

POPCORN GO

Plataforma de *Streaming*

Frederico Oliveira (201904580)
Nuno Jesus (201905477)
Tomás Martins (201704976)

Índice

1. Contexto.....	3
2. Diagrama UML.....	4
3. Diagrama UML Revisto.....	5
4. Esquema Relacional.....	6
5. Dependências Funcionais e Formas Normais.....	7
6. Restrições.....	9
7. Interrogações.....	12
8. Gatilhos.....	13
9. Participação.....	14

1. Contexto

POPCORN GO é uma plataforma online que permite visualizar conteúdo multimédia (**séries** e **filmes** dos mais variados tipos). Pretendemos, por isso, criar uma base de dados para gerir a logística inerente à sua atividade e armazenar a informação referente à mesma.

Dos **utilizadores** da plataforma pretende-se armazenar um nome de utilizador, um endereço de email, a data de nascimento, o nome completo e uma palavra-passe com pelo menos 8 caracteres. Para utilizar a plataforma, o utilizador precisa de uma **subscrição**, que tem uma data de início e uma data de fim, podendo ter uma duração de 1, 6 ou 12 meses. Para pagar a subscrição, armazenam-se também dados do **cartão de crédito**, nomeadamente, número do cartão, PIN de segurança, data de validade e nome do titular do cartão.

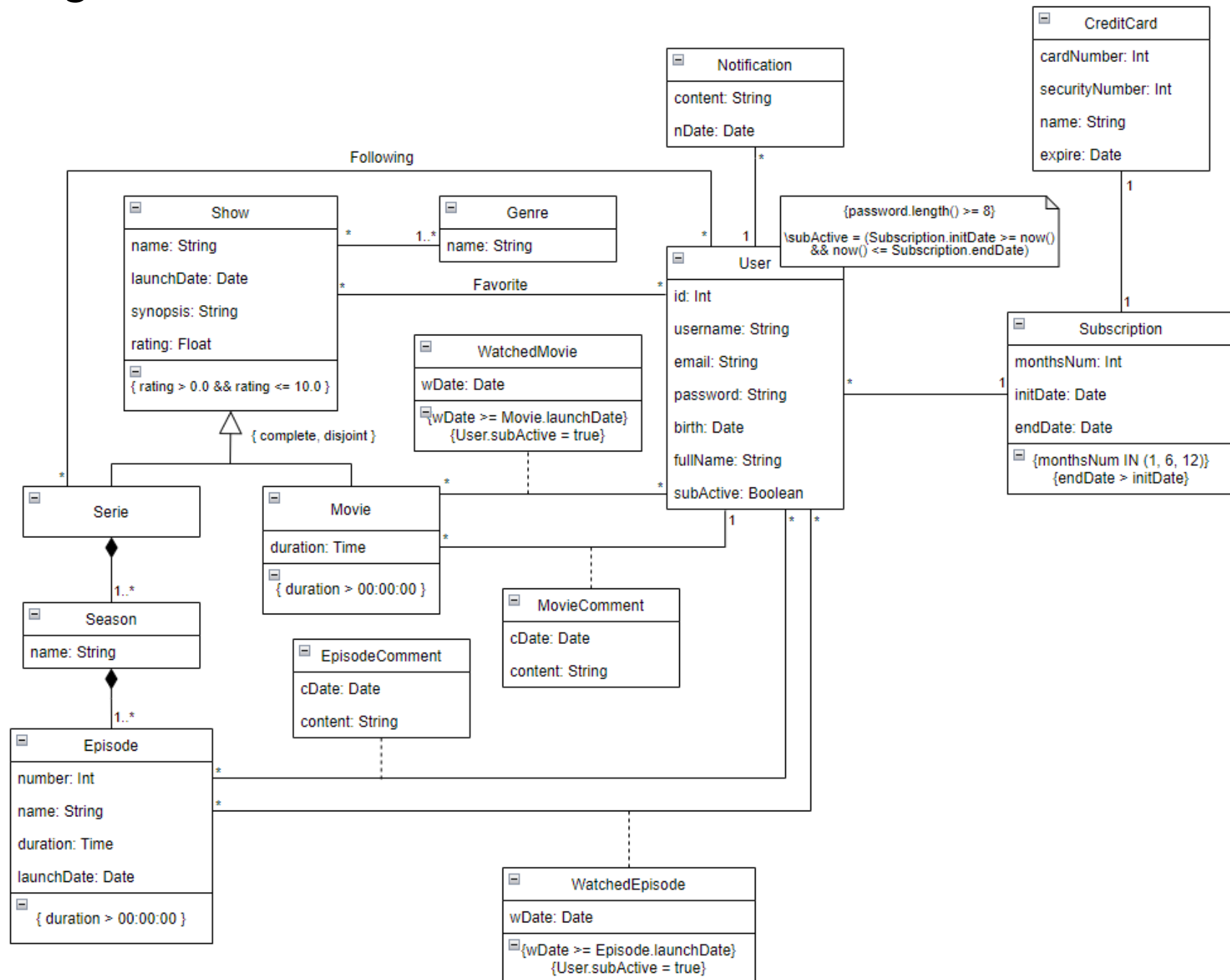
Apenas quando possui uma subscrição, é que o utilizador pode assistir a programas.

A plataforma tem **programas** de vários **géneros** dos quais se sabem o nome, data de lançamento, sinopse e pontuação de 0.0 a 10.0. Os programas podem ser **filmes**, dos quais se sabe a duração, ou **séries**, compostas por **temporadas**, das quais se sabe o número, e estas compostas por **episódios**, dos quais se sabem o número, o nome, a duração e a data de lançamento.

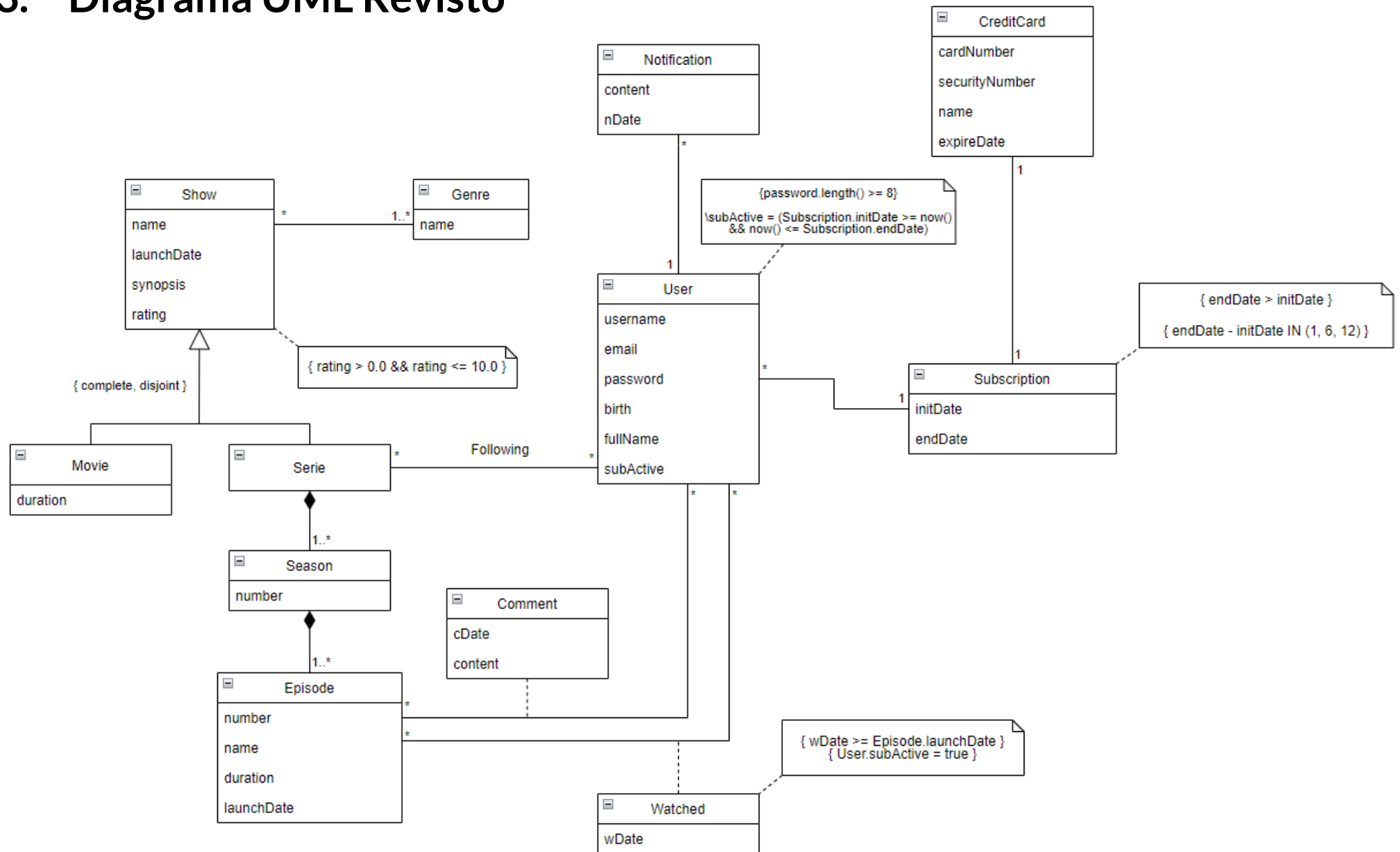
No caso das séries, o utilizador pode escolher seguir a série para ser **notificado** do lançamento de novos episódios, pode comentar episódios e responder a comentários de outros utilizadores ou dele próprio ou então marcar episódios como vistos à medida que vai vendo, para mais tarde retomar a partir do episódio em que ficou.

Além de ser notificado de novos episódios, pode ser notificado quando a subscrição estiver perto da data limite ou o cartão de crédito associado perto da data de validade.

2. Diagrama UML



3. Diagrama UML Revisto



4. Esquema Relacional

- **User**(id, username, email, password, birth, fullName, subActive, id_Subscription → Subscription)
- **Notification**(id, content, nDate, id_User → User)
- **Subscription**(id, initDate, endDate, id_CreditCard → CreditCard)
- **CreditCard**(id, cardNumber, securityNumber, name, expireDate)
- **Show**(id, name, launchDate, synopsis, rating)
- **Movie**(id → Show, duration)
- **Serie**(id → Show)
- **Season**(id, number, id_Serie → Serie)
- **Episode**(id, number, name, duration, launchDate, id_Season → Season)
- **Genre**(id, name)
- **Following**(id_Serie → Serie, id_User → User)
- **Comment**(id_Episode → Episode, id_User → User, cDate, content)
- **Watched**(id_Episode → Episode, id_User → User, wDate)
- **ShowGenre**(id_Show → Show, id_Genre → Genre)

5. Dependências Funcionais e Formas Normais

User(id, username, email, password, birth, fullName, subActive, id_Subscription → Subscription)

- {id} → {username, email, password, birth, fullName, subActive, id_Subscription}
- {username} → {id, email, password, birth, fullName, subActive, id_Subscription}
- {email} → {id, username, password, birth, fullName, subActive, id_Subscription}

Notification(id, content, nDate, id_User → User)

- {id} → {content, nDate, id_User}

Subscription(id, initDate, endDate, id_CreditCard → CreditCard)

- {id} → {initDate, endDate, id_CreditCard}
- {id_CreditCard} → {id, initDate, endDate}

CreditCard(id, cardNumber, securityNumber, name, expireDate)

- {id} → {cardNumber, securityNumber, name, expireDate}
- {cardNumber} → {id, securityNumber, name, expireDate}

Show(id, name, launchDate, synopsis, rating)

- {id} → {name, launchDate, synopsis, rating}
- {name} → {id, launchDate, synopsis, rating}

Movie(id → Show, duration)

- {id} → {duration}

Serie(id → Show)

- Dependência Funcional Trivial

Season(id, name, id_Serie → Serie)

- {id} → {name, id_Serie}

Episode(id, number, name, duration, launchDate, id_Season → Season)

- {id} → {number, name, duration, launchDate, id_Season}

Genre(id, name)

- {id} → {name}
- {name} → {id}

Following(id_Serie → Serie, id_User → User)

- Dependência Funcional Trivial

Comment(id_Episode → Episode, id_User → User, cDate, content)

- {id_Episode, id_User} → {cDate, content}

Watched(id_Episode → Episode, id_User → User, wDate)

- {id_Episode, id_User} → {wDate}

ShowGenre(id_Show → Show, id_Genre → Genre)

- Dependência Funcional Trivial

5.1. Análise das Formas Normais

Segundo a noção de Forma Normal de Boyce-Codd, dadas as dependências funcionais de uma relação na forma $\overline{A} \rightarrow \overline{B}$, em que \overline{A} e \overline{B} são conjuntos de atributos dessa relação, então essa relação está em BCNF se, e só se $\overline{A} \rightarrow \overline{B}$ é trivial ou \overline{A} é uma superchave da relação. Visto que as relações **Serie**, **Following**, **ShowGenre** têm apenas dependências funcionais triviais e as restantes relações têm dependências geradas apenas com recurso a superchaves, então todas as tabelas estão em BCNF, pelo que o esquema relacional está em BCNF.

6. Restrições

User:

- Não podem existir dois utilizadores (instâncias de **User**) com o mesmo **ID** (**PRIMARY KEY**).
- Não podem existir dois utilizadores com o **username** ou o mesmo **email** (**UNIQUE**).
- Cada utilizador tem, obrigatoriamente, de ter 1 único **username**, 1 único **email**, 1 única **password** e 1 estado da subscrição (**subActive** está ativa ou inativa) pelo que são atributos que têm de estar preenchidos com valores do tipo respectivo (**NOT NULL**).
- **id_Subscription** refere-se à subscrição do utilizador (**REFERENCES** ou **FOREIGN KEY** ou **CHAVE ESTRANGEIRA**).
- **password** tem que ter 8 ou mais caracteres (**CONSTRAINT validPass CHECK LENGTH(password)> 8**).

Notification:

- Não podem existir duas notificações (instâncias de **Notification**) com o mesmo **ID** (**PRIMARY KEY**).
- Não podem existir notificações vazias (**content**) sem dados (**NOT NULL**).
- **id_User** faz referência ao **User** que recebe a notificação (**Notification**) da qual **id_User** faz parte (**REFERENCES** ou **FOREIGN KEY** ou **CHAVE ESTRANGEIRA**).

Subscription:

- Não podem existir duas subscrições (tipos de **Subscription**) com o mesmo **ID** (**PRIMARY KEY**).
- Não podem existir duas subscrições com o mesmo **id_CreditCard** (**UNIQUE**).
- **initDate**, **endDate** e **id_CreditCard** têm que, obrigatoriamente, ter um valor associado (**NOT NULL**).
- **initDate** tem que ser menor que **endDate** (**CONSTRAINT validDate CHECK (DATE(initDate) < DATE(endDate))**).
- O número de meses subscrito tem que ser 1, 6, 12 meses (1 ano) (**CONSTRAINT numMonthsSubscribed CHECK (DATE(endDate) IN (DATE(initDate, '+1 month'), DATE(initDate, '+6 months'), DATE(initDate, '+1 year'))**)).
- **id_CreditCard** faz referência ao **CreditCard** que é utilizado para pagar a respetiva **Subscription** (**REFERENCES** ou **FOREIGN KEY** ou **CHAVE ESTRANGEIRA**).

CreditCard:

- Não podem existir dois cartões de crédito (instâncias de **CreditCard**) com o mesmo **ID** (**PRIMARY KEY**).
- Não podem existir dois cartões de crédito com número de cartão (**cardNumber**) igual (**UNIQUE**).
- Os atributos **cardNumber**, **name**, **securityNumber** e **expiryDate** têm, obrigatoriamente, que estar preenchidos (**NOT NULL**) com informação do tipo respetivo do atributo.

- O atributo **cardNumber** tem alocação para 16 dígitos no máximo (**INT(16,0)**), mas tem que ter exatamente 16 dígitos (**CONSTRAINT validCardNumber CHECK (cardNumber > 9999999999999999)**).
- O atributo **securityNumber** tem alocação para 3 dígitos no máximo (**INT(3,0)**), mas tem que ter exatamente 3 dígitos (**CONSTRAINT validSecurityNumber CHECK (securityNumber > 99)**).

Show:

- Não podem existir dois programas (instâncias de **Show**) com o mesmo **ID** (**PRIMARY KEY**).
- Os atributos **name**, **launchDate**, **synopsis**, **rating** têm, obrigatoriamente, que estar preenchidos (**NOT NULL**) com informação do tipo respetivo do atributo.
- A pontuação (**rating**) de um **Show** tem que ser maior ou igual a 0.0 e menor ou igual a 10.0 (**CONSTRAINT validRating CHECK (rating >= 0.0 and rating <= 10.0)**).
- Não podem existir duas instâncias de **Show** com o mesmo nome (**name**) (**UNIQUE**).

Movie:

- Não podem existir dois filmes (instâncias de **Movie**) com o mesmo **ID** (**PRIMARY KEY**).
- A duração (**duration**) de um **Movie** não pode ser nula (**NOT NULL**).
- **ID** faz referência ao **Show** do qual **Movie** faz parte (**REFERENCES** ou **FOREIGN KEY** ou **CHAVE ESTRANGEIRA**).
- **duration** não pode ter duração nula (**CONSTRAINT validDuration CHECK(duration != '00:00:00')**).

Serie:

- Não podem existir duas séries (instâncias de **Serie**) com o mesmo **ID** (**PRIMARY KEY**).
- **ID** faz referência ao **Show** do qual **Serie** faz parte (**REFERENCES** ou **FOREIGN KEY** ou **CHAVE ESTRANGEIRA**).

Season:

- Não podem existir duas temporadas (instâncias de **Season**) com o mesmo **ID** (**PRIMARY KEY**).
- O número da **Season** (**number**) tem que, obrigatoriamente, ter um valor associado (**NOT NULL**).
- **id_Serie** faz referência à **Serie** da qual a **Season** em questão faz parte (**REFERENCES** ou **FOREIGN KEY** ou **CHAVE ESTRANGEIRA**).
- **number** tem que ser maior que 0 (**CONSTRAINT validSeasonNumber CHECK (number > 0)**).

Episode:

- Não podem existir dois episódios (instâncias de **Episode**) com o mesmo **ID** (**PRIMARY KEY**).
- Os atributos **number**, **name**, **duration** e **launchDate** têm que ter valores associados do tipo respetivo (**NOT NULL**).

- **id_Season** faz referência à **Season** do **Show** do qual o **Episode** faz parte (**REFERENCES** ou **FOREIGN KEY** ou **CHAVE ESTRANGEIRA**).
- A duração de um episódio não pode ser nula (**CONSTRAINT validDuration CHECK (duration != '00:00:00')**).

Genre:

- Não podem existir dois gêneros (instâncias de **Genre**) com o mesmo **ID** (**PRIMARY KEY**).
- Não podem existir dois **Genre** com nomes (**name**) iguais (**UNIQUE**) e **name** tem, obrigatoriamente, de estar preenchido com valor do tipo respectivo do atributo (**NOT NULL**).

Following:

- **id_Serie** faz referência à **Serie** a ser seguida e **id_User** faz referência ao **User** a seguir determinada **Serie** (**REFERENCES** ou **FOREIGN KEY** ou **CHAVE ESTRANGEIRA**).
- Por cada par **User-Serie**, só pode existir uma instância de **Following** (**PRIMARY KEY**).

Comment:

- **id_Episode** faz referência ao **Episode** em que o comentário (instância de **Comment**) foi feito e **id_User** faz referência ao **User** que fez esse mesmo **Comment** (**REFERENCES** ou **FOREIGN KEY** ou **CHAVE ESTRANGEIRA**).
- O conteúdo (**content**) e data (**cDate**) de um **Comment** têm, obrigatoriamente, que ter valor atribuído do tipo do atributo (**NOT NULL**).
- Cada **Comment** não pode ter mais do que um par **Episode-User** associado (**PRIMARY KEY**).

Watched:

- **id_Episode** faz referência ao **Episode** em que foi visto (instância de **Watched**) foi feito e **id_User** faz referência ao **User** que viu esse mesmo **Episode** (**REFERENCES** ou **FOREIGN KEY** ou **CHAVE ESTRANGEIRA**).
- A data de visualização (**wdate**) de um **Episode** tem, obrigatoriamente, que ter valor atribuído do tipo do atributo (**NOT NULL**).
- Cada instância de **Watched** não pode ter mais do que um par **Episode-User** associado (**PRIMARY KEY**).

ShowGenre:

- **id_Show** faz referência ao **Show** classificado com um **Genre** e **id_Genre** faz referência ao **Genre** usado para classificar o **Show** da instância (**REFERENCES** ou **FOREIGN KEY** ou **CHAVE ESTRANGEIRA**).
- Cada instância de **ShowGenre** não pode ter mais do que um par **Show-Genre** associado (**PRIMARY KEY**).
- Cada instância de **Show** tem que ter pelo menos 1 instância de **Genre** associada.

7. Interrogações

1. Listar programa de um género específico (no caso, *Western*).
2. Listar episódios que ainda não saíram.
3. Listar programa que contêm a palavra “game” no seu título.
4. Listar quantos utilizadores que não seguem a série “Loki”.
5. Listar os episódios da primeira temporada de “Raised by Wolves”.
6. Listar a duração média (em minutos) dos filmes por cada género.
7. Listar quais os comentários que foram postados antes de um episódio ser lançado.
8. Listar os nomes dos 10 utilizadores que mais episódios viram, por ordem descendente.
9. Listar todos os episódios que foram vistos por cada utilizador no último dia das suas subscrições.
10. Listar todos os utilizadores que apesar de possuírem uma subscrição ativa, não possuem qualquer tipo de interação com o conteúdo multimédia.

8. Gatilhos

1. Notificar o utilizador quando novo episódio for lançado. (**NOTA:** é utilizado o *auto increment* no *id* da tabela **Notification** para inserir uma nova notificação, de modo a não inserir o *id* manualmente).
2. Um utilizador tem que possuir uma subscrição ativa para visualizar episódios de uma série e não pode ver episódios ainda por lançar.
3. Quando um programa for adicionado, tem que ter pelo menos um género associado.

9. Participação dos elementos do grupo

Apesar do grupo não participar de forma homogénea ao longo de cada entrega, consideramos que as diferentes intervenções ao longo do trabalho complementam-se, tornando a contribuição de cada elemento balanceada.