

# PROGRAMAÇÃO ORIENTADA A OBJETOS II

Lucas de Mira<sup>1</sup>, Mateo Motta<sup>2</sup>

Universidade da Região de Joinville(UNIVILLE)  
Joinville – SC – Brasil

lucas.mira@univille.br, mateo.motta@univille.br

## 1. Descrição do sistema

Com uma vasta coleção que inclui milhões de músicas e playlists cuidadosamente organizadas, o GigatonPlay oferece aos usuários a possibilidade de explorar facilmente uma grande variedade de estilos musicais, artistas e sons com apenas um toque. Seja para revisitar suas canções preferidas, criar a trilha sonora perfeita para o seu dia ou descobrir novos artistas, nosso aplicativo oferece uma experiência musical única. A interface é simples e intuitiva, com recomendações feitas sob medida para os gostos do usuário, além de permitir o download de músicas para ouvir offline, garantindo controle total sobre a sua biblioteca, em qualquer ambiente ou ocasião.

## 2. Requisitos Funcionais

### 1. Adicionar Artista:

- O sistema deve permitir que eu adicione um novo artista, informando o nome, nacionalidade e gênero musical.

### 2. Listar Artistas:

- Eu quero visualizar todos os artistas cadastrados na minha biblioteca.

### 3. Adicionar Álbum:

- O sistema deve permitir que eu adicione um álbum, informando o nome e selecionando o artista associado.

### 4. Adicionar Música:

- Eu quero poder adicionar uma nova música a um álbum, informando o nome da música, duração, artista e álbum.

### 5. Listar Músicas:

- O sistema deve permitir que eu visualize todas as músicas disponíveis na minha biblioteca.

### 6. Editar Música:

- Se necessário, eu quero editar os detalhes de uma música existente, como o nome ou duração.

### 7. Excluir Música:

- O sistema deve permitir que eu exclua músicas da minha coleção.

## 8. Validação de Dados:

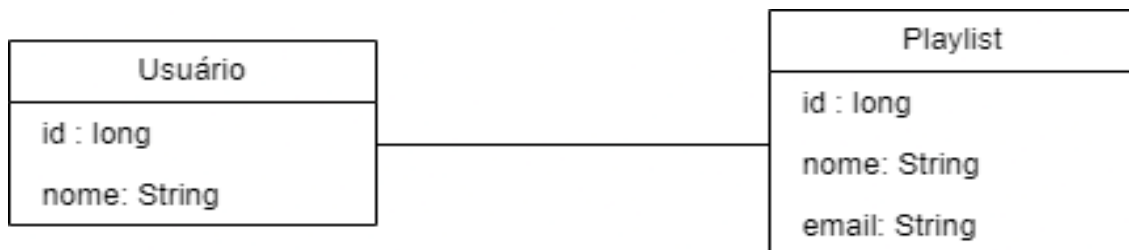
- O sistema deve garantir que todos os dados inseridos sejam válidos, como nome do artista, álbum e música.

## 9. Feedback ao Usuário:

- Após cada ação (adição, edição, exclusão), o sistema deve fornecer um feedback claro confirmando o sucesso da operação.

### 2.1. História de Usuário

Como usuário do GigatonPlay, quero organizar minha coleção de músicas e artistas para construir uma biblioteca musical rica e diversificada. Quando adiciono um novo artista, quero informar seu nome, nacionalidade e estilo musical. Além disso, gostaria de criar álbuns para cada um desses artistas, facilitando a organização das minhas músicas. Ao inserir uma faixa em um álbum, é necessário que o sistema permita alterar o nome, a duração da música e selecionar o artista e álbum correspondente. Quero também poder visualizar todos os artistas, álbuns e músicas registrados, e ter a opção de editar ou remover qualquer item da minha coleção. O sistema deverá validar as informações inseridas, assegurando que não ocorram erros, e eu espero receber uma confirmação clara após cada ação, para garantir que minha biblioteca está sempre atualizada e organizada



## 3. Codificação

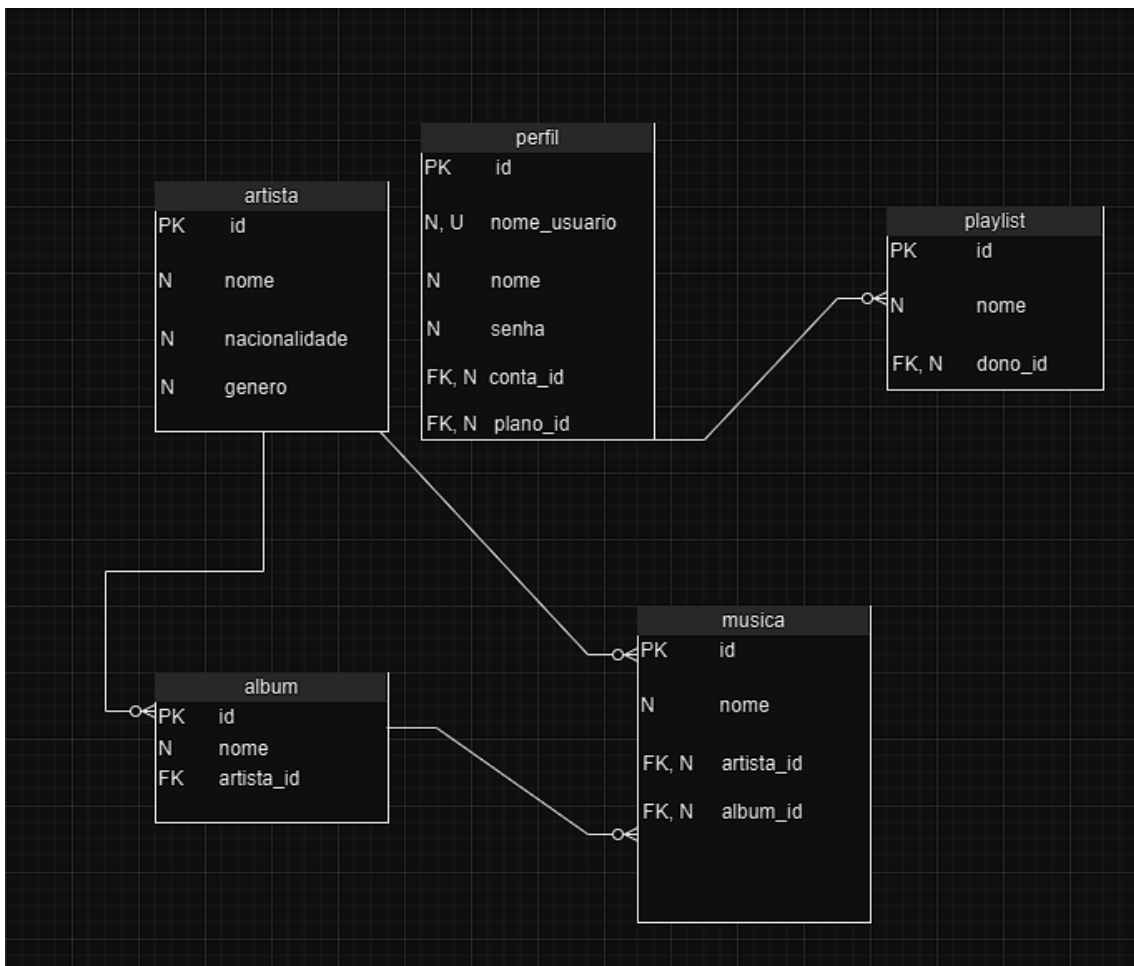
O diagrama UML descreve três entidades principais: Playlist, Música e uma concepção abstrata de álbum. A entidade Playlist possui um identificador único (id) e um nome (nome), podendo conter múltiplas músicas, configurando uma relação "um para muitos" com a entidade Música. Por sua vez, a entidade Música tem atributos como id, nome e duração, e pode estar associada a diversas playlists. Além disso, o diagrama mostra que um álbum é composto por várias músicas, refletindo outra relação "um para muitos".

Em uma implementação em Java, essa estrutura pode ser desenvolvida com a classe Playlist contendo uma lista de objetos da classe Música, enquanto a classe Música pode ter uma ligação opcional com a entidade Álbum. Assim, o diagrama destaca como as playlists são formadas por coleções de músicas e como essas músicas também podem estar vinculadas a álbuns, espelhando o funcionamento de aplicativos de música atuais.

#### 4. Banco de Dados

O diagrama UML das entidades Conta, Álbum e Playlist ilustra a estrutura e as relações entre elas no banco de dados. A entidade Conta é responsável pela gestão de usuários, identificados por um id único e um campo email, que deve ser exclusivo. A entidade Álbum representa os discos musicais e estabelece uma relação "muitos para um" com a entidade Artista, onde um artista pode possuir diversos álbuns, mas cada álbum está relacionado a um único artista.

A entidade Playlist contém listas de músicas e perfis de ouvintes, configuradas por meio de relações "muitos para muitos", permitindo que uma playlist tenha várias músicas e esteja associada a diversos perfis de usuários. Além disso, cada playlist tem um perfil de proprietário definido por uma relação "muitos para um".



## **5. Conclusão**

O aplicativo de música oferece uma experiência altamente personalizada, permitindo que os usuários explorem, organizem e aproveitem suas músicas favoritas de forma ilimitada. As histórias de usuários demonstram a importância de uma interface intuitiva e flexível, que facilita a criação e o gerenciamento de playlists, proporcionando ao usuário controle total sobre sua experiência musical.

O diagrama de classes reflete essa estrutura, destacando as interações entre as entidades Playlist, Música e Usuário, e como essas conexões garantem uma experiência fluida e personalizada, adaptada ao gosto de cada pessoa. Assim, o sistema transforma a experiência de ouvir música em algo único e acessível, onde quer que o usuário esteja.