



# Serviço de Streaming de Música

Base de dados

Licenciatura em Engenharia Informática e Computação

Ana Belém Meireles Bilhoto - up202305035

Maria Luiza Henriques Vieira - up202304306

João Pedro Pinto Lunet - up202207150

# Índice:

<b>1. Contexto.....</b>	<b>2</b>
<b>2. Diagrama inicial do modelo conceptual (UML).....</b>	<b>3</b>
<b>2.1. Integração de Inteligência artificial.....</b>	<b>4</b>
1. Relacionamento entre "Música" e "Artista" .....	4
2. Nova Entidade "Ouvintes" .....	4
3. Detalhamento de "Ouvinte" e "Ouvinte de Podcasts" .....	5
4. Playlists e Colaboradores.....	5
5. Tipo de Assinatura (Premium/Gratuita).....	5
6. Histórico de Gostos e Seguintos.....	5
7. Atributos adicionais para "Playlist" e "Episódio" .....	6
8. Melhoria dos Nomes de Atributos e Entidades.....	6
9. Relacionamentos.....	6
<b>2.2. Diagrama final do modelo conceptual (UML).....</b>	<b>7</b>
<b>3. Avaliação do trabalho individual dos participantes.....</b>	<b>8</b>

# 1. Contexto

A plataforma que desenhamos é um serviço de streaming de música com funcionalidades semelhantes ao Spotify, que oferece aos utilizadores acesso a uma vasta biblioteca de conteúdos áudio e a oportunidade de interagir com outros utilizadores de diversas formas, tanto uns com os outros quanto com os criadores e artistas. Aquando a criação da conta, o utilizador insere o modo de contacto, ou seja, e-mail ou número de telefone, e escolhe o plano de assinatura.

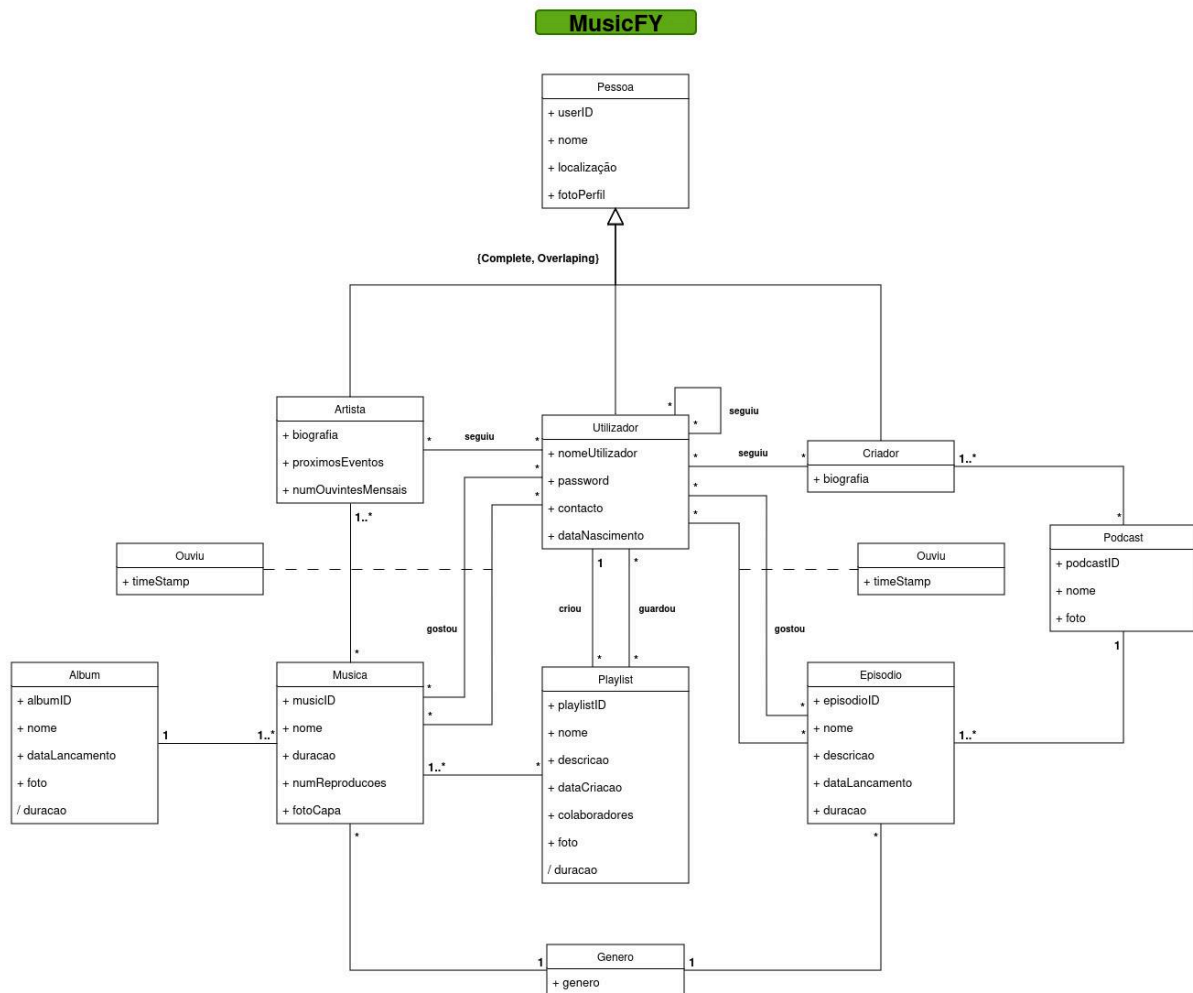
Entre as principais funcionalidades, os utilizadores podem seguir perfis de outros utilizadores e artistas, criar e partilhar playlists personalizadas, adicionar músicas a playlists e guardar playlists criadas por outros. Além disso, a plataforma permite que os utilizadores guardem nos seus favoritos não apenas músicas e álbuns, mas também episódios de podcasts.

Outra funcionalidade importante é o histórico de audição, que regista o conteúdo que o utilizador ouviu até ao momento, para além de guardar o progresso em episódios, permitindo assim um fácil acesso a músicas, álbuns e podcasts previamente ouvidos.

A partir de todos esses dados, como preferências musicais, interações com playlists e histórico de audição, a plataforma aplica algoritmos avançados para recomendar conteúdos que sejam do interesse do utilizador. Desta forma, é possível oferecer uma experiência personalizada, sugerindo músicas, álbuns, artistas e podcasts de acordo com o gosto individual de cada utilizador.

Além de melhorar a experiência de recomendação, essas informações também são utilizadas internamente para gerar estatísticas que permitem à plataforma refinar constantemente os seus serviços, ajustando funcionalidades e melhorando a interação entre utilizadores e artistas.

## 2. Diagrama inicial do modelo conceptual (UML)



## 2.1. Integração de Inteligência artificial

Para esta tarefa optamos por usar o ChatGPT, pois este utiliza o modelo IA mais avançado (GPT-4o mini).

### Prompt:

Olá! Em anexo está o UML para uma plataforma de streaming de música, parecida com o Spotify. Gostaríamos de saber que melhorias podemos fazer ao UML.

### Resposta:

Olá! Eu vi o diagrama UML que enviaste para a plataforma de streaming de música "MusicFY", que está bastante bem estruturado. No entanto, há algumas sugestões de melhoria que podem ajudar a refinar o modelo e torná-lo mais robusto:

#### 1. Relacionamento entre "Música" e "Artista"

- O relacionamento entre a classe "Música" e "Artista" está como 1..\* (um artista pode ter várias músicas, e uma música pode ter vários artistas). Isso faz sentido para colaborações entre artistas. Contudo, pode ser útil especificar se o artista é principal ou colaborador numa música. Podes adicionar um atributo extra ou uma nova associação para diferenciar.

**Avaliação Crítica:** Não acrescentamos essa sugestão, visto que na plataforma não há a diferenciação entre o artista principal e o artista colaborador. A música pertence igualmente a discografia de ambos.

#### 2. Nova Entidade "Ouvintes"

- **NumOuvintesMensais em Artista:** Se o campo "numOuvintesMensais" em "Artista" for atualizado frequentemente, pode ser mais eficiente mover isso para uma entidade separada que armazena essas estatísticas.

**Avaliação Crítica:** Optamos por seguir esta sugestão e criar uma classe *Ouvintes* com o atributo "numOuvintesMensais", uma vez que este atributo seria atualizado sempre que um novo utilizador ouvisse uma música do artista em questão, tornando assim a base de dados mais eficiente.

### 3. Detalhamento de "Ouvinte" e "Ouvinte de Podcasts"

- Atualmente, tanto a "Música" como o "Podcast" têm uma associação com "Ouviu" para registrar o histórico de reproduções. No entanto, o diagrama parece tratar episódios e músicas da mesma forma. Talvez fosse interessante separar melhor esses dois tipos de conteúdo, adicionando, por exemplo, um tipo de métrica diferente para cada um (como contagem de minutos ouvidos para episódios).

**Avaliação crítica:** Seguimos esta sugestão e acrescentamos o atributo "progressoEpisodio" na associação "Ouviu" entre o "Utilizador" e o "Podcast", para que assim o utilizador possa voltar a ouvir a partir do momento em que parou pela última vez.

### 4. Playlists e Colaboradores

- A classe "Playlist" tem um campo de "colaboradores". Uma playlist pode ter vários colaboradores, mas parece que esse relacionamento poderia ser melhor estruturado. Um modelo mais claro poderia ser a criação de uma tabela associativa entre "Utilizador" e "Playlist" para representar essa colaboração de forma mais explícita.

**Avaliação crítica:** Optamos por apenas alterar a multiplicidade da associação "criou" para que uma playlist pudesse ser criada por mais do que 1 utilizador.

### 5. Tipo de Assinatura (Premium/Gratuita)

- A classe "Utilizador" poderia ter um atributo para determinar o tipo de assinatura (gratuita ou premium), que pode alterar as funcionalidades disponíveis para o utilizador. Este campo pode influenciar a quantidade de saltos de música, a qualidade do áudio ou a exibição de anúncios.

**Avaliação crítica:** Adicionamos o atributo "planoAssinatura" à classe "Utilizador" permitindo agora diferenciar utilizadores com plano pago, ou gratuito.

### 6. Histórico de Gostos e Seguintes

- Existe um relacionamento entre o "Utilizador" e a "Música" (gostou), mas se um utilizador pudesse gostar ou seguir mais conteúdos, como álbuns, playlists, podcasts ou criadores, seria útil padronizar um histórico de interações para todos esses tipos de conteúdo. Isto facilitaria a expansão futura do modelo.

**Avaliação crítica:** Optamos por não fazer essa alteração uma vez que os gostos em cada conteúdo são guardados separadamente e de maneiras diferentes, sendo assim desnecessário um histórico de interações.

## 7. Atributos adicionais para "Playlist" e "Episódio"

- As playlists poderiam ter um campo para armazenar a duração total baseada na soma das durações das músicas. Também seria interessante adicionar um campo de visualizações para entender quantas vezes a playlist foi acedida.
- Para "Episódio", seria bom incluir uma relação com tags ou temas, o que ajudaria os utilizadores a encontrar episódios por tópicos de interesse.

**Avaliação crítica:** Procedemos à alteração do nome do atributo "duracao" para "duracaoTotal" de modo a tornar a análise do modelo conceptual mais simples e direta. Substituímos também a associação das classes "*Genero*" e "*Episodio*" pela criação da classe *Temas* que se associa por sua vez à classe "*Episodio*". A associação entre as classes "*Genero*" e "*Podcast*" também foi adicionada, permitindo assim uma maior diferenciação e organização entre Podcasts e os respectivos episódios.

## 8. Melhoria dos Nomes de Atributos e Entidades

- Nomes de atributos mais claros: Alguns nomes de atributos poderiam ser mais descritivos. Por exemplo, em vez de "duracao", poderias usar algo como "duracaoSegundos" para especificar a unidade da duração.

**Avaliação crítica:** Acabamos por seguir esta sugestão, uma vez que torna a interpretação do modelo conceptual mais fácil.

## 9. Relacionamentos

- Múltiplos Géneros: Parece que tanto "Música" quanto "Álbum" têm uma relação de 1 para 1 com "Género". Na prática, uma música ou álbum pode ter múltiplos géneros. Sugiro mudar o relacionamento para muitos para muitos.

**Avaliação crítica:** Neste caso decidimos adotar esta medida, visto que se assemelha mais à realidade do conceito de género musical, em comparação com a multiplicidade implementada anteriormente.

Estas são algumas melhorias que podem refinar o modelo UML, facilitando futuras expansões da plataforma. Se precisares de mais alguma ajuda ou se houver outra área do diagrama que gostarias de discutir mais a fundo, estou à disposição!





### 3. Avaliação do trabalho individual dos participantes

Trabalhamos em conjunto durante a maior parte do trabalho e dividimos algumas tarefas a serem feitas individualmente, que após concluídas discutimos entre nós e finalizamos o relatório em grupo.