



Projeto de BDAD, 2ª entrega  
2019 - 2020

Grupo 702

Jéssica Nascimento - up201806723

Rafael Cristino - up201806680

Xavier Pisco - up201806134

# Índice

Contexto	2
Diagrama UML	4
Esquema Relacional	5
Análise das Dependências Funcionais	6
Análise das Formas Normais	7
Lista e Forma de Implementação de Restrições	9

# Contexto

O **IMBd<sub>ad</sub>** (Internet Movie Base de Dados) é uma base de dados destinada a guardar informações sobre **filmes (Movie)**, **séries (Series)** e os seus **atores** e **produtores** (ambos da classe **Celebrity**).

Para além desta informação existem os **utilizadores (User)** da plataforma, que podem criar **listas (List)** e **avaliar (Review)**. Cada **utilizador** é reconhecido pelo seu **email** e pode ter uma **foto de perfil** e é habilitado a criar **listas** de **filmes** e/ou **séries** que são identificadas por um **nome** e, conforme a escolha do utilizador, podem ser **públicas** ou **privadas**.

Uma **Review** é caracterizada por **título**, **texto** e **pontuação**. Ela relaciona-se com o utilizador que a criou e com a produção ou episódio a que se refere (tem que estar relacionada obrigatoriamente ou com uma produção ou com um episódio). A partir da pontuação de todas as reviews sobre uma determinada produção ou episódio, calcula-se a pontuação do/a mesmo/a.

Tanto os **utilizadores** como as **celebridades** são classes filhos/filhas da classe **person (Person)**, a qual guarda informação sobre o seu **nome** e a sua **data de nascimento**.

Cada celebridade pode participar numa **produção (Production)** ou num **episódio (Episode)** específico com qualquer **Role** (através de relações entre as classes, ao que melhor se aplicar - se apenas participar num episódio de uma série, apenas se relacionará com esse episódio).

Estas relações distinguem-se da seguinte forma: uma participação num episódio específico (não recorrente) é caracterizada por uma relação tripla entre **Episode**, **Celebrity** e **Role** (se o role for relacionado com atuar no episódio, o nome da personagem fica guardado na classe de associação da relação como **characterName**); a participação numa produção é caracterizada por uma relação tripla entre **Production**, **Celebrity** e **Role** (se o role for relacionado com atuar no episódio, o nome da personagem fica guardado na classe de associação da relação como **characterName**);

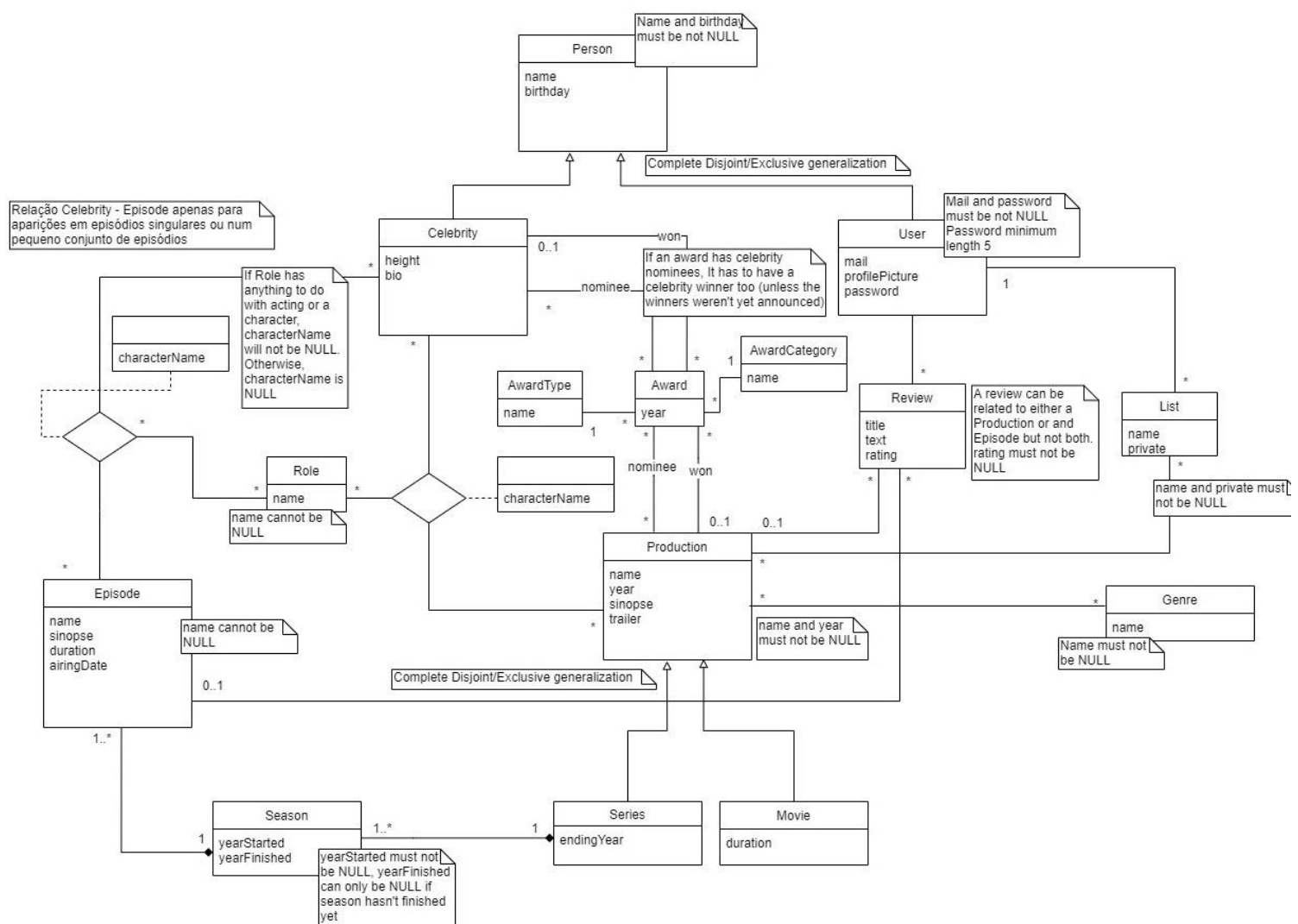
Os **filmes** e as **séries** são classes filhos/filhas da classe **produção (Production)** a qual guarda informação sobre o seu **nome**, **ano**, **sinopse** e o **trailer** e cada um está relacionado com uma ou mais **géneros (Genre)**. Por sua vez, os

**filmes** têm o atributo **duração** enquanto as **séries** têm o **ano de término** e estão associadas à classe **temporada (Season)**. Esta classe é composta pelos atributos **ano de começo** e **ano de término** e associa-se também aos **episódios (Episode)** que dela fazem parte, os quais são compostos pelo seu **nome**, **sinopse**, **duração** e **data de estreia**.

Por fim, as classes **prémio (Award)**, **tipo de prémio (AwardType)** e **categoria do prémio (AwardCategory)** são utilizadas para guardar informações sobre os prémios ganhos pelos **filmes**, **séries** e **celebridades** sendo que o tipo se refere ao nome do prémio, por exemplo, Óscar, e a categoria à sua categoria, por exemplo, melhor ator.

A classe **Award** relaciona-se com exatamente um **AwardType** e uma **AwardCategory** e possui o atributo **year** (Ano em que o prémio foi vencido). Para além disso, relaciona-se com uma ou nenhuma produção e com uma ou nenhuma celebridade (caso o prémio for, por exemplo, melhor ator, este relaciona-se com uma celebridade e uma produção; se for melhor filme apenas se relaciona com a produção; se não se relacionar nem com uma produção nem com uma celebridade, significa que o vencedor ainda não foi anunciado (no entanto, podem já haver nomeados)) através da relação **won** que identifica os vencedores do prémio. Com a relação **nominee**, o prémio relaciona-se com todas as celebridades e produções que foram/são nomeadas para o receber.

# Diagrama UML revisto



# Esquema Relacional

**Person** (id, name, birthday)

**Celebrity** (personId -> Person, height, bio)

**User** (personId -> Person, mail, profilePicture, password)

**Production** (id, name, year, sinopse, trailer)

**Movie** (productionId -> Production, duration)

**Series** (productionId -> Production, endingYear)

**Season** (id, seriesId -> Series, yearStarted, yearFinished)

**Episode** (id, seasonId -> Season, name, sinopse, duration, airingDate)

**Role** (id, name)

**ProductionRole** (productionId -> Production, celebId -> Celebrity, roleId -> Role, characterName)

**EpisodeRole** (episodId -> Episode, celebId -> Celebrity, roleId -> Role, characterName)

**Review** (id, userId -> User, title, text, rating)

**ReviewProduction** (reviewId -> Review, productionId -> Production)

**ReviewEpisode** (reviewId -> Review, episodId -> Episode)

**List** (id, name, personId -> User, private)

**BelongsToList** (listId -> List, productionId -> Production)

**Genre** (id, name)

**BelongsToGenre** (productionId -> Production, genreId -> Genre)

**Award** (id, year, productionId -> Production, awardTypeId -> awardType, awardCategoryId -> AwardCategory, celebId -> Celebrity)

**AwardType** (id, name)

**AwardCategory** (id, name)

**NomineeProduction** (awardId -> Award, productionId -> Production)

**NomineeCelebrity** (awardId -> Award, celebId -> Celebrity)

# Análise das Dependências Funcionais

**Person:** {id} -> {name, birthday}

**Celebrity:** {personId} -> {height, bio}

**User:** {personId} -> {mail, username, profilePicture, password}  
{mail} -> {personId, username, profilePicture, password}

**Production:** {id} -> {name, year, sinopse, trailer}

**Movie:** {productionId} -> {duration}

**Series:** {productionId} -> {endingYear}

**Season:** {id} -> {seriesId, yearStarted, yearFinished}

**Episode:** {id} -> {seasonId, name, sinopse, duration, airingDate}

**Role** {id} -> {name}

**ProductionRole** {productionId, celebId, roleId} -> {characterName}

**EpisodeRole** {episodeId, celebId, roleId} -> {characterName}

**Review:** {id} -> {userId, title, text, rating}

**ReviewProduction:** {reviewId} -> {productionId}

**ReviewEpisode:** {id} -> {episodeId}

**List:** {id} -> {name, personId, private}

**BelongsToList:** {listId, productionId} -> {}

**Genre:** {name} -> {id}

**BelongsToGenre:** {productionId, genreId} -> {}

**Award** {id} -> {year, productionId, awardTypeId, awardCategoryId, celebId}

**AwardType** {id} -> {name}

**AwardCategory** {id} -> {name}

**NomineeProduction** {awardId, productionId} -> {}

**NomineeCelebrity** {awardId, celebId} -> {}

# Análise das Formas Normais

Analisando os fechos de cada dependencia:

**Person:** {id}+ = {id, name, birthday}

**Celebrity:** {personId}+ = {personId, height, bio}

**User:** {personId}+ = {personId, mail, profilePicture, password}  
          {mail}+ = {personId, mail, profilePicture, password}

**Production:** {d}+ = {id, name, year, sinopse, trailer}

**Movie:** {productionId}+ = {productionId, duration}

**Series:** {productionId}+ = {productionId, endingYear}

**Season:** {id}+ = {id, seriesId, yearStarted, yearFinished}

**Episode:** {id}+ = { id, seasonId, name, sinopse, duration, airingDate}

**Role** {id}+ -> {id, name}

**ProductionRole** {productionId, celebId, roleId}+ -> {productionId, celebId, roleId, characterName}

**EpisodeRole** {episodId, celebId, roleId}+ -> {episodId, celebId, roleId, characterName}

**Review:** {id}+ = {id, userId, title, text, rating}

**ReviewProduction:** {reviewId}+ = {reviewId, productionId}

**ReviewEpisode:** {reviewId}+ = {reviewId, episodId}

**List:** {id}+ = {id,, name, personId, private}

**BelongsToList:** {listId, productionId}+ = {listId, productionId}

**Genre:** {id}+ = {name, id}

**BelongsToGenre:** {productionId, genreId}+ = {productionId, name}

**Award** {id}+ = {id, year, productionId, awardTypeId, awardCategoryId, celebId}

**AwardType** {id}+ = {id, name}

**AwardCategory** {id}+ = {id, name}

**NomineeProduction** {awardId, productionId}+ = {awardId, productionId}

**NomineeCelebrity** {awardId, celebId}+ = {awardId, celebId}



Podemos concluir que todas as relações respeitam a Terceira Forma Normal uma vez que seguem todas as seguintes regras:

- Todos os atributos contêm valores atômicos e os seus valores são singulares (1NF);
- Todos os atributos membros que fazem parte de uma chave são primos (2NF);
- Não existem dependências transitivas, ou seja, um atributo depende exclusivamente de uma chave primária (3NF).

Em todas as relações, para cada dependência, o lado esquerdo corresponde a uma superchave. Assim sendo, para além da Terceira Forma Normal, todas as relações se encontram na Forma Normal Boyce-Codd.

# Lista e Forma de Implementação de Restrições

## Person:

- id PRIMARY KEY
- name, birthday NOT NULL

## Celebrity:

- personId PRIMARY KEY
- personId FOREIGN KEY

## User:

- personId PRIMARY KEY
- personId FOREIGN KEY
- mail UNIQUE
- mail NOT NULL
- password NOT NULL
- password CHECK (length > 5)

## Production:

- id PRIMARY KEY
- name, year NOT NULL

## Movie:

- productionId PRIMARY KEY
- productionId FOREIGN KEY

## Series:

- productionId PRIMARY KEY
- productionId FOREIGN KEY

## Season:

- id PRIMARY KEY
- seriesId FOREIGN KEY
- seriesId NOT NULL
- yearStarted NOT NULL

## Episode:

- id PRIMARY KEY
- seasonId FOREIGN KEY
- seasonId NOT NULL
- name NOT NULL

## Role:

- id PRIMARY KEY
- name NOT NULL
- name UNIQUE

## ProductionRole:

- productionId, celebId, roleId PRIMARY KEY
- productionId, celebId, roleId FOREIGN KEY
- productionId NOT NULL
- celebId NOT NULL
- roleId NOT NULL

## EpisodeRole:

- episodeId, celebId, roleId PRIMARY KEY

## Review:

- id PRIMARY KEY
- userId FOREIGN KEY
- userId NOT NULL
- rating NOT NULL CHECK(rating > 0 and rating <= 10)

## ReviewProduction:

- reviewId PRIMARY KEY
- reviewId, productionId FOREIGN KEY
- reviewId NOT NULL
- productionId NOT NULL

## ReviewEpisode:

- reviewId PRIMARY KEY
- reviewId, episodeId FOREIGN KEY
- episodeId NOT NULL
- reviewId NOT NULL

## List:

- id PRIMARY KEY
- name NOT NULL
- private NOT NULL
- personId FOREIGN KEY

→ personId NOT NULL

→ awardId NOT NULL

**BelongsToList:**

→ listId, productionId PRIMARY KEY

→ listId, productionId FOREIGN KEY

→ listId NOT NULL

→ productionId NOT NULL

→ celebId NOT NULL

**Genre:**

→ Id PRIMARY KEY

→ name UNIQUE

→ name NOT NULL

**BelongsToGenre:**

→ productionId, genreId PRIMARY KEY

→ productionId, genreId FOREIGN KEY

→ productionId NOT NULL

→ genreId NOT NULL

**Award:**

→ id PRIMARY KEY

→ year NOT NULL

→ awardTypeId, awardCategoryId  
FOREIGN KEY

→ awardTypeId NOT NULL

→ awardCategoryId NOT NULL

**AwardType:**

→ id PRIMARY KEY

→ name NOT NULL

→ name UNIQUE

**AwardCategory:**

→ Id PRIMARY KEY

→ name NOT NULL

→ name UNIQUE

**NomineeProduction:**

→ awardId, productionId PRIMARY KEY

→ awardId, productionId FOREIGN KEY

→ awardId NOT NULL

→ productionId NOT NULL

**NomineeCelebrity:**

→ awardId, celebId PRIMARY KEY

→ awardId, celebId FOREIGN KEY