

Bases de Dados
*Management of gaming
streams*

2MIEICO1 - Group 106

March 8, 2020

Gabriel Rocco up201800172@fe.up.pt

Muriel Pinho up201700132@fe.up.pt

Description

This project is based on the management of a gaming stream service, similar to *Twitch*.

The database is divided into eight classes, these classes are:

- **User:** it stores all data related to the user, like his id, username, password, profilePicture, birthDate, email and age the latter being calculated based on the difference between his birth date and the current date. The **User** class has a direct relation with 4 other classes, a user can send and receive a **Message**, he can follow and own a **Channel**, and he can also follow a specific **Game**.
- **Stream:** the stream is the heart of this service, A stream is a live transmission made on a **Channel** owned by a **User** playing a **Game** for other users to watch. A Stream has 5 fields, title, startTime, uptime, ageRestriction and viewerCount. Title describes what the streamer is doing at that specific stream, startTime is the time the stream started, uptime is how much time has passed since the streamer started the stream, ageRestriction is used to protect children from watching inappropriate streams and viewerCount is the number of users watching that stream at the moment. This class relates to **Game**, with a stream having at least one **Game** being played at a time, but also the user can play multiple games over the course of the stream. A stream can also have Tags, describing what the stream is about, with all streams having at least one **Tag**.
- **Message:** it stores the data from messages sent by the **User**, storing the fields content and dateSent.

- **Channel:** a channel is owned by one user and it is where the user can create an **Stream**, the channel stores the fields biography and followerCount, with biography being a brief description of the channel and it's owner and followerCount being a field calculated based on the number of Follow relations a **Channel** has with other Users that are not the owner. It relates to **Stream** class as a channel can have multiple streams but all streams must be streamed on only one channel.
- **Game:** the game class stores all the information related to the games streamed on the platform, a game has a title, followerCount, description and totalViewers, with title and description identifying the game, followerCount being the number of **Users** that follow this game and totalViewers being the sum of all the viewerCount fields of **streams** from this game that are live.
- **Tag:** a tag is used to describe a **Stream**, it only has a title field and the same tag can be used on multiple streams at the same time.
- **TopGameStreams:** it is a association class with the fields viewerCount and position, being fields from a **Stream** that is ranked as a top **Stream** , this class is used to generate a list that contains the streams from a specific **Game** ranked in order based on the viewerCount from that specific **Game**.
- **TopTagStream:** it is identical to the TopGameStreams class, the only difference is that the ranking is bases on a **Tag** instead of a **Game**

Class Diagram - UML

