

# Sistema de Streaming de Músicas XPO

Carlos Eduardo Tomasi, Gustavo Henrique Antonius

Engenharia de Software

Universidade da Região de Joinville (UNIVILLE) – Joinville, SC – Brazil

carlos.tomasi@univille.br, gustavo.antonius@univille.br

## 1. Introdução

O sistema será uma ferramenta de streaming para que os usuários possam escutar músicas de forma simplificada, ele terá funcionalidades como pesquisar músicas, montar sua playlist e adicionar os artistas com seus álbuns e músicas.

## 2. Requisitos Funcionais

### 2.1 - Adicionar música a playlist (usuário) Muito para Muitos

Como usuário gostaria de adicionar músicas a nas minhas playlist salvas, dessa formas ficaria mais fácil encontrar minhas músicas favoritas.

A Figura 01 é o diagrama de classe da história de usuário 01. A entidade Playlist vai conter uma lista de Music e representa uma associação do tipo Muito para Muitos, ou seja, uma playlist pode ter muitas músicas e uma música pode estar em músicas playlist. A entidade Playlist, terá os atributos: id e name, enquanto a entidade Music terá os atributos: id e name.

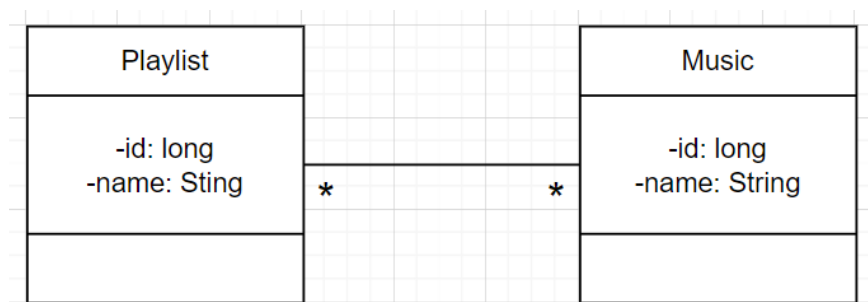


Figura 1. Diagrama de classe das entidades da História de Usuário 01.



Figura 2. Modelo Entidade Relacionamento do sistema para salvar música na playlist.

## 2.2 - Pesquisar Músicas (usuário)

Como usuário gostaria de poder pesquisar minhas músicas por nome, dessa forma ficaria mais fácil encontrar minhas músicas favoritas.

A Figura 03 é o diagrama de classe da história de usuário 02. A entidade Music vai ter os atributos: id e name.

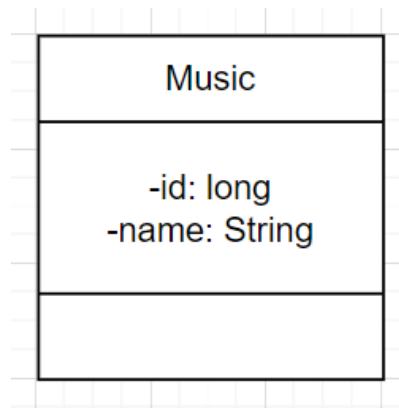


Figura 3. Diagrama de classe das entidades da História de Usuário 02.

## 2.3 - Cadastrar Artistas (adm) Tabela Artista só

Como administrador da streaming quero poder cadastrar os artistas e bandas, para que assim fiquem disponíveis na plataforma.

A Figura 04 é o diagrama de classe da história de usuário 03. A entidade Artist vai representar os novos artistas que serão cadastrados na plataforma. A entidade Artist, terá os atributos: id, name e nationality.

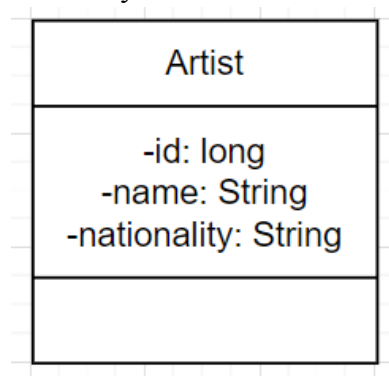


Figura 4. Diagrama de classe das entidades da História de Usuário 03.

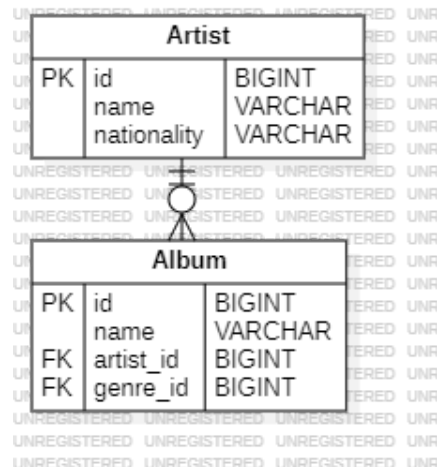


Figura 5. Modelo Entidade Relacionamento para cadastrar artistas.

## 2.4 - Cadastrar Álbum (adm) Muitos para um (gênero) Muitos para um (Artista)

Como administrador da streaming quero poder cadastrar os álbuns de músicas, para que assim fiquem disponíveis na plataforma.

A Figura 06 é o diagrama de classe da história de usuário 04. A entidade gênero vai conter uma lista de álbuns e representa uma associação do tipo muitos para um, ou seja, um gênero musical pode conter vários álbuns, porém um álbum irá conter apenas um gênero musical. A entidade Genre, terá os atributos: id e name, enquanto a entidade Album terá os atributos: id e name. Também temos a entidade artistas que vai conter uma lista de álbuns e representa uma associação do tipo Muitos para Um, ou seja, um artista pode conter vários álbuns, porém um álbum irá conter apenas um artista. A entidade Artist, terá os atributos: id, name e nationality, enquanto a entidade Album terá os atributos: id e name.

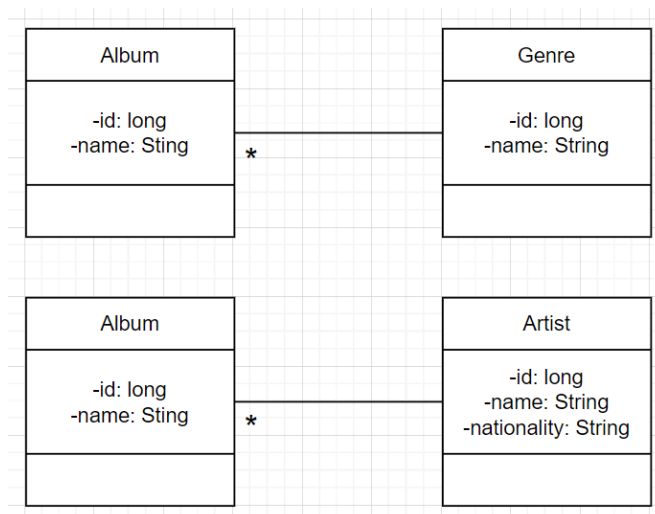
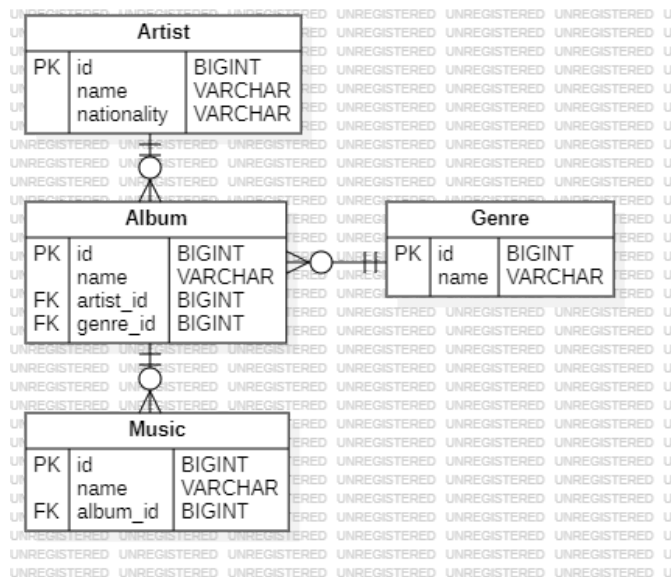


Figura 6. Diagrama de classe das entidades da História de Usuário 04.

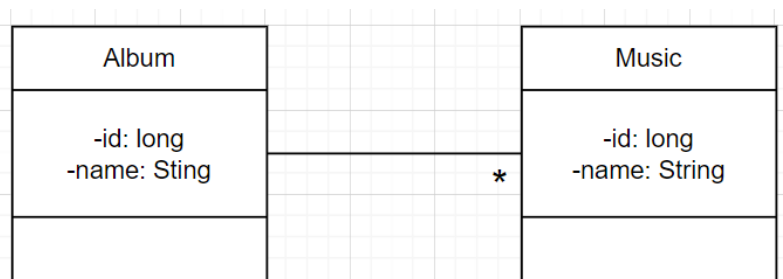


**Figura 7. Modelo Entidade Relacionamento para cadastrar álbuns.**

## 2.5 - Cadastrar Música (adm) Muitos para um (álbum)

Como administrador da streaming quero poder cadastrar as músicas, para que assim fiquem disponíveis na plataforma.

A Figura 08 é o diagrama de classe da história de usuário 05. A entidade álbum vai representar as novas músicas que serão cadastrados na plataforma e representa uma associação do tipo muito para um, ou seja, um álbum pode ter muitas músicas e uma música poderá estar apenas em um álbum. A entidade Album, terá os atributos: id e name, enquanto a entidade Music terá os atributos: id e name.



**Figura 8. Diagrama de classe das entidades da História de Usuário 05.**

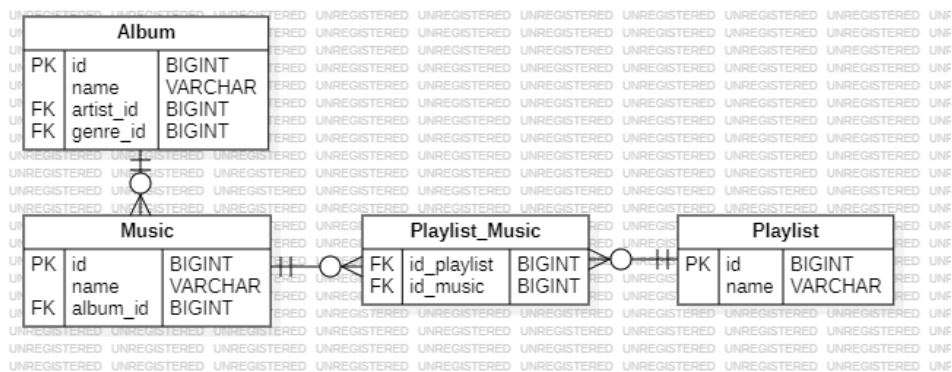


Figura 9. Modelo Entidade Relacionamento para cadastrar músicas.

## 2.6 - Cadastrar Gênero Musical (adm) Tabela Gênero

Como administrador da streaming quero poder separar as músicas cadastradas por gêneros musicais, dessa forma o usuário terá uma melhor experiência ao utilizar a plataforma.

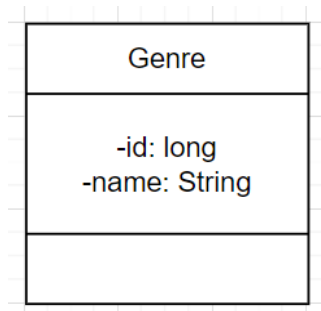


Figura 10. Diagrama de classe das entidades da História de Usuário 06.

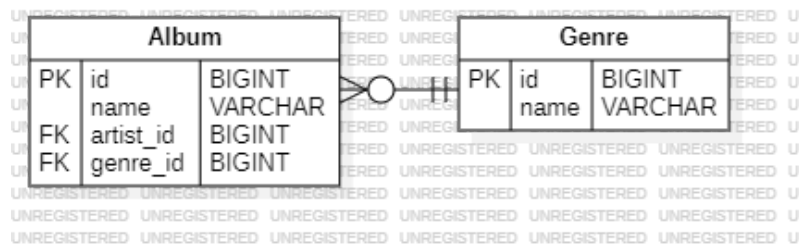


Figura 11. Modelo Entidade Relacionamento para cadastrar gênero musical.

2.7 - Criar playlist (usuário) Tabela Playlist

Como usuário, gostaria de poder criar minhas próprias playlists com minhas músicas favoritas.

A Figura 7 é o diagrama de classe da história de usuário 07. A entidade playlist vai representar os novas playlist criadas pelos usuários na plataforma. A entidade Playlist, terá os atributos: id e nome.

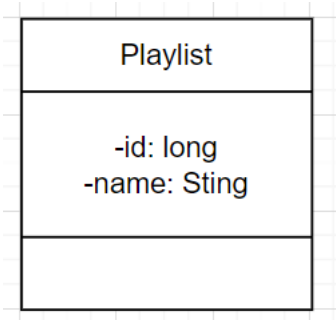


Figura 12. Diagrama de classe das entidades da História de Usuário 07.



Figura 13. Modelo Entidade Relacionamento para criar playlist.

3. Codificação

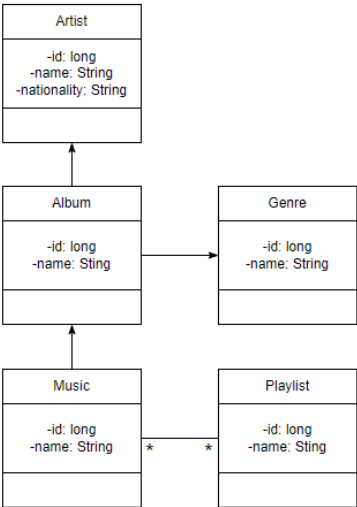


Figura 14. Diagrama de classes de todas as entidades.

### 3.1. Entidade XOP

#### Entidade Artista

```
@Entity
public class Artist {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private String name;

    private String nationality;
}
```

#### Entidade Gênero

```
@Entity
public class Genre {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private String name;
}
```

#### Entidade Album

```
@Entity
public class Album {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private String name;

    @ManyToOne
    @JoinColumn(name = "artist_id", referencedColumnName = "id")
    private Artist artist;

    @ManyToOne
    @JoinColumn(name = "genre_id", referencedColumnName = "id")
    private Genre genre;
}
```

## Entidade Música

```
@Entity
public class Music {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private String name;

    @ManyToOne
    @JoinColumn(name = "album_id")
    private Album album;
}
```

## Entidade Playlist

```
@Entity
public class Playlist {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private String name;

    @ManyToMany
    @JoinTable(
        name = "PlaylistMusic",
        joinColumns = @JoinColumn(name = "id_playlist"),
        inverseJoinColumns = @JoinColumn(name = "id_music")
    )
    private List<Music> musics;
}
```



#### 4. Banco de dados

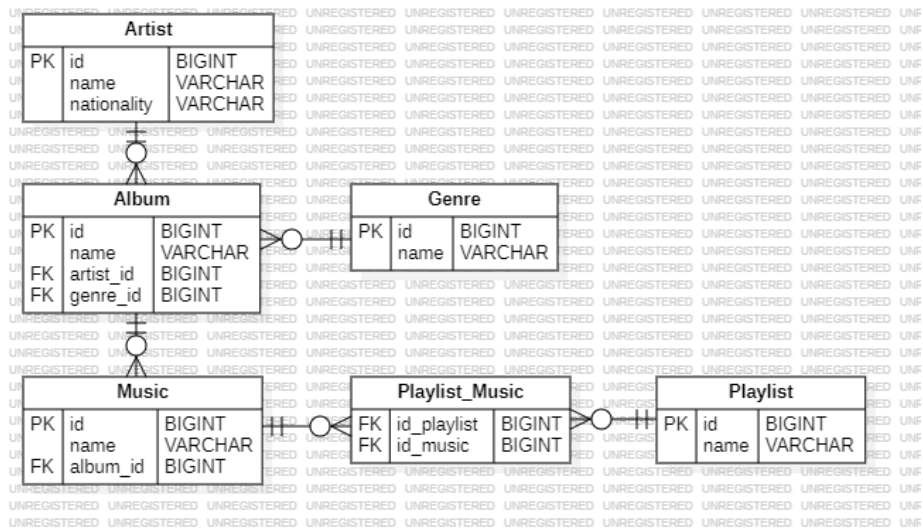


Figura 15. Modelo Entidade Relacionamento do Sistema de Música.

#### 5. Conclusão

Concluimos que o presente trabalho, uma plataforma de streaming de músicas nos ajudou a compreender melhor as interações entre duas entidades, a interação com o banco de dados, realização de diagramas a fim de nos tornar mais experientes na linguagem Java e suas ferramentas.

Utilizamos o Many-to-Many, criando uma tabela relacional entre músicas e playlist, além do Many-to-One para os demais relacionamentos.