

# Sistema de Música ‘Sonora’

Amanda Crispim, Alexandre de Mira

Engenharia de Software

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

amanda.crispim@univille.br, alexsandre.junior@univille.br

## 1. Introdução

Sonora é um aplicativo fictício de músicas, onde os usuários podem criar e compartilhar suas playlists com outros usuários, adicionar suas músicas favoritas e escolher ordenar suas playlists por ordem em que foram adicionadas ou ordem alfabética.

## 2. Requisitos Funcionais

Os requisitos funcionais do Sistema ‘Sonora’ serão apresentados em forma de história de usuário.

### 2.1. História de Usuário 01

Como músico que usa o sistema Sonora, eu quero poder adicionar minhas músicas, definir um título para elas e um gênero musical.

A Figura 01 representa o diagrama de classe da história de usuário 01. A entidade Autor vai conter uma lista de Música e representa uma associação do tipo One to Many, pois o Autor pode cadastrar várias músicas, mas uma música pertence a apenas um autor.

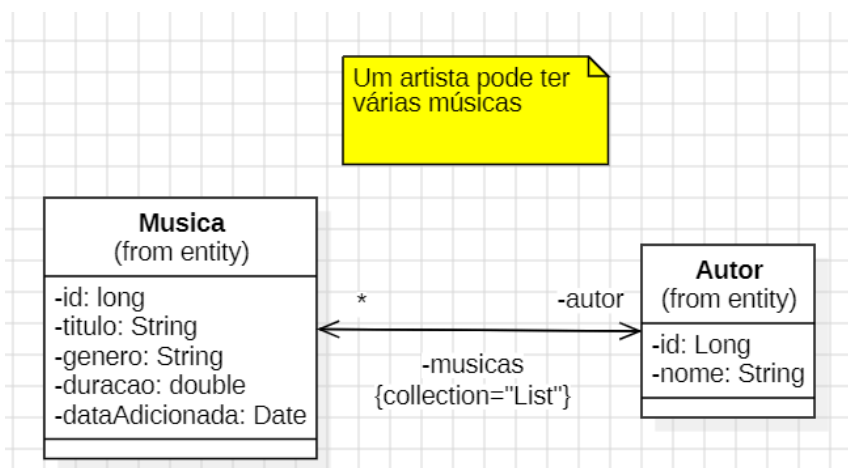
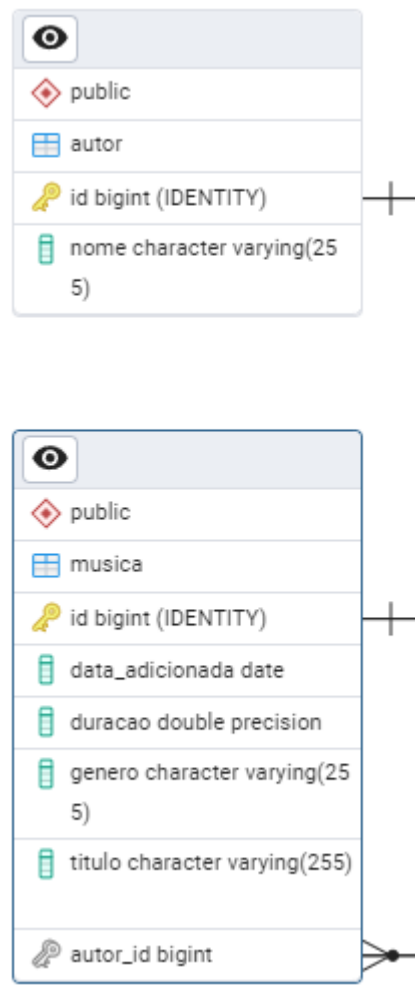


Figura 1. Diagrama de classe das entidades da História de Usuário 01.

A Figura 2 ilustra o diagrama de entidade-relacionamento das tabelas autor e musica. A tabela música contém uma chave estrangeira autor\_id que referencia a tabela autor, ilustrando a associação de várias músicas a um único autor.



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

## 2.2. História de Usuário 02

Como usuário do Sonora, quero poder criar minhas playlists e adicionar todas as músicas que eu quiser. Também quero poder organizar a minha playlist de maneira personalizada, seja por ordem alfabética ou pela ordem em que adicionei as músicas.

A Figura 3 mostra o diagrama de classe para a história do usuário 02, envolvendo as entidades usuário, usuario\_favoritos, e playlist. Os usuários podem criar múltiplas playlists e adicionar favoritos, com cada playlist contendo diversas músicas.

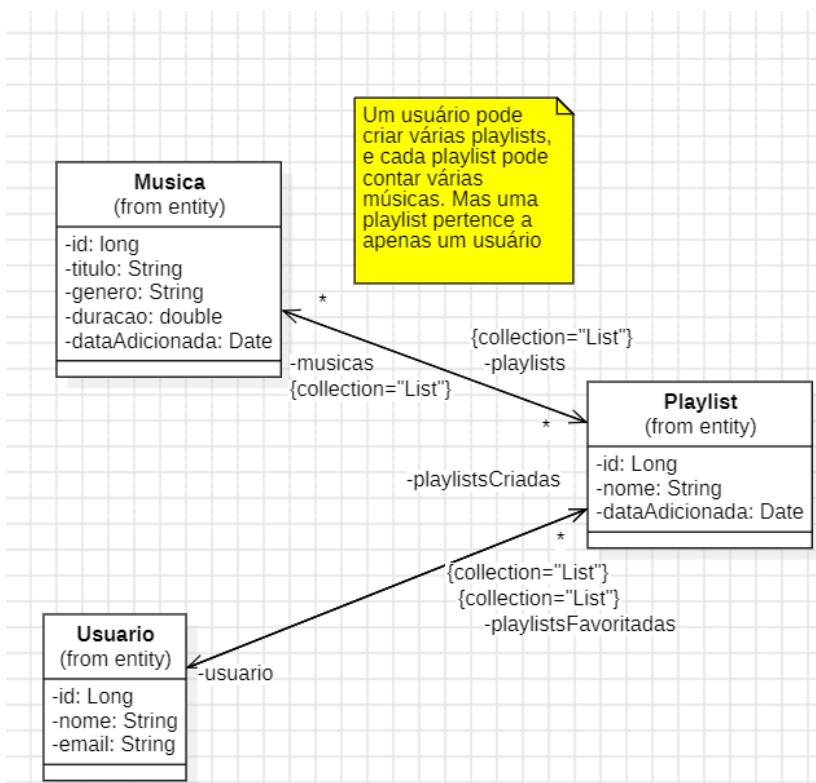


Figura 3. Diagrama de classe das entidades da História de Usuário 02.

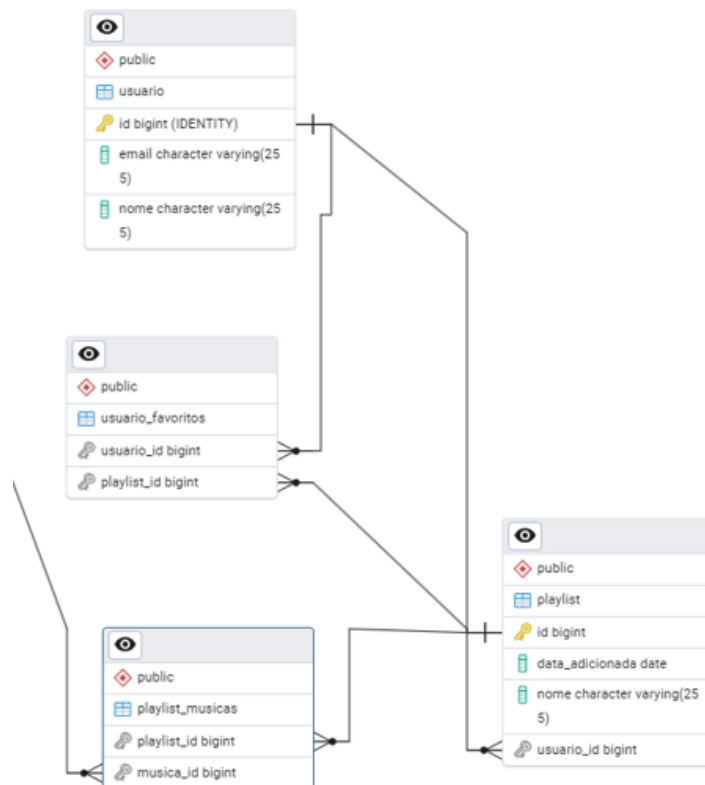


Figura 4. Modelo Entidade Relacionamento da História de Usuário 02.

### 3. Entidades

#### 3.1. Entidade Usuário

A entidade Usuário possui uma relação ‘One to Many’ com a entidade Playlist, ou seja, um usuário pode ter várias playlists, mas uma playlist pertence a apenas um usuário.

```
1: @Entity
public class Usuario {

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

    private String nome;
    private String email;

    @OneToMany(mappedBy = "usuario", cascade =
CascadeType.ALL)
    private List<Playlist> playlistsCriadas = new
ArrayList<>();

    @ManyToMany
    @JoinTable(
        name = "usuario_favoritos",
        joinColumns = @JoinColumn(name =
"usuario_id"),
        inverseJoinColumns = @JoinColumn(name =
"playlist_id")
    )
    private List<Playlist> playlistsFavoritadas = new
ArrayList<>();

}
```

Figura 5. Código da entidade Usuário

### 3.2. Entidade Playlist

A entidade Playlist possui uma relação de ‘Many to Many’ com a entidade Música, pois uma playlist pode conter várias músicas, ao mesmo tempo que uma música pode pertencer a mais de uma playlist. Do mesmo modo, uma playlist possui apenas um usuário dono, criando um relacionamento do tipo ‘Many to One’ com a entidade usuário, porém ela pode ser compartilhada, assim, ela pode entrar na lista de playlists de outro usuário.

```
1: @Data
@Entity
public class Playlist {

    @Id
    @GeneratedValue
    private Long id;

    private String nome;

    @Temporal(TemporalType.DATE)
    private Date dataAdicionada;

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

    @ManyToMany
    @JoinTable(
        name = "playlist_musicas",
        joinColumns = @JoinColumn(name =
"playlist_id"),
        inverseJoinColumns = @JoinColumn(name =
"musica_id")
    )
    private List<Musica> musicas = new ArrayList<>();

}
```

Figura 6. Código da entidade Playlist

### 3.2. Entidade Música

A entidade Música possui uma relação ‘Many to Many’ com a entidade Playlist, ou seja, uma música pode pertencer a várias playlists, e uma playlist pode possuir várias músicas. Porém, seu relacionamento com a entidade Autor é de ‘Many to One’, pois uma música pertence a apenas um autor, e um autor pode possuir várias músicas associadas a ele.

```
@Data
@Entity
public class Musica {

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

    private String titulo;

    private String genero;

    private double duracao;

    @ManyToOne
    @JoinColumn(name = "autor_id")
    private Autor autor;

    @Temporal(TemporalType.DATE)
    private Date dataAdicionada;

    @ManyToMany(mappedBy = "musicas")
    private List<Playlist> playlists = new ArrayList<>();

}
```

Figura 7. Código da entidade Música

### 3.4. Entidade Autor

A entidade Autor possui uma relação ‘One to Many’ com a entidade Música, pois um autor pode ter várias músicas cadastradas em seu nome, mas uma música pertence somente a um autor.

```
1: @Data
@Entity
public class Autor {

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

    private String nome;

    @OneToMany(mappedBy = "autor", cascade =
CascadeType.ALL, orphanRemoval = true)
    private List<Musica> musicas;
```

Figura 8. Código da entidade.

## 4. Banco de dados

Apresentar as tabelas e os relacionamentos. Apresentar o MER completo em forma de figura XY.

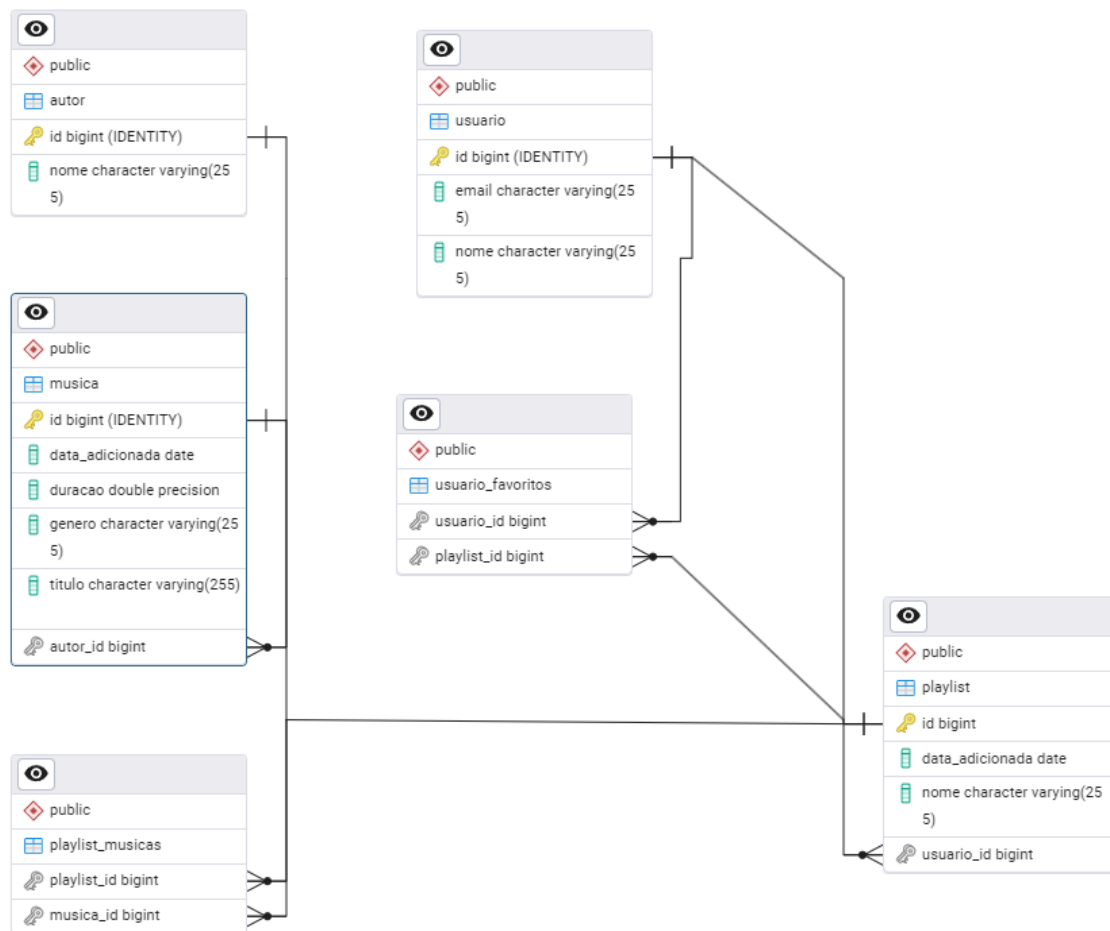


Figura 9. Modelo Entidade Relacionamento do sistema Sonora

## 4. Conclusão

Neste projeto, foram modeladas quatro principais entidades: **Usuário**, **Playlist**, **Autor**, e **Música**. A estrutura foi projetada para capturar as interações principais no aplicativo Sonora, com base em suas funcionalidades de criação, compartilhamento e personalização de playlists. A entidade **Usuário** desempenha um papel central, representando os indivíduos que interagem com o sistema. Cada usuário pode criar e gerenciar múltiplas playlists, o que é refletido pelo relacionamento de One to Many entre **Usuário** e **Playlist**. Além disso, os usuários podem favoritar playlists de outros usuários, o que permite a construção de uma relação de compartilhamento de conteúdo no sistema. As **playlists**, por sua vez, são compostas por uma coleção de músicas, criando um relacionamento Many to Many entre as entidades **Playlist** e **Música**. Isso reflete a capacidade de os usuários personalizarem suas playlists com qualquer conjunto de músicas disponíveis na plataforma, assim como a possibilidade de uma mesma música pertencer a múltiplas playlists. A entidade **Música** contém informações essenciais como título, duração e gênero, e é associada a um **Autor**, representando a



relação de One to Many entre **Autor** e **Música**. Cada autor pode ter diversas músicas associadas a ele, mas cada música pertence exclusivamente a um autor.