

Faculdade de Engenharia da Universidade do Porto



## Base de dados para um serviço de *streaming* de vídeo ao vivo

Bases de Dados 2023/24 - Licenciatura em Engenharia Informática e Computação

Grupo 504 - Turma 2LEIC05

### Estudantes & Autores:

Bruno Ricardo Soares Pereira de Sousa Oliveira [up202208700@up.pt](mailto:up202208700@up.pt)

Rodrigo Lourenço Ribeiro [up202206396@up.pt](mailto:up202206396@up.pt)

João Pedro Martins Mendes [up202208783@up.pt](mailto:up202208783@up.pt)

# Índice

Descrição do contexto.....	3
Modelo conceptual (versão inicial).....	4
Integração de ferramenta de IA generativa .....	5
Modelo conceptual (versão final) .....	6
Participação dos membros do grupo .....	7

## Descrição do contexto

No âmbito do projeto da unidade curricular de Bases de Dados, pretendemos desenvolver uma base de dados para um **serviço de *streaming* de vídeo ao vivo**. Esta base de dados deve armazenar informação relativa às contas dos utilizadores, vídeos, mensagens, subscrições e outros.

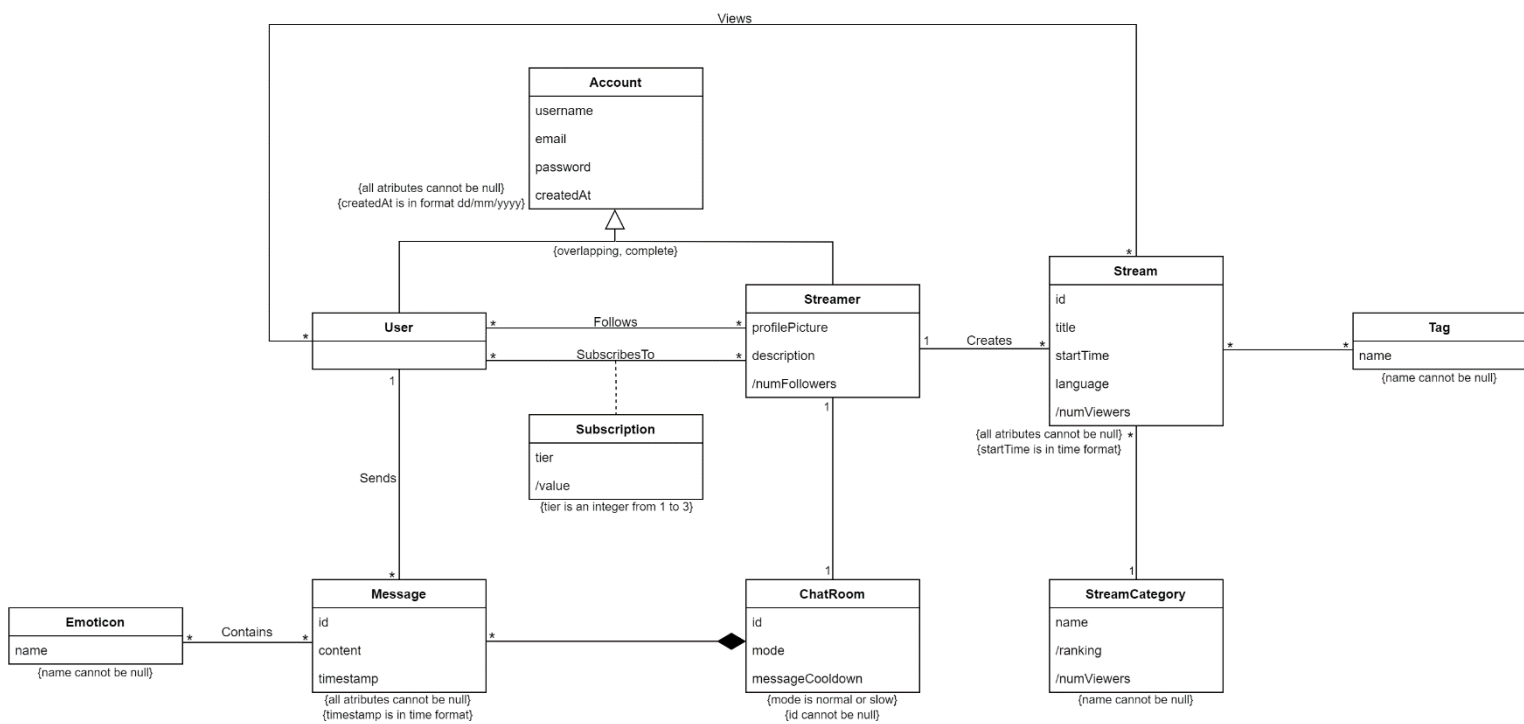
Para participar no serviço de *streaming*, é necessário criar uma **conta**, fornecendo um nome de utilizador, email e palavra-passe. A data de criação também será registada.

Cada conta pode ser de **usuário** e/ou **streamer**. Um *streamer* pode iniciar vídeos ao vivo, também chamados de **streams**, que os usuários conseguem assistir. Um usuário tem a opção de seguir *streamers*, ou pode também lhes subscrever, escolhendo um *tier* (de 1 a 3) que tem um preço associado. Quanto às contas de *streamer*, deve-se guardar adicionalmente uma foto de perfil, descrição e número de seguidores.

Cada *streamer* tem a sua própria **chat room** onde os usuários podem comunicar entre si através de mensagens. Estas têm associadas a hora em que foram enviadas e podem conter emoticons que são reconhecidos pelo seu nome. Uma *chat room* pode estar em dois modos diferentes: normal ou lento. Quando está no modo lento, existe um espaço de tempo entre o qual os usuários têm de esperar para enviar uma nova mensagem.

Um *streamer*, ao criar uma *stream*, deve indicar o título, as *tags* (que caracterizam o conteúdo), a linguagem e a categoria da *stream*, que será de uma lista fornecida pelo serviço. Cada *stream* tem associada uma hora de começo e número de visualizadores. Quanto às categorias de *stream*, deve-se guardar o nome, o número de visualizadores atuais da categoria e a posição no ranking, determinada pelo anterior.

## Modelo conceptual (versão inicial)



## Integração de ferramenta de IA generativa

A ferramenta que usámos nesta etapa foi o **ChatGPT**, um *chatbot* online de inteligência artificial, que comunica através de diálogos em texto com o utilizador.

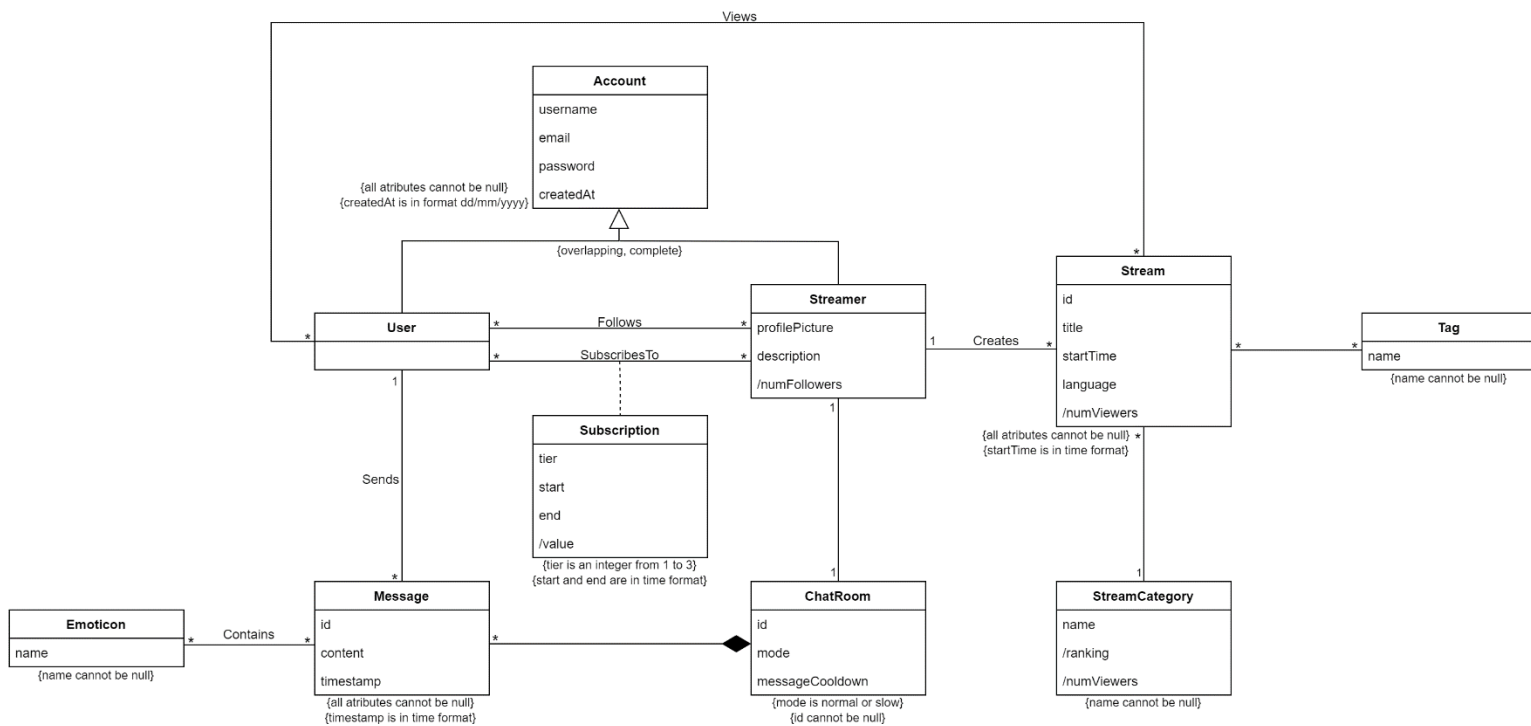
O primeiro *prompt* que demos à ferramenta foi: *“We are doing a project that requires us to create a simple database for a live video streaming service, like Twitch. Can you help us create a schema?”*. Como resposta, o ChatGPT deu um esquema com apenas três tabelas, *“Users”*, *“Streams”* e *“Comments”*, assim como alguns atributos para cada. Não obstante, esta solução apresentou-se demasiado simples para o nosso projeto.

O segundo *prompt* fornecido foi: *“The classes that we have and their attributes are Account(username, email, password, createdAt), Streamer(profilePicture, description, numFollowers), Subscription(tier, value), User(), Stream(id, title, startTime, language, numViewers), StreamCategory(name, ranking, numViewers), ChatRoom(id, mode, messageCooldown), Message(id, content, timestamp) and Emoticon(name). Can you recommend some changes to our attributes?”*. Desta vez, o ChatGPT respondeu com várias sugestões para os atributos de cada classe, e algumas das melhorias mostraram-se bastante oportunas. O exemplo mais notório foi a inclusão dos atributos *“start”* e *“end”* à classe *“Subscription”*, para registar o início e o fim, respetivamente, da subscrição.

O terceiro *prompt* fornecido foi: *“Can you make a UML conceptual model with this information?”*. A resposta do ChatGPT mencionava que elaborar um modelo em UML com texto não seria muito apropriado e apresentou, em alternativa, uma descrição com as classes, atributos e associações. As classes eram exatamente as referidas e as associações que a ferramenta forneceu tinham multiplicidades que não faziam muito sentido, como forçar um usuário a só seguir um *streamer*, pelo que a resposta não foi muito útil.

Concluindo, o ChatGPT, apesar de ter dado algumas sugestões bastante relevantes para o nosso modelo, grande parte da informação apresentada era repetida, genérica ou demasiado inconsistente para ser relevante. Apesar de não apresentar uma solução satisfatória ao problema sozinho, o ChatGPT mostrou o seu poder como uma ferramenta complementar ao trabalho, e as suas sugestões efetivamente se refletiram numa pequena melhoria do modelo conceptual, com a adição dos atributos *“start”* e *“end”* à classe *“Subscription”*.

## Modelo conceptual (versão final)



## Participação dos membros do grupo

Na primeira fase do projeto, todos os membros do grupo participaram de forma crítica na elaboração do esquema do modelo conceptual, dando sugestões para classes, associações e informação a armazenar em cada. Além disso, o Bruno foi responsável pela conceção do relatório e do modelo conceptual em UML, assim como deu vários retoques finais nas outras partes da entrega, o João elaborou a descrição do problema e fez alguns aperfeiçoamentos na estrutura do modelo conceptual e o Rodrigo encarregou-se da parte da integração da ferramenta de IA.