

Bases de Dados 2018/2019

## **Gestão de Streaming de Música**

**Grupo 710:**

**André Mamprin Mori** - up201700493

**Daniel Gazola Bradaschia** - up201700494

# Descrição

O tema escolhido para nosso projeto é uma base de dados para um serviço de Streaming de música (ex. Spotify, Apple Music).

A classe principal do projeto é a *User*. Considerando suas relações, a princípio um utilizador pode seguir outros utilizadores, *playlists* e artistas, sendo essa última relação do tipo não-recíproca. Nota-se que diversas informações são guardadas referentes ao utilizador, nomeadamente: nome, idade, email, imagem de perfil.

Por fim, existe uma relação de agregação entre as classes *User* e *Subscription*, sendo que a última armazena informações sobre a escolha de inscrição do utilizador, tais como o tipo, a taxa, método de pagamento e data de validade.

Em seguida, temos a classe *Playlist* que, como dito anteriormente, pode ser seguida por um utilizador assim como pode ser de pertence. A classe possui como atributos o nome, imagem de capa e descrição. É importante apontar a existência de uma subclasse nomeada *Top*, com atributo *data*, que se trata de uma *playlist* gerada a partir das músicas mais ouvidas.

Por definição, *playlists* são constituídas por uma ou mais músicas, que são representadas pela classe *Song*, com atributos de nome, tempo de duração, e contagem de reproduções. Existe uma relação direta com a classe *Top*, com a presença de uma classe de atributo na qual se armazena o índice da posição da música em um *ranking* de mais ouvidas.

A partir da relação de composição da classe *Song*, temos a classe *Album*, com seus respectivos atributos. Nota-se que esta classe é necessária para a organização e armazenamento do trabalho dos artistas, representados pela classe *Artist* e que tem uma relação de concatenação com *Album*.

A classe *Artist*, com estrutura composta por nome, imagem de perfil, imagem de capa e biografia, possui duas relações intrínsecas com duas outras: *Genre* e *Concert*. Da primeira temos uma classe que definirá uma característica fulcral de todo artista, gravando seu nome e uma breve descrição sobre seu estilo musical. Já da classe *Concert* temos uma relação direta em que artistas podem ter, ou não, um concerto agendado para o futuro.

Devido a relação naturalmente associativa entre *Artist* e *Concert* foi escolhido implementar uma classe *Location*, responsável por armazenar o local do concerto. Por fim, percebeu-se possíveis relações entre *Location*, *User*, *Artist* e *Top*, sendo relevante guardar a localização de cada uma dessas classes, assim foram estabelecidas tais relações.

# Atributos

**User:** *Name, Email, Profile Picture, Age*

**Subscription:** *Subscription type, Subscription Fee, Payment Method, Expiration Date*

**Playlist:** *Name, Cover Image, Description*

**Top:** *Date*

**Song:** *Name, Length, Play Counts*

**Album:** *Name, Number of Songs, Year, Label, Cover Picture*

**Artist:** *Name, Profile Picture, Cover Picture, Bio*

**Genre:** *Name, Description*

**Concert:** *Date*

**Location:** *City, Country*

