

# Sistema de Música Harmoniq

Carlos Silva, Lorenzo Poncio

Engenharia de Software/Sistemas de Informação  
Universidade da Região de Joinville (UNIVILLE) – Joinville, SC – Brasil

Carlos.silva23@univille.br, lorenzo.poncio@univille.br

## 1. Introdução

Este trabalho visa a modelagem de um sistema de música com foco na criação de playlists, organização de músicas e interação entre usuários. A partir das necessidades levantadas pelos mesmos, foram desenvolvidas três histórias, representadas por diagramas de classes e de entidade-relacionamento. Cada uma das histórias aborda funcionalidades essenciais para a experiência do usuário na plataforma, como a busca por músicas e artistas, a criação de playlists e a possibilidade de compartilhar essas listas com outros perfis. O objetivo principal do sistema é melhorar a descoberta musical e permitir uma experiência social interativa entre os usuários da plataforma. A modelagem proposta atende a diversos relacionamentos e requisitos técnicos, como a associação de músicas com álbuns e artistas, perfis de usuários vinculados a planos de assinatura e a possibilidade de interações sociais por meio do compartilhamento de playlists.

## 2. Requisitos Funcionais

Os requisitos funcionais do Sistema Harmoniq serão apresentados em forma de história de usuário.

### 2.1. História de Usuário 01

Como um usuário do Sistema de Música, desejo criar playlists personalizadas, adicionando músicas de diferentes álbuns e artistas, organizando-as conforme minhas preferências e compartilhando com outros usuários da plataforma. Cada álbum no sistema deverá ser associado a um único artista, contendo seu nome, nacionalidade e gênero musical, facilitando a descoberta de novas músicas.

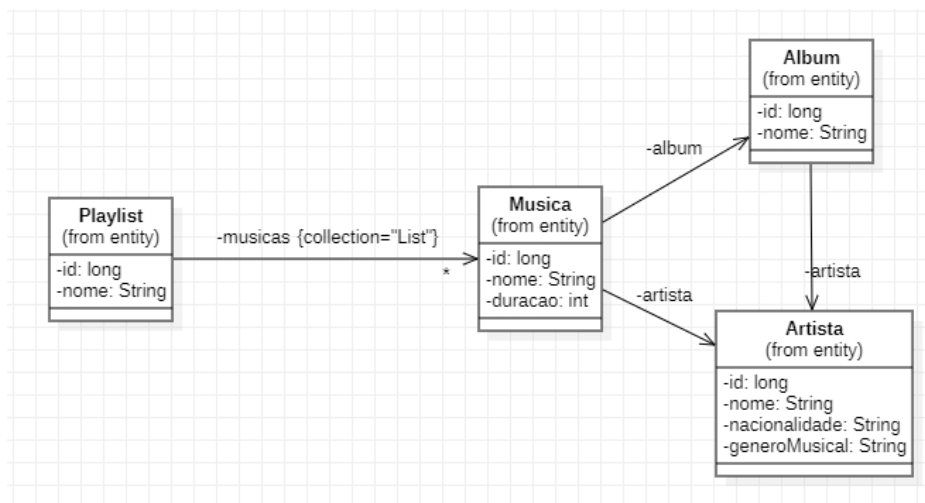


Figura 1. Diagrama de classe das entidades da História de Usuário 01.

A Figura 02 o diagrama de classe da história de usuário 01. A tabela playlist\_musicas contém as chaves estrangeiras da tabela Playlist e Musica. Mesmo não estando no diagrama de classes, essa tabela surgiu devido ao relacionamento do tipo muitos para muitos.

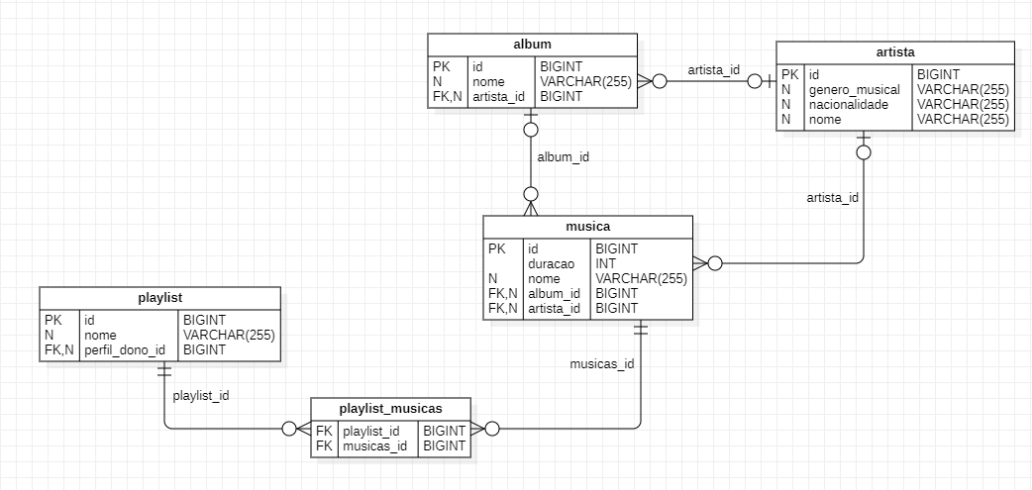


Figura 2. Modelo Entidade Relacionamento da História de Usuário 01.

2.2. História de Usuário 02

Cada usuário poderá ter uma conta, identificada por um e-mail único. Esses perfis, por sua vez, estão vinculados a planos de assinatura específicos e podem criar e compartilhar playlists, além de seguir outros perfis para descobrir músicas e artistas. As playlists permitem que os usuários organizem e compartilhem suas músicas favoritas com amigos e familiares, melhorando a experiência de descoberta musical e tornando-a mais social e interativa.

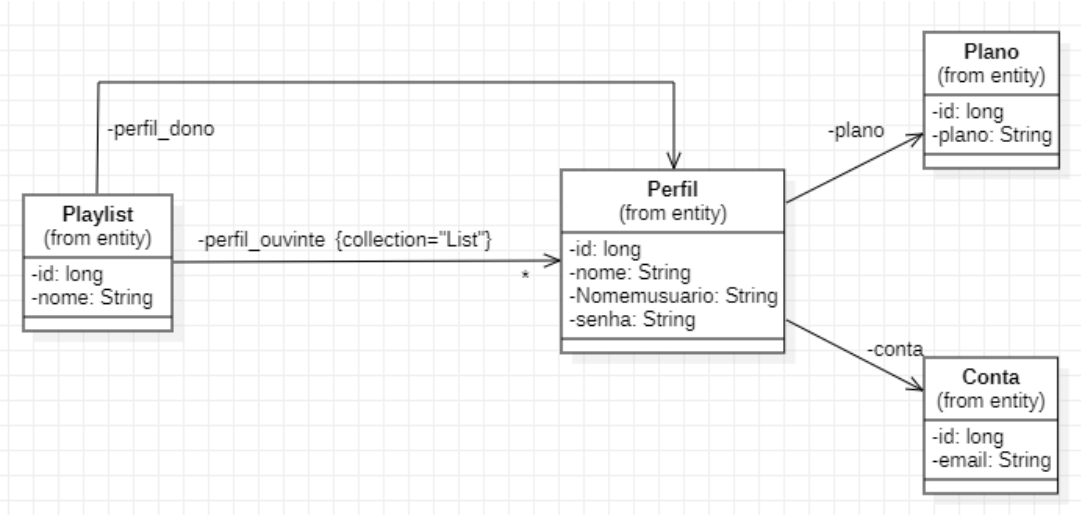


Figura 3. Diagrama de classe das entidades da História de Usuário 02.

A Figura 04 o diagrama de classe da história de usuário 02. A tabela playlist\_perfil\_ouvinte contém as chaves estrangeiras da tabela Playlist e Perfil. Mesmo não estando no diagrama de classes, essa tabela surgiu devido ao relacionamento do tipo muitos para muitos, onde um perfil pode ter acesso a várias playlists e uma playlist pode ter vários perfis ouvintes.

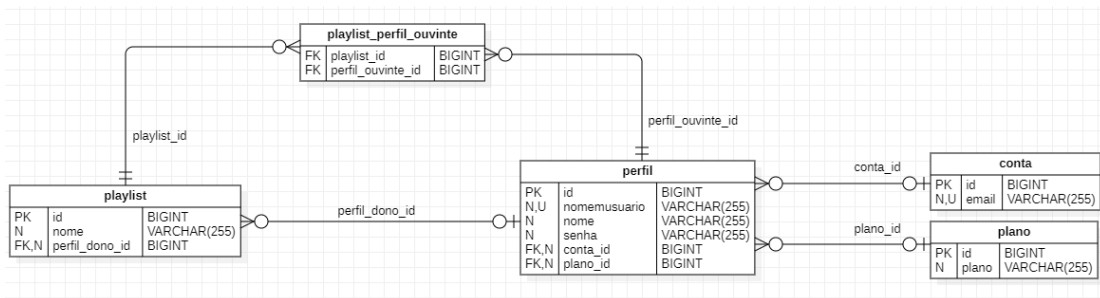


Figura 4. Modelo Entidade Relacionamento da História de Usuário 01.

### 2.3. História de Usuário 03

A busca por músicas será baseada em informações como o nome da música, o álbum em que ela está e o artista que a produziu. Já na busca por artistas, o usuário poderá encontrar detalhes como o nome, a nacionalidade e o gênero musical, facilitando a exploração de novos estilos e artistas conforme suas preferências.

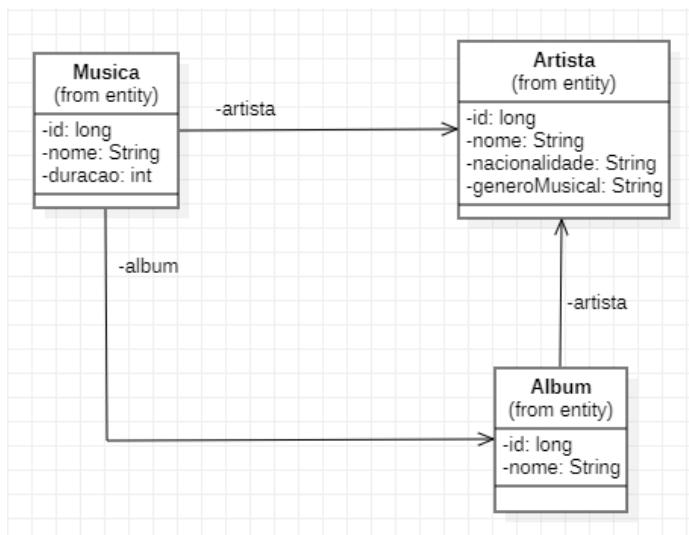


Figura 5. Diagrama de classe das entidades da História de Usuário 03.

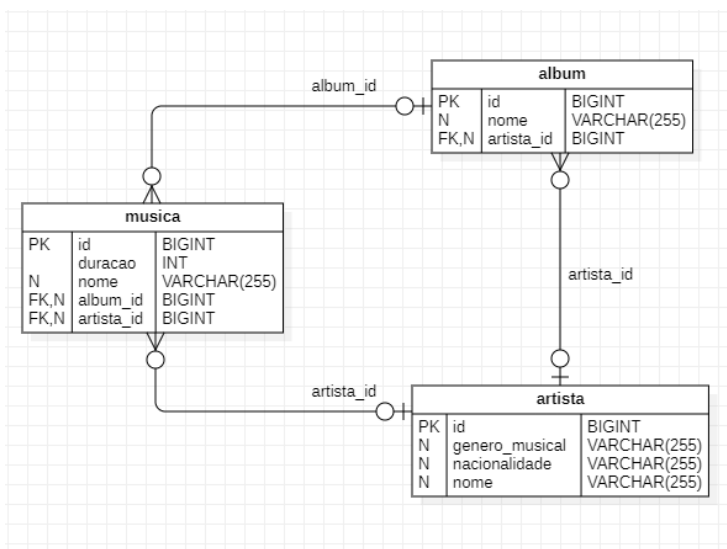


Figura 6. Modelo Entidade Relacionamento da História de Usuário 03.

### 3. Codificação

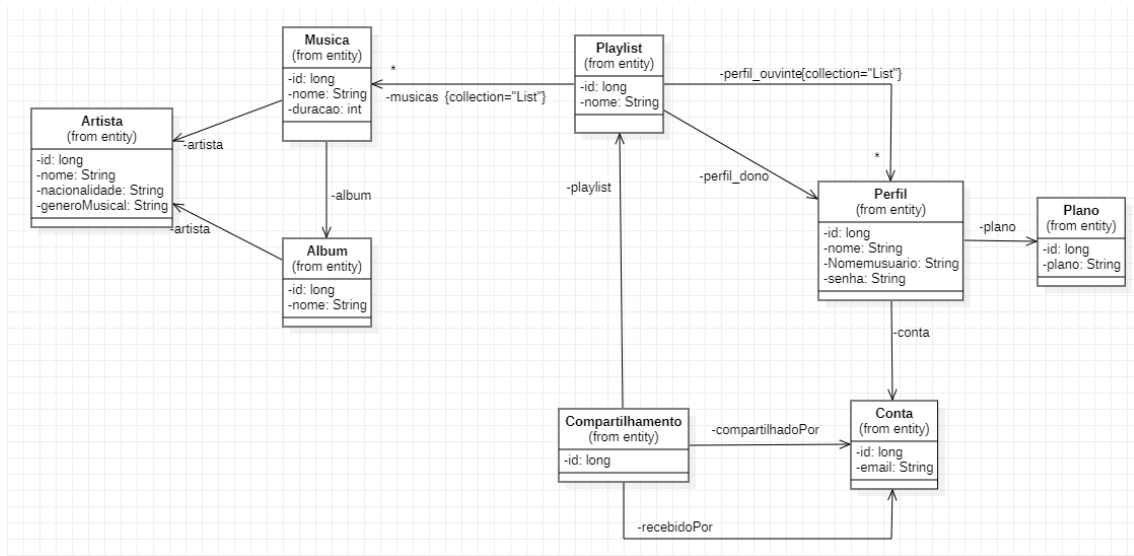


Figura 7. Diagrama de classe do Sistema de música Harmoniq

#### 3.1. Entidade Playlist

A entidade Playlist representa a lista de músicas que o usuário pode criar para salvar suas músicas favoritas.

```
public class Playlist {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private long id;
    private String nome;

    @ManyToMany
    private List<Musica> musicas;
    @ManyToMany
    private List<Perfil> perfil_ouvinte;
    @ManyToOne
    private Perfil perfil_dono;
}
```

Figura 8. Código da entidade Playlist

#### 3.2. Entidade Perfil

A entidade Perfil representa as informações de um usuário ou conta em um sistema, incluindo detalhes específicos que podem ser associados a esse usuário.

```

public class Perfil {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private long id;

    @ManyToOne
    private Conta conta;

    private String nome;

    @ManyToOne
    private Plano plano;

    @Column (unique = true) //nome deve ser único
    private String Nomemusuario;

    private String senha;
}

```

Figura 9. Código da entidade Perfil

### 3.3. Entidade Musica

A entidade Musica representa faixas na qual o usuário pode escutar, baixar ou adicionar a playlists.

```

public class Musica {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private long id;
    private String nome;
    private int duracao;

    // Propria Classe / Atributo
    @ManyToOne
    private Artista artista;
}

```

```

    @ManyToOne
    private Album album;
}

```

Figura 10. Código da entidade Musica

### 3.4. Entidade Compartilhamento

A entidade compartilhamento representa a ação de compartilhar músicas, álbuns ou playlists entre usuários da plataforma.

```

public class Compartilhamento {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private long id;

    @ManyToOne
    private Conta compartilhadoPor;

    @ManyToOne
    private Conta recebidoPor;

    @ManyToOne
    private Playlist playlist;

}

```

Figura 11. Código da entidade Compartilhamento

## 4. Banco de dados

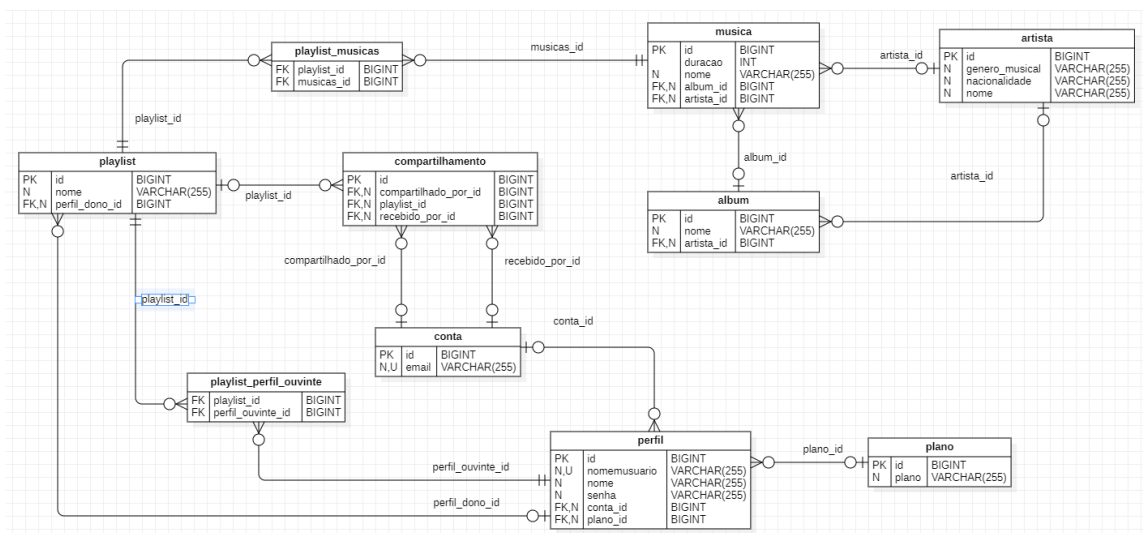


Figura 12. Modelo Entidade Relacionamento do Sistema de Música Harmoniq

#### **4. Conclusão**

O sistema modelado atende às principais necessidades levantadas nas histórias de usuários, fornecendo uma estrutura robusta para o gerenciamento de playlists e para a interação social entre os usuários da plataforma. A modelagem ER proposta possibilita relacionamentos muitos-para-muitos, como visto nas tabelas "playlist\_musicas" e "playlist\_perfil\_ouvinte", o que garante flexibilidade na associação de músicas com playlists e perfis de ouvintes. Além disso, a busca otimizada por músicas e artistas baseada em atributos como nome, nacionalidade e gênero musical proporciona uma melhor experiência ao usuário. Assim, a modelagem apresentada garante a escalabilidade e a usabilidade da plataforma para futuros desenvolvimentos.