somente em 2020
GROUP BY perf_prod.id_producao) ppqnt

ON(ppqnt.id_producao = prod_auvi.id_producao) --Liga o ID da produção da entidade prod_audiovisual com perfil_prod
```

Bastou então ligar esse select à um RIGHT JOIN (foi escolhido o right join pois os dados predominantes estavam vindo das relações anteriores, a saber, os dados sobre a nota média IMDB dos atores, o título da produção, o tipo e a nota IMDB da produção). Nessa perspectiva, tínhamos a ligação do top 5 com todo o resto:

```
RIGHT JOIN
  (SELECT COUNT(perf_prod.id_producao) as Vezes_Assistidas,
        perf_prod.id_producao
  FROM public.perfil_prod perf_prod
  WHERE EXTRACT(year FROM perf_prod.ult_acesso)='2020' --Limita que a data de acesso seja somente em 2020
  GROUP BY perf_prod.id_producao) ppqnt

  ON(ppqnt.id_producao = prod_auvi.id_producao) --Liga o ID da produção da entidade prod_audiovisual com perfil_prod
```

Por fim, bastou agrupar os dados pela quantidade de vezes assistidas, o título, o tipo e a nota IMDB da obra e ordenar com base nas vezes assistidas de forma descendente e limitada por tuplas.

Toda a query, ficou da seguinte forma:

```
SELECT
  ROUND(AVG(act.nota_imdb_ator)::numeric,2) as "Média de notas IMDB dos atores", --Média das notas
  prod_auvi.titulo as "Título da produção", --Título da obra
  prod_auvi.tipo as "Tipo da produção", --Tipo da obra
  prod_auvi.nota_imdb as "Nota IMDB da produção", --Opcional do grupo: nota da obra
  ppqnt.Vezes_Assistidas --Quantidade de vezes que a produção foi assistida (necessário para o top 5)
FROM ator act
INNER JOIN prod_audio_ator prod_auvi_act -- Primeira ligação: Tabela ator com a
  ON act.id_ator = prod_auvi_act.id_ator -- tabela de relacionamento prod_audio_ator
INNER JOIN prod_audiovisual prod_auvi -- Após adquirir os IDs de cada ator,
  on prod_auvi_act.id_producao = prod_auvi.id_producao -- liga os atores a produção audiovisual
RIGHT JOIN
  (SELECT COUNT(perf_prod.id_producao) as Vezes_Assistidas,
        perf_prod.id_producao
  FROM public.perfil_prod perf_prod
  WHERE EXTRACT(year FROM perf_prod.ult_acesso)='2020' --Limita que a data de acesso seja somente em 2020
  GROUP BY perf_prod.id_producao) ppqnt
  ON(ppqnt.id_producao = prod_auvi.id_producao) --Liga o ID da produção da entidade prod_audiovisual com perfil_prod
GROUP BY (ppqnt.Vezes_Assistidas, prod_auvi.titulo, prod_auvi.tipo, prod_auvi.nota_imdb)
ORDER BY ppqnt.Vezes_Assistidas DESC LIMIT 5 --Top 5 (maior para o menor)
```

Essa consulta resultou nos seguintes dados:

| Média de notas IMDB dos atores<br>numeric | Título da produção<br>character varying (100) | Tipo da produção<br>character varying | Nota IMDB da produção<br>numeric | vezes assistidas<br>bigint |
|---|---|---------------------------------------|----------------------------------|----------------------------|
| 7.00                                      | Bear Grylls - A prova de tudo                 | Documentário                          | 7.8                              | 15                         |
| 8.50                                      | F is for Family                               | Série                                 | 7.2                              | 11                         |
| 10.00                                     | Homem de Ferro                                | Filme                                 | 9.2                              | 11                         |
| 9.80                                      | Project Power                                 | Filme                                 | 7.3                              | 11                         |
| 7.90                                      | Animais Fantásticos e Onde H...               | Filme                                 | 9.3                              | 10                         |

#### 4. CONSIDERAÇÕES FINAIS

Os comandos referentes à criação das tabelas geradas pelo **StarUml** com base no modelo relacional, bem como os dados inseridos pelo grupo e eventuais consultas podem ser acessadas na pasta de consultas, juntamente com esse arquivo.

Nosso github possui todos os dados também: <https://github.com/Clube-do-Enrolado/BancoDeDados-FeiFlix>

Além disso, os MERs podem ser vistos nos seguintes links:

(Etapa 1): <https://lucid.app/lucidchart/invitations/accept/8f4f0df9-aca5-4a61-93f7-16186c3fd46a>

(Etapa 2): <https://lucid.app/lucidchart/invitations/accept/2cc7de07-5938-43b6-8287-1e86969d544c>

Obrigado pela atenção :)