

Bases de Dados

Gestão de Stream de Música

2MIEIC01 – Grupo 102

(12 de março de 2017)

Bárbara Silva	up201505628@fe.up.pt
Julieta Frade	up201506530@fe.up.pt
Miguel Fernandes	up201503538@fe.up.pt

Descrição

Este projeto baseia-se na gestão de um serviço de *stream* de música, semelhante ao *Spotify*.

Em primeiro lugar, consideremos a classe **Utilizador** e suas inter-relações. Um usuário da plataforma tem a possibilidade de seguir outros utilizadores e **Intérpretes**, sendo esta última relação não-recíproca. É-lhes também proporcionado um serviço de troca de **Mensagens** – instanciou-se uma classe de associação para memorização da sua data de envio e conteúdo.

A base de dados armazena uma quantidade de dados considerável relativamente ao **Utilizador**, como o seu id, *username*, foto de perfil e idade – um atributo derivado, calculado através da sua data de nascimento.

Cada **Utilizador** pode guardar diferentes **Dispositivos**, como telemóvel, computador ou tablet para poder usufruir do serviço em diferentes locais, sendo, obviamente, obrigatório o registo de pelo menos um.

Caso o usuário procure mais funcionalidades, existe ainda a opção de elevar o estado da conta para **Utilizador Premium**, à custa de uma mensalidade fixa. Este *upgrade* inclui a remoção de anúncios, descarregamento de músicas e tempo de reprodução ilimitado. Para distinguir os dois modelos de conta, idealizamos duas relações de generalização. A classe **Utilizador Free** mantém registo do tempo de audição decorrido; a classe **Utilizador Premium** guarda o valor da mensalidade, dado que diferentes planos de pagamento poderão resultar em tarifas distintas. Ambas agem como extensão da classe-mãe **Utilizador**.

De seguida, examinemos o conceito de **Playlist**. Esta estrutura possui um nome, imagem e descrição, todos atribuídos por um utilizador do sistema.

Assim, de facto, uma lista de reprodução estabelece duas relações particulares com um usuário: a de propriedade – um cliente pode ser dono de várias playlists, porém cada uma destas possui um só dono – e a de acompanhamento – um cliente pode seguir várias playlists distintas e, do mesmo modo, a playlist poderá ser seguida por vários clientes diferentes.

Uma **Playlist** é, evidentemente, constituída por **Músicas**, uma estrutura com nome, duração, número de reproduções e um **Género** musical predominante associados. Considerou-se fazer sentido, do ponto de vista conceptual, a noção de playlist apenas ser válida quando esta não se encontra vazia, justificando a multiplicidade 1..* utilizada.

Introduzido o conceito de **Música** nesta abstração, abordemos as suas ligações com as outras classes da nossa *database*. Uma música estabelece uma relação de composição com **Álbum**, gerando uma classe de associação no processo, que preserva a sua posição na lista de faixas. Evidencie-se o uso deste tipo de associação, que restringe uma música a unicamente um álbum. Não se considerou a existência de coletâneas por se tratar de um caso excessivamente particular e comprometer a interpretação lógica e imediata da estrutura.

Iremos agora aprofundar o conceito de **Intérprete**, cuja estrutura é composta por um nome, uma *flag* de verificado – que se trata de um booleano ativado para artistas que verificam o seu perfil – foto de perfil, de capa e biografia. Esta classe tem relações intrínsecas com duas outras: **Local** e **Concerto**.

É relevante apontar que, do mesmo modo que **Música** e **Álbum** estabelecem, entre si, uma relação de composição, também **Álbum** e **Intérprete**. Foi considerado que um grupo apenas poderá ser considerado, efetivamente, um grupo musical, se tiver produzido algum álbum. Note-se que se considerou um **Single** como um tipo de álbum. De tal forma, o conceito de **Intérprete** seria infundado aquando da existência

Dada a natureza associativa destas três classes, deduzimos que seria lógico integrá-las num tuplo. De tal forma, um **Intérprete** poderá ter (ou não) **Concertos** agendados para o futuro, cada um destes armazenando uma data e realizando-se num único **Local**. Para completar o tuplo, basta associar **Intérprete** ao seu **Local** de origem.

Por último, o sistema gera uma lista ordenada das músicas mais ouvidas num dado **Local**, o **Top** de músicas, no qual cada uma terá uma posição, ou seja, uma classificação entre todas as outras presentes no mesmo. Assim, associamos **Local** e **Música**, a partir de **Top**.

Atributos

Utilizador

- Nome
- *Username*
- *Password*
- Foto de Perfil
- Data de Nascimento
- *E-Mail*
- / Idade

Utilizador Premium

- Mensalidade

Utilizador Free

- Tempo Limite

Mensagem

- Conteúdo
- Data de Envio

Dispositivos

- Nome

Tipo de Dispositivo

- Nome do Tipo

Playlist

- Nome
- Imagem
- / Duração
- Descrição

Intérprete

- Nome
- Verificação
- Foto de Perfil
- Foto de Capa
- Biografia

Álbum

- Nome
- Capa
- Ano

Tipo de Álbum

- Nome do Tipo

Música

- Nome
- Duração
- Reproduções
- Género

Género

- Nome

Concerto

- Data

Local

- Cidade
- País

Top

- Posição

Diagrama de Classes – UML

