

Sistema de Música XPO

Luca Vinicius Mesadri, Aramis Otero Fernandes

Engenharia de Software/Sistemas de Informação
Universidade da Região de Joinville (UNIVILLE) – Joinville, SC – Brazil
luca.mesadri@univille.br, aramis.fernandes@univille.br

1. Introdução

É um sistema de música online que permite ao usuário ouvir músicas, criar playlists, marcar músicas favoritas, explorar gêneros e artistas, e muito mais.

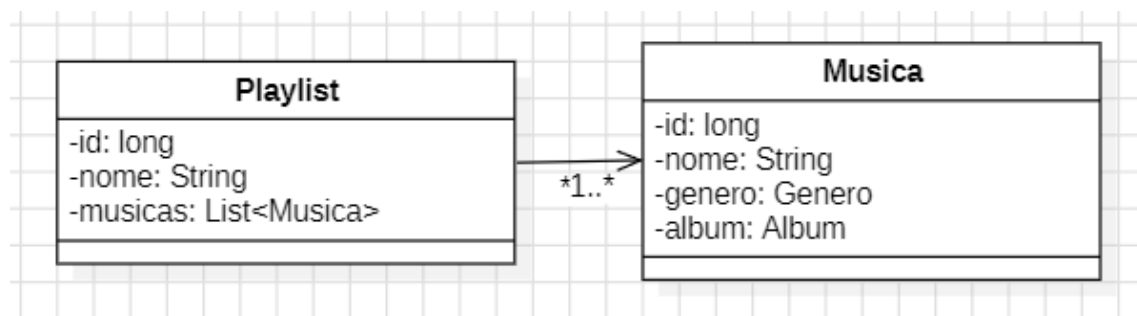
2. Requisitos Funcionais

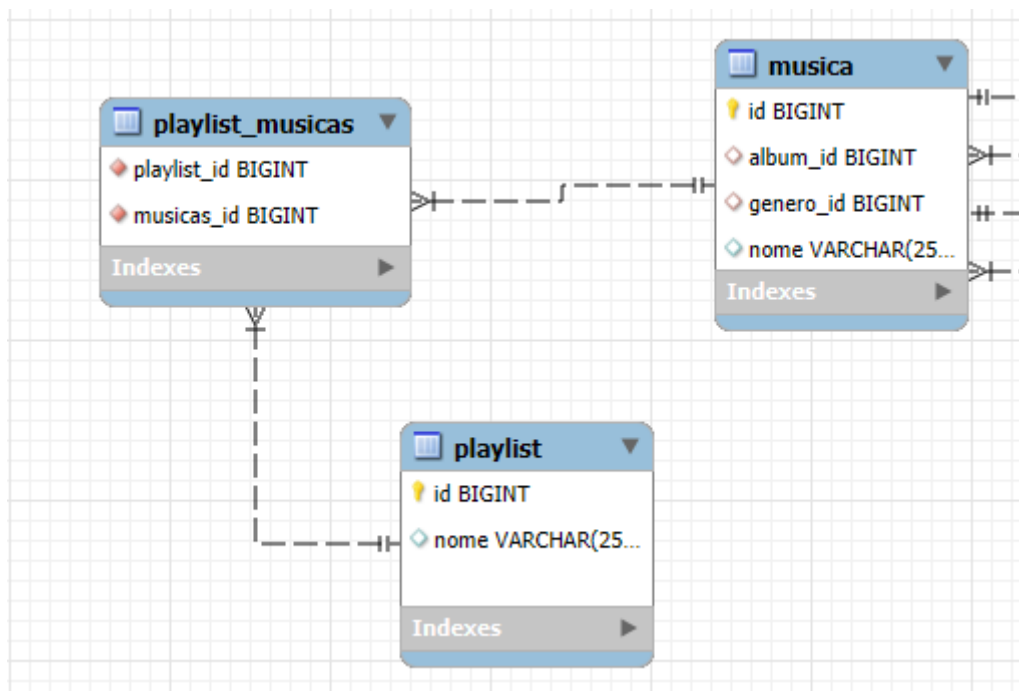
Os requisitos funcionais são apresentados como histórias de usuário, descrevendo as funcionalidades do sistema do ponto de vista do usuário final.

2.1. História de Usuário 01

Cenário: Personalização de playlists por Lucas

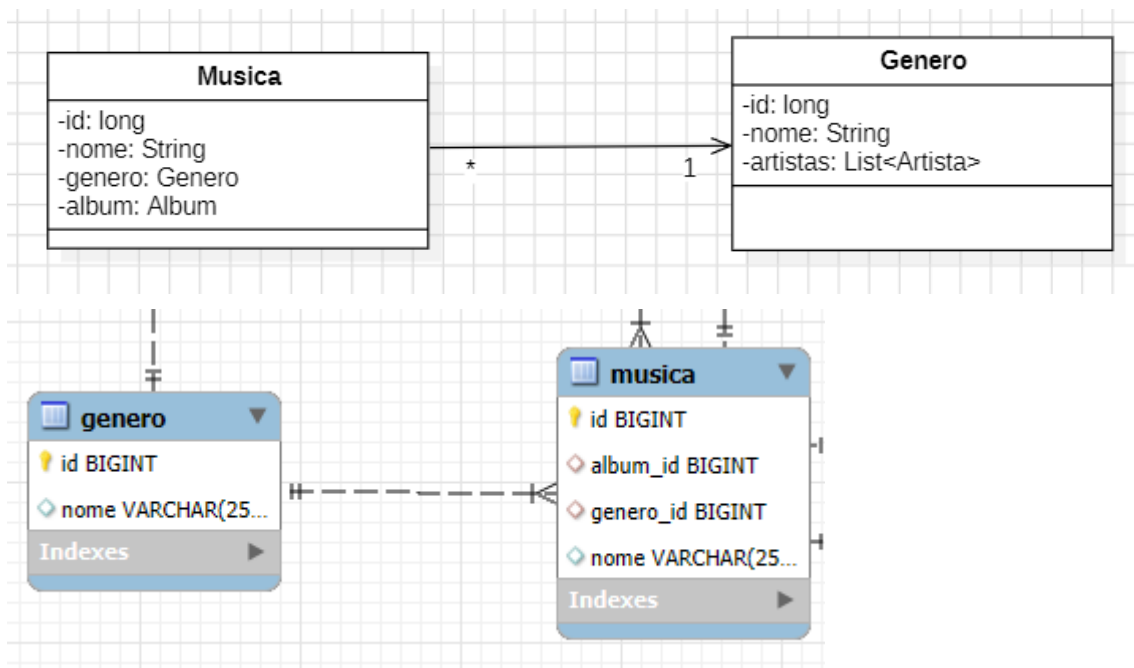
Jonas sempre gostou de organizar suas músicas em playlists temáticas. Ele acessa o sistema de música para criar novas playlists, adicionar suas músicas favoritas e organizá-las de acordo com seus gêneros preferidos. A funcionalidade de playlists permite que Lucas adicione, remova ou reorganize as músicas de acordo com suas preferências.





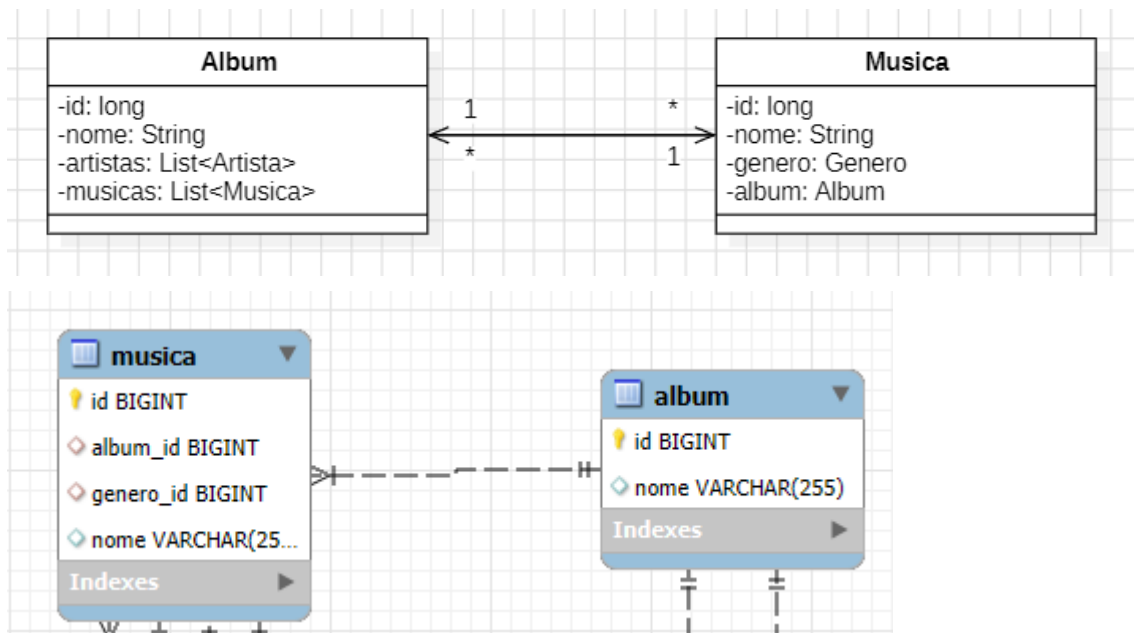
2.2. História de Usuário 02

Jonas gosta de descobrir novos estilos musicais. Ao explorar a seção de gêneros, ele pode visualizar todas as músicas categorizadas por seu gênero. Isso facilita a descoberta de novos artistas e músicas que ele ainda não conhecia. O sistema permite que ele filtre as músicas por gênero e adicione suas favoritas às suas playlists.



2.3. História de Usuário 03

Jonas, apaixonado por álbuns completos, acessa o sistema para visualizar todos os álbuns disponíveis. Ele encontra o seu álbum favorito, que contém várias músicas. Com o relacionamento estabelecido entre as entidades Álbum e Música, Lucas consegue ver todas as músicas que fazem parte de cada álbum. Ele pode escolher ouvir o álbum completo ou selecionar faixas específicas para adicionar às suas playlists.

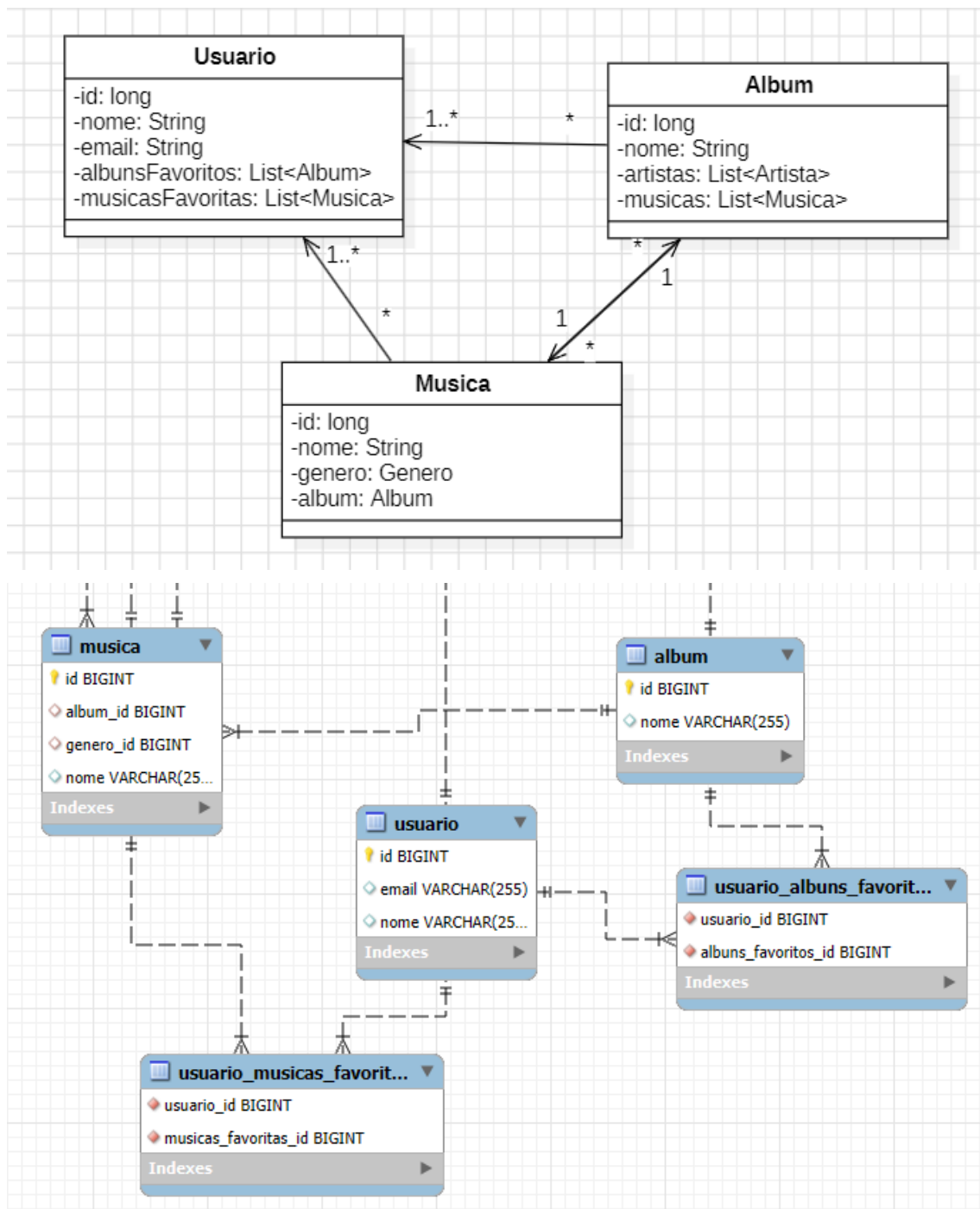


2.4. História de Usuário 04

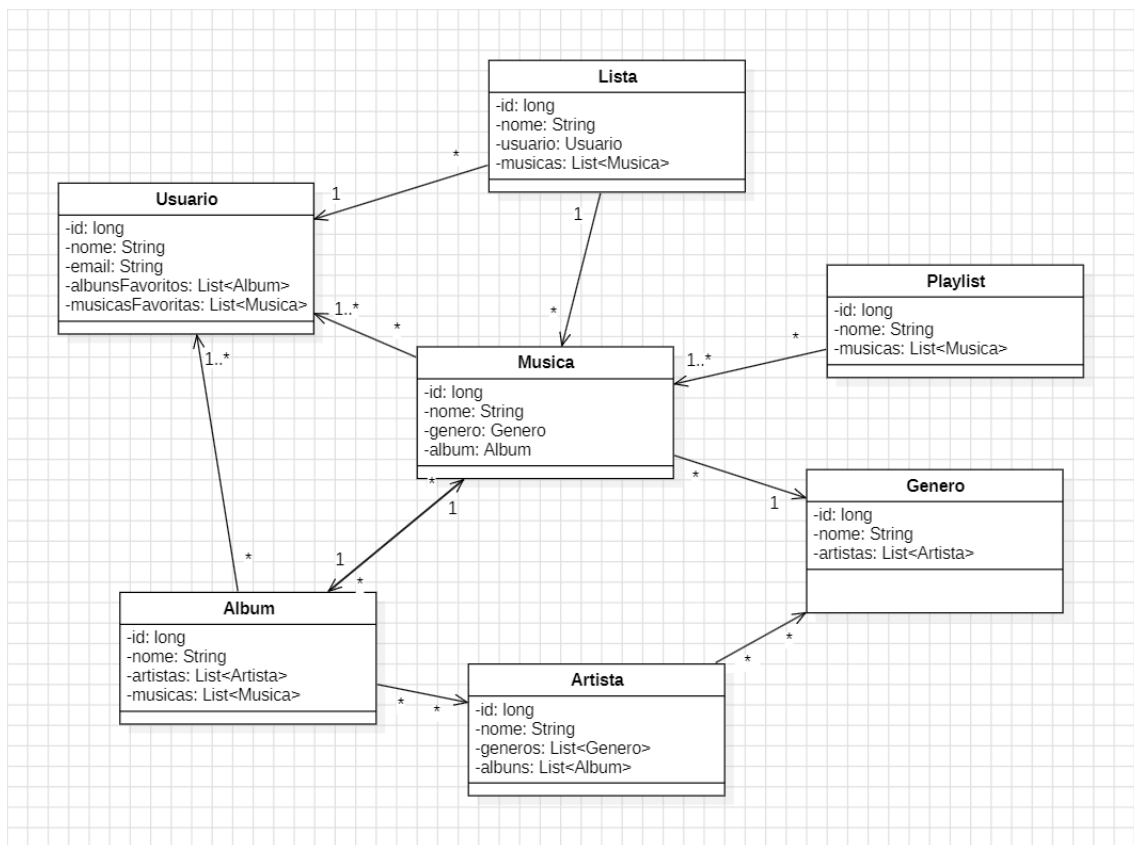
Jonas tem várias músicas e álbuns que ele considera suas favoritas. Com a funcionalidade de favoritos, ele pode marcar suas músicas e álbuns preferidos. A relação entre o usuário e as músicas/álbuns permite que Lucas organize suas músicas favoritas em uma seção especial, que ele acessa com frequência.

2.5. História de Usuário 05

Jonas tem uma grande admiração por diversos artistas e gosta de explorar suas discografias. Através do sistema, ele consegue visualizar todos os álbuns e músicas de um artista específico. Ao acessar o perfil de um artista que ele curte, o sistema exibe todos os álbuns lançados por aquele artista, permitindo que Lucas explore mais a fundo suas criações. Ele também pode adicionar músicas ou álbuns inteiros de seus artistas favoritos às suas playlists.



3. Codificação



Entidade Usuario:

```

public class Usuario {

    @ManyToMany
    private List<Album> albunsFavoritos;

    @ManyToMany
    private List<Musica> musicasFavoritas;
}
  
```

Entidade Playlist:

```

public class Playlist {

    @ManyToMany
    private List<Musica> musicas;

}

public class Musica {
  
```

```
@ManyToOne
private Album album;

@ManyToOne
private Genero genero;
```

Entidade Lista:

```
public class Lista {
```

```
@ManyToOne
@JoinColumn(name = "usuario_id")
private Usuario usuario;

@ManyToMany
@JoinTable(name = "lista_musica",
    joinColumns = @JoinColumn(name = "lista_id"),
    inverseJoinColumns = @JoinColumn(name = "musica_id"))
private List<Musica> musicas;
```

Entidade Gênero:

```
public class Genero {
```

```
@ManyToMany(mappedBy = "generos")
private List<Artista> artistas;
```

Entidade Artista:

```
public class Artista {
```

```

@ManyToMany
@JoinTable(name = "artista_genero",
    joinColumns = @JoinColumn(name = "artista_id"),
    inverseJoinColumns = @JoinColumn(name = "genero_id"))
private List<Genero> generos;

@ManyToMany(mappedBy = "artistas")
private List<Album> albuns;

```

Entidade Álbum:

```

public class Album {

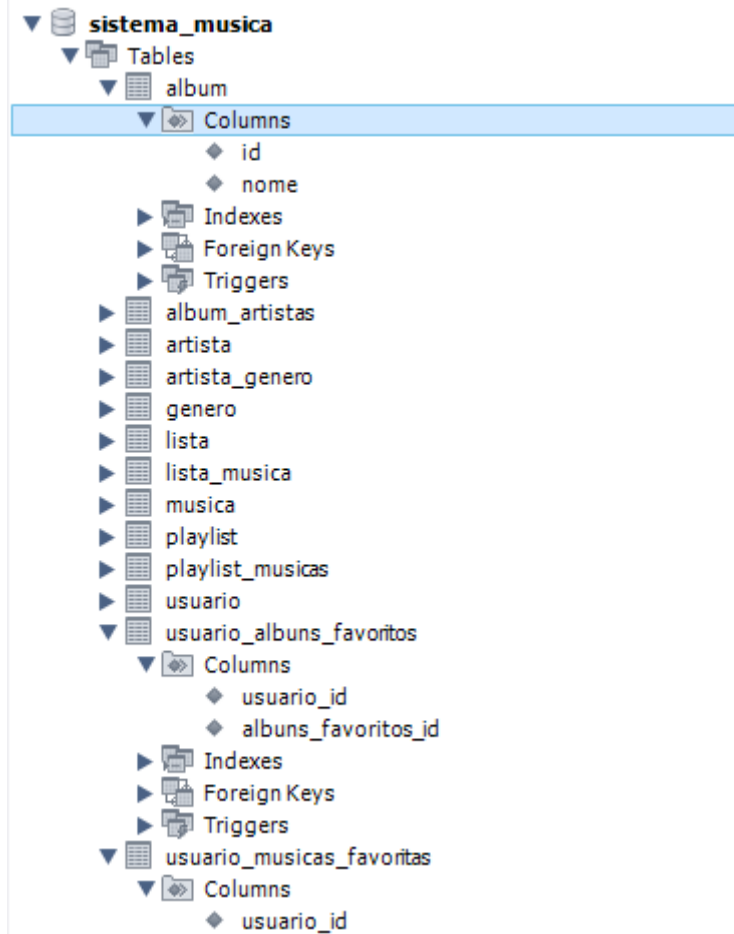
    @ManyToMany
    private List<Artista> artistas;

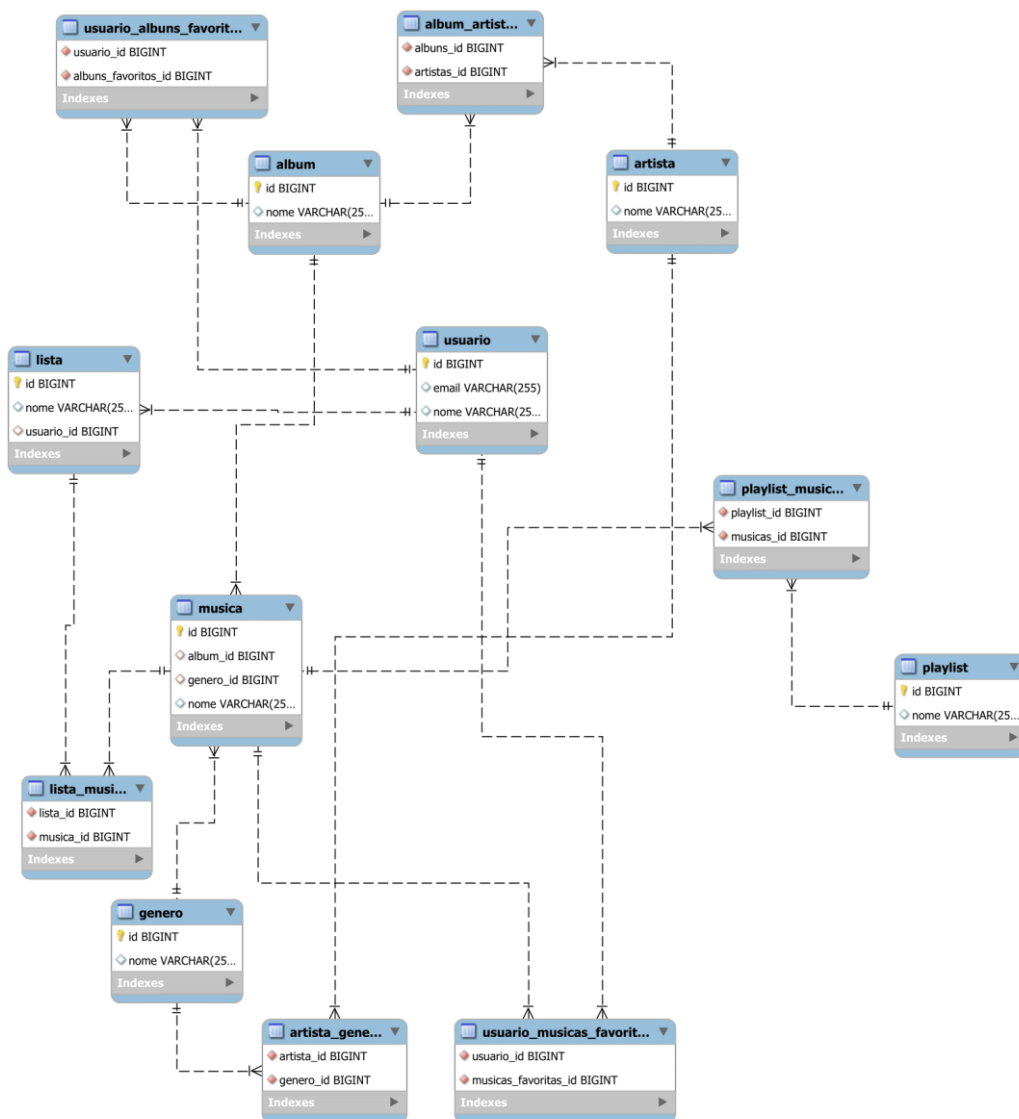
    @OneToMany(mappedBy = "album")
    private List<Musica> musicas;

```

4. Banco de dados

Aqui algumas das tabelas abertas (apenas algumas por motivo de falta de espaço):





4. Conclusão

O Sistema de Música foi projetado para fornecer uma experiência personalizada aos usuários, permitindo a criação e gerenciamento de playlists e a marcação de músicas e álbuns favoritos. O uso do framework Spring e do banco de dados MySQL garante um desenvolvimento eficiente e uma integração robusta entre as entidades desenvolvidas no sistema.