

Sistema de Música SPM

João Matheus, Lucas Bigas

Engenharia de Software

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

`joao.matheus@univille.br, lucaspadilha@univille.br`

1. Introdução

O sistema de música permite que os usuários façam streaming de músicas, criem e compartilhem playlists, sigam artistas, e descubram novas músicas com base em recomendações personalizadas. Ele deve suportar cadastro de usuários, artistas e músicas, além de fornecer a infraestrutura para manipulação de playlists e recomendações com base em interações do usuário.

2. Requisitos Funcionais:

2.1. História de Usuário 01: Cadastro de Usuário

Como usuário, eu quero me cadastrar no sistema, onde cada usuário tenha um identificador único, um nome de usuário, uma senha e um e-mail, para que eu possa acessar minhas músicas favoritas.

2.2. História de Usuário 02: Criação de Playlists

Como usuário, eu quero poder criar e gerenciar minhas playlists, de forma que cada playlist tenha um identificador único, um título para resumir seu conteúdo, e uma descrição opcional, para que eu possa organizar melhor minhas músicas e facilitar o acesso a elas. Várias playlist vão poder adicionar várias músicas nelas.

2.3 História de Usuário 03: Músicas Favoritas

Como usuário, quero marcar músicas como favoritas, para que eu possa criar uma lista personalizada de músicas que eu gosto, permitindo o acesso rápido e fácil a elas a partir do meu perfil, essas músicas vai ter um álbum e um artista. Vários usuários vai poder acessar essas músicas.

2.4 História de Usuário 04: Lista de Artistas Favoritos

Como usuário, quero adicionar artistas à minha lista de favoritos, para que eu possa acompanhar seus novos lançamentos e shows, e acessar suas músicas facilmente. Vários usuários vão ter uma lista de artistas favoritos nela.

3. Diagrama de Classe (UML)

O diagrama de classe mostrará as relações entre as principais entidades do sistema: usuário, playlist, artista, música, curtidas, álbum e artista favoritos. Ele deve representar:

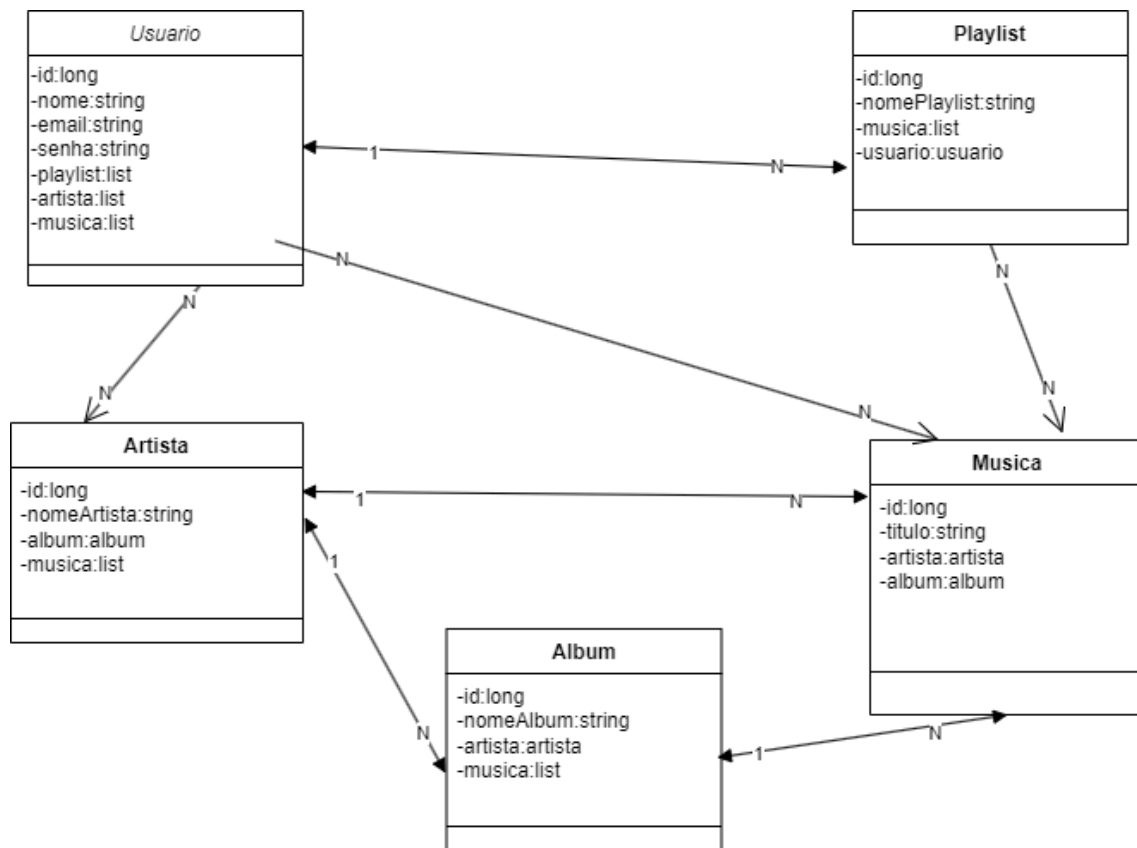


Figura do Diagrama de classe do SPM

3.1. Entidade SPM

A entidade do SMP representa as entidades de acordo com o diagrama, Foca nos relacionamentos de entidades e entre elas também.

```
1: @Data
2: @NoArgsConstructor
3: @Entity
4: public class Usuario {
5:     @Id
```

```

6:     @GeneratedValue(strategy = GenerationType.IDENTITY)
7:     private long id;
8:     private String nome;
9:     private String email;
10:    private String senha;
11:    @OneToMany(mappedBy = "usuario", cascade = CascadeType.ALL)
12:    private List<Playlist> playlists;
13:    @ManyToMany
14:    @JoinTable(
15:        name = "usuario_artista",
16:        joinColumns = @JoinColumn(name = "usuario_id"),
17:        inverseJoinColumns = @JoinColumn(name = "artista_id")
18:    )
19:    private List<Artista> artistas;
20:    @ManyToMany
21:    @JoinTable(
22:        name = "usuario_musica",
23:        joinColumns = @JoinColumn(name = "usuario_id"),
24:        inverseJoinColumns = @JoinColumn(name = "musica_id")
25:    )
26:    private List<Musica> musicas;
27: }

```

Figura 1. Código da entidade Usuario.

```

1: @Data
2: @NoArgsConstructor
3: @Entity
4: public class Playlist {
5:     @Id
6:     @GeneratedValue(strategy = GenerationType.IDENTITY)
7:     private Long id;
8:     private String nomePlaylist;

```

```

9:     @ManyToOne
10:     @JoinColumn(name = "usuario_id")
11:     private Usuario usuario;
12:     @ManyToMany
13:     @JoinTable(
14:         name = "playlist_musica",
15:         joinColumns = @JoinColumn(name = "playlist_id"),
16:         inverseJoinColumns = @JoinColumn(name = "musica_id")
17:     )
18:     private List<Musica> musicas;
19: }

```

Figura 2. Código da entidade Playlist.

```

1: @Data
2: @NoArgsConstructor
3: @Entity
4: public class Musica {
5:     @Id
6:     @GeneratedValue(strategy = GenerationType.IDENTITY)
7:     private Long id;
8:     private String titulo;
9:     @ManyToOne
10:     @JoinColumn(name = "artista_id")
11:     private Artista artista;
12:     @ManyToOne
13:     @JoinColumn(name = "album_id")
14:     private Album album;
15: }

```

Figura 3. Código da entidade Música.

```

1: @Data
2: @NoArgsConstructor

```

```
3:  @Entity
4:  public class Artista {
5:      @Id
6:      @GeneratedValue(strategy = GenerationType.IDENTITY)
7:      private Long id;
8:      private String nomeArtista;
9:      @OneToMany(mappedBy = "artista", cascade = CascadeType.ALL)
10:     private List<Album> albums;
11:     @OneToMany(mappedBy = "artista")
12:     private List<Musica> musicas;
13: }
```

Figura 4. Código da entidade Artista.

```
1:  @Data
2:  @NoArgsConstructor
3:  @Entity
4:  public class Album {
5:      @Id
6:      @GeneratedValue(strategy = GenerationType.IDENTITY)
7:      private Long id;
8:      private String nomeAlbum;
9:      @OneToMany(mappedBy = "album")
10:     private List<Musica> musicas;
11:     @ManyToOne
12:     @JoinColumn(name = "artista_id")
13:     private Artista artista;
14: }
```

Figura 5. Código da entidade Álbum.

4. Banco de dados

Aqui estão as tabelas do banco de dados em forma de diagrama tipo MER.

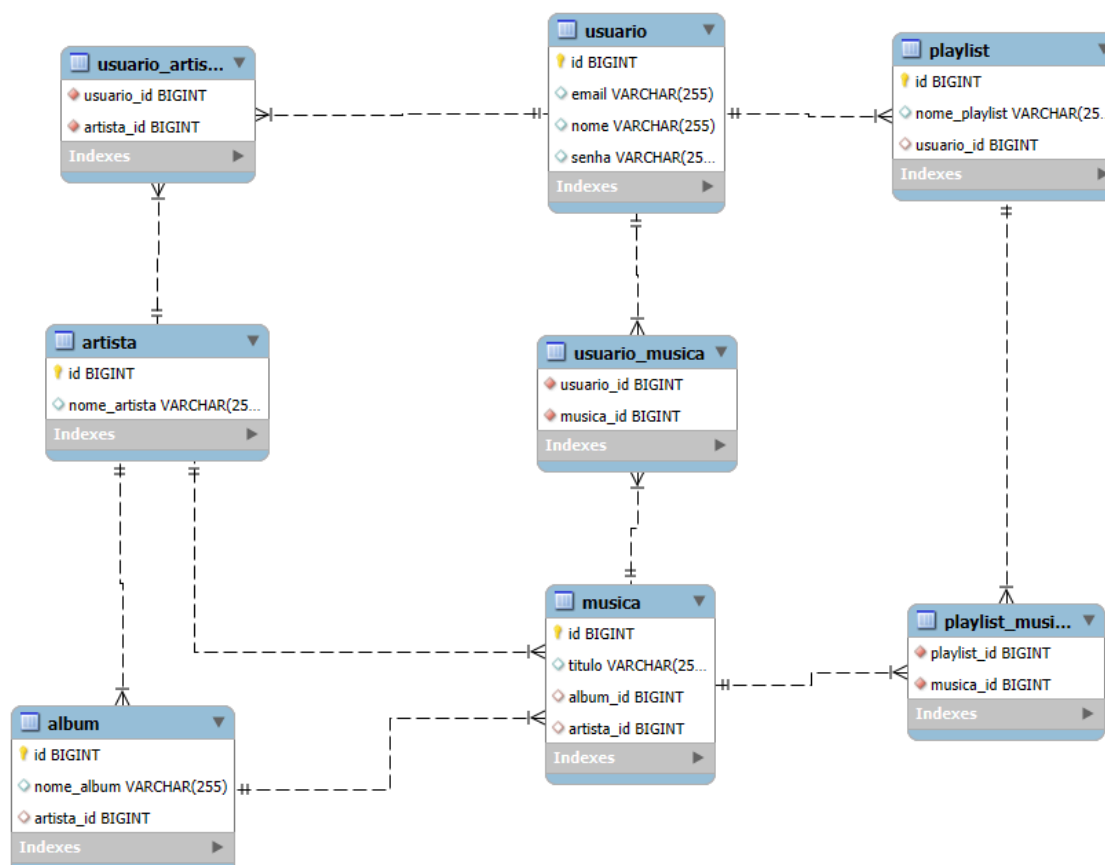


Figura do Modelo Entidade Relacionamento do SPM

4. Conclusão

O Sistema de Música SPM foi desenvolvido para oferecer uma experiência rica e interativa para usuários que desejam explorar, organizar e compartilhar suas músicas e playlists. Com base nas histórias de usuário, foram identificadas funcionalidades essenciais, como cadastro, criação de playlists e a opção de marcar músicas e artistas como favoritos. A estrutura do banco de dados foi planejada para garantir eficiência e integridade referencial entre as entidades, utilizando modelagem UML e o Modelo Entidade-Relacionamento (MER). Isso proporciona um sistema escalável e adaptável a futuras demandas.

Referências

GILLEANES T. A. GUEDES. UML 2: Uma Abordagem Prática. Disponível em: <https://books.google.com.br/books?hl=pt-BR&lr=&id=mJxMDwAAQBAJ&oi=fnd&pg=PA10&dq=diagrama+de+classe+uml&ots=xatWUjWLn1&sig=dNMmUtMILVBImIT9viAQMVCvam8#v=onepage&q=diagrama%20de%20classe%20uml&f=false>.

Acesso em: 29 set. 2024.

EDUARDO BEZERRA. Princípios de Análise e Projeto de Sistemas com UML. Disponível:

https://d1wqtxts1xzle7.cloudfront.net/36964560/AMS_Cap5_Diagrama_Classe-libre.pdf. Acesso em: 29 set. 2024.