Sistema de Música iTunes

Leonardo Latzke, Renan Apolinario

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

1. Introdução

O Sistema de Música iTunes é uma plataforma digital que permite aos usuários ouvir, criar playlists, seguir artistas e interagir com amigos. Inspirado por serviços como Spotify, o sistema oferece funcionalidades que visam proporcionar uma experiência musical rica e personalizada. O iTunes integra recursos sociais, permitindo que os usuários compartilhem suas músicas e playlists, promovendo a descoberta musical através de recomendações baseadas em gostos pessoais.

2. Requisitos Funcionais

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

2.1. História de Usuário 01

Como usuário do Sistema de música, eu quero buscar e ouvir músicas, para que eu possa desfrutar das minhas canções favoritas, muitos usuários vai ter uma lista de músicas que ele gosta.

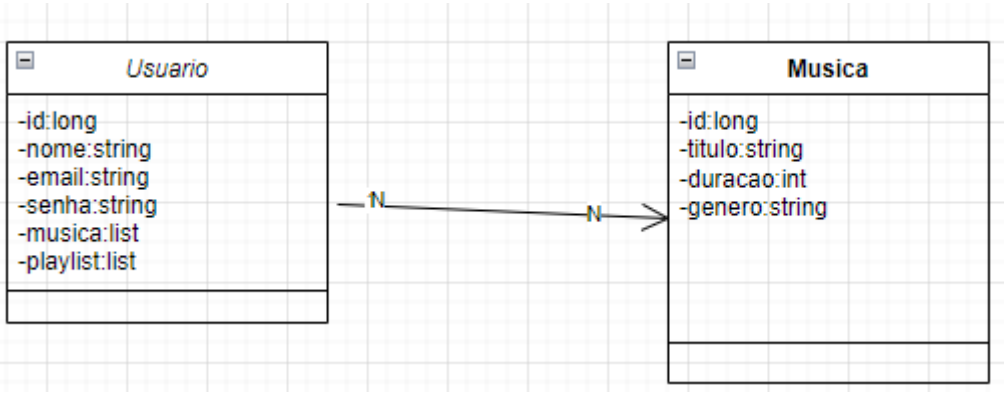

Figura 1. Diagrama de classe das entidades da História de Usuário 01.

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

2.2. História de Usuário 02

Como usuário do Sistema de música, eu quero criar e gerenciar playlists, para que eu possa organizar minhas músicas favoritas, muitas playlist vai ter muitas musicas adicionadas nela.

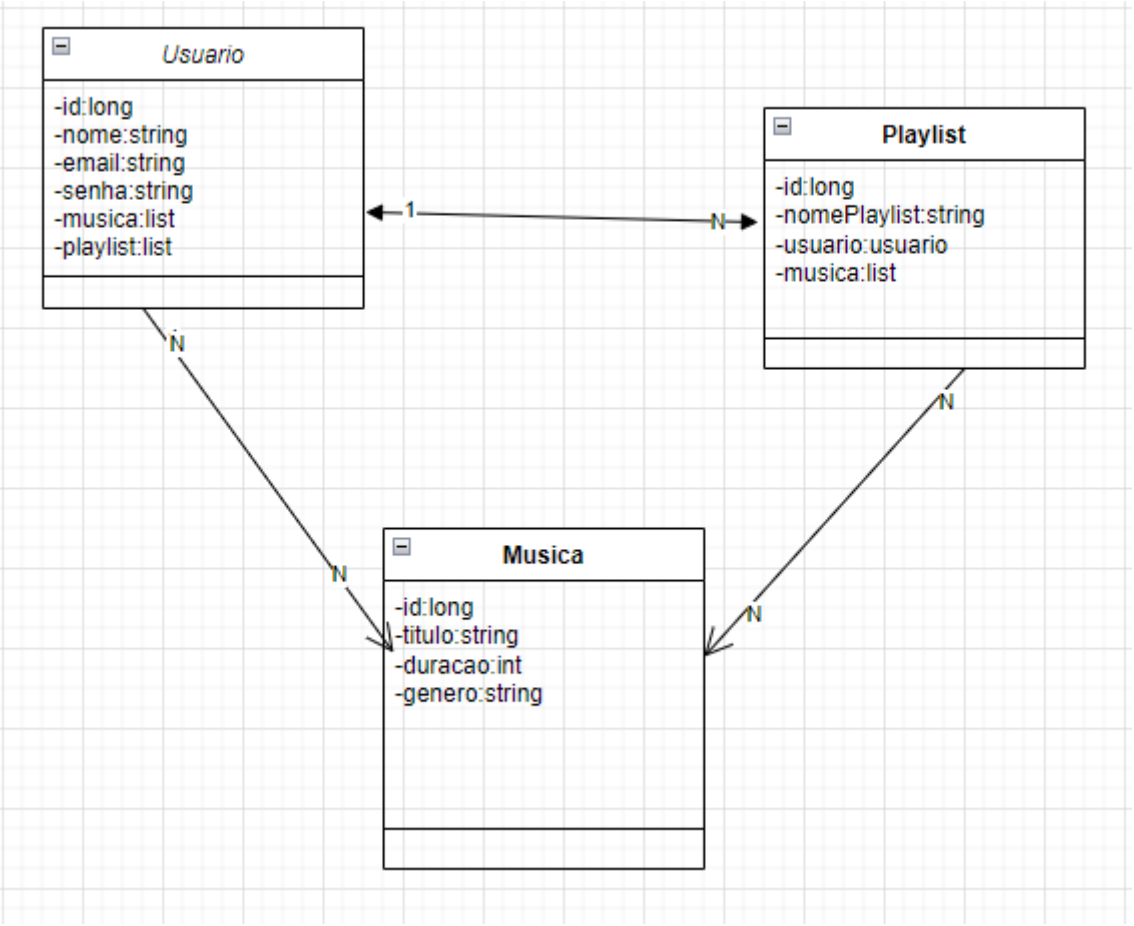

Figura 2. Diagrama de classe das entidades da historia de usuario 02

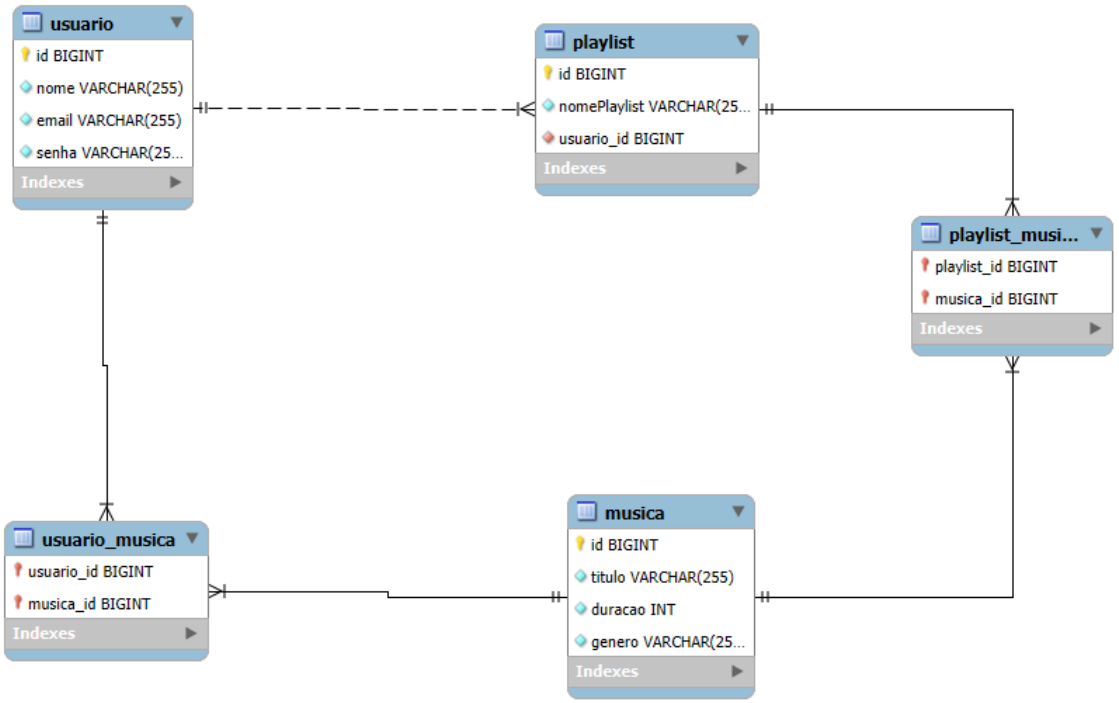

Figura 2. Modelo Entidade Relacionamento da História de Usuário 02.

2.3. História de Usuário 03

Como usuário do Sistema de música, eu quero explorar álbuns, para que eu possa descobrir novas músicas e artistas.

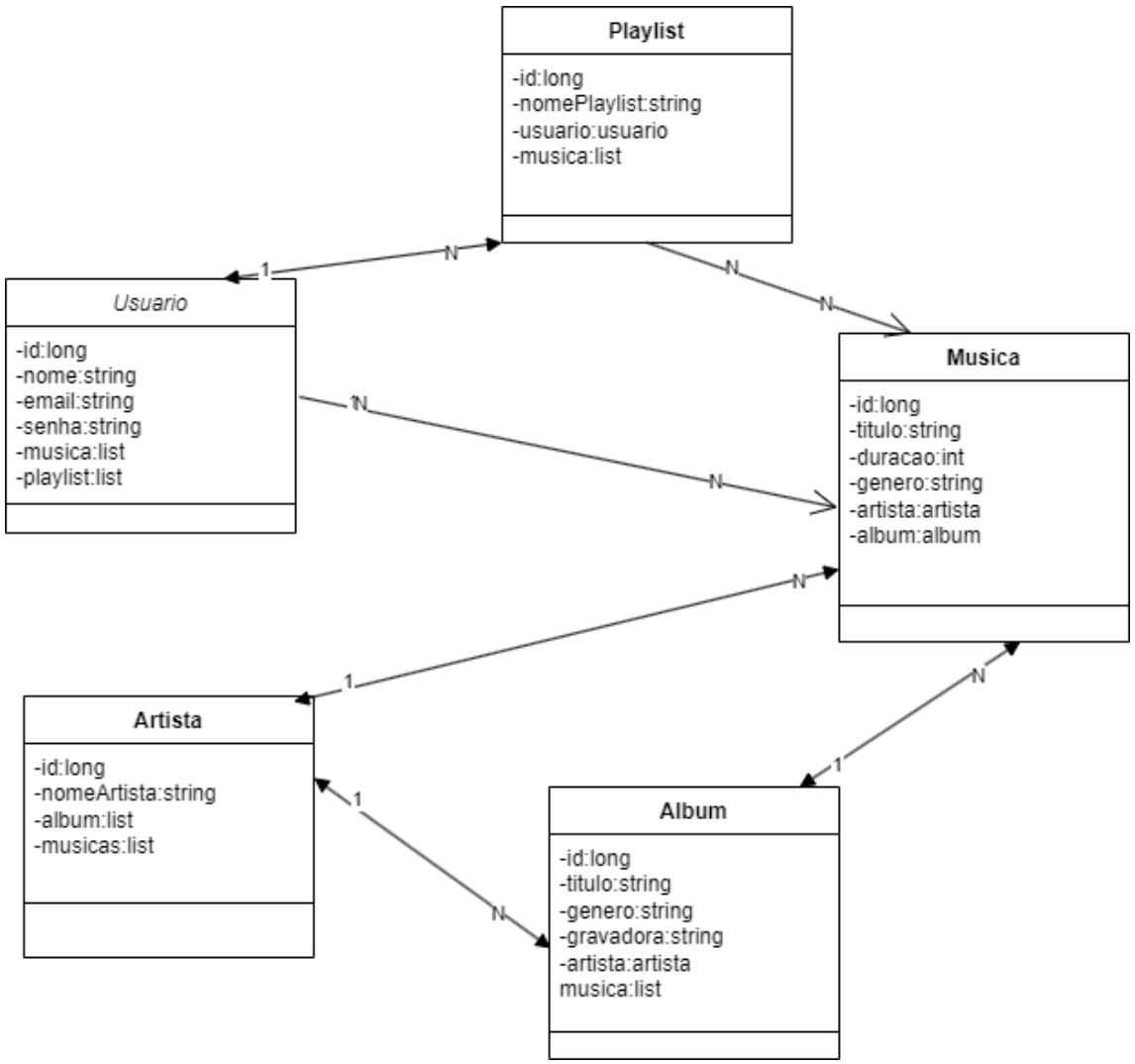

Figura 3. Diagrama de classe das entidades da historia de usuario 03

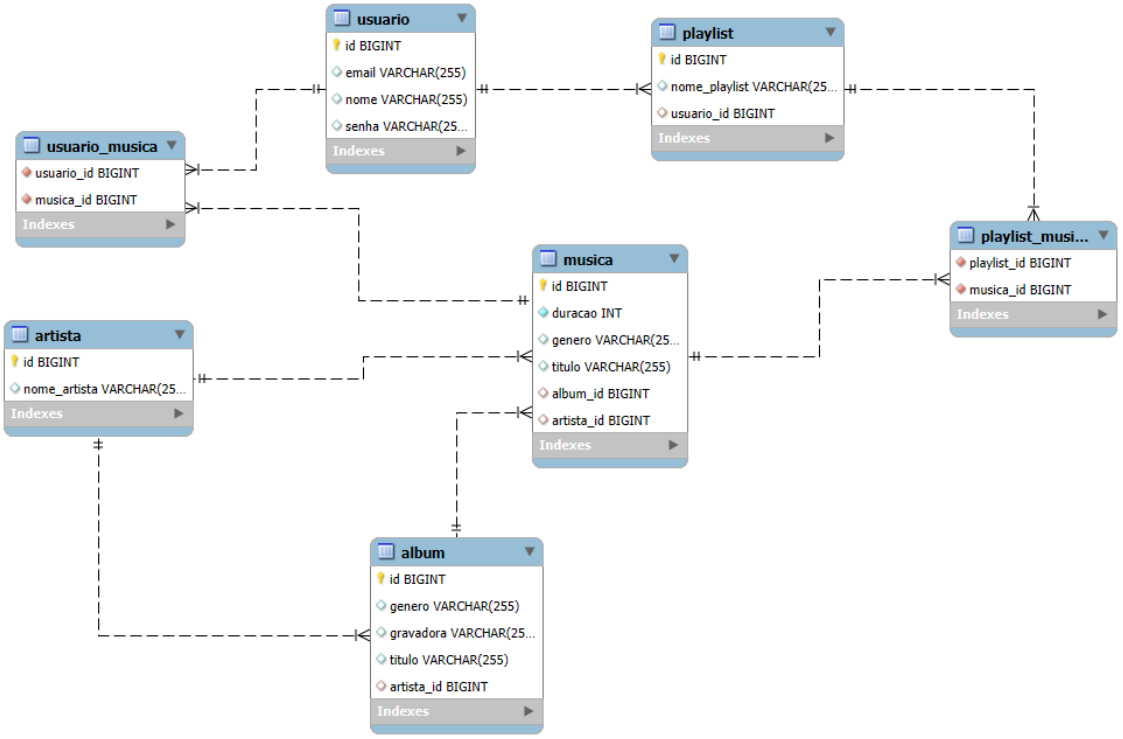

Figura 3. Modelo Entidade Relacionamento da História de Usuário 03.

3. Codificação

Apresentar o diagrama de classe UML

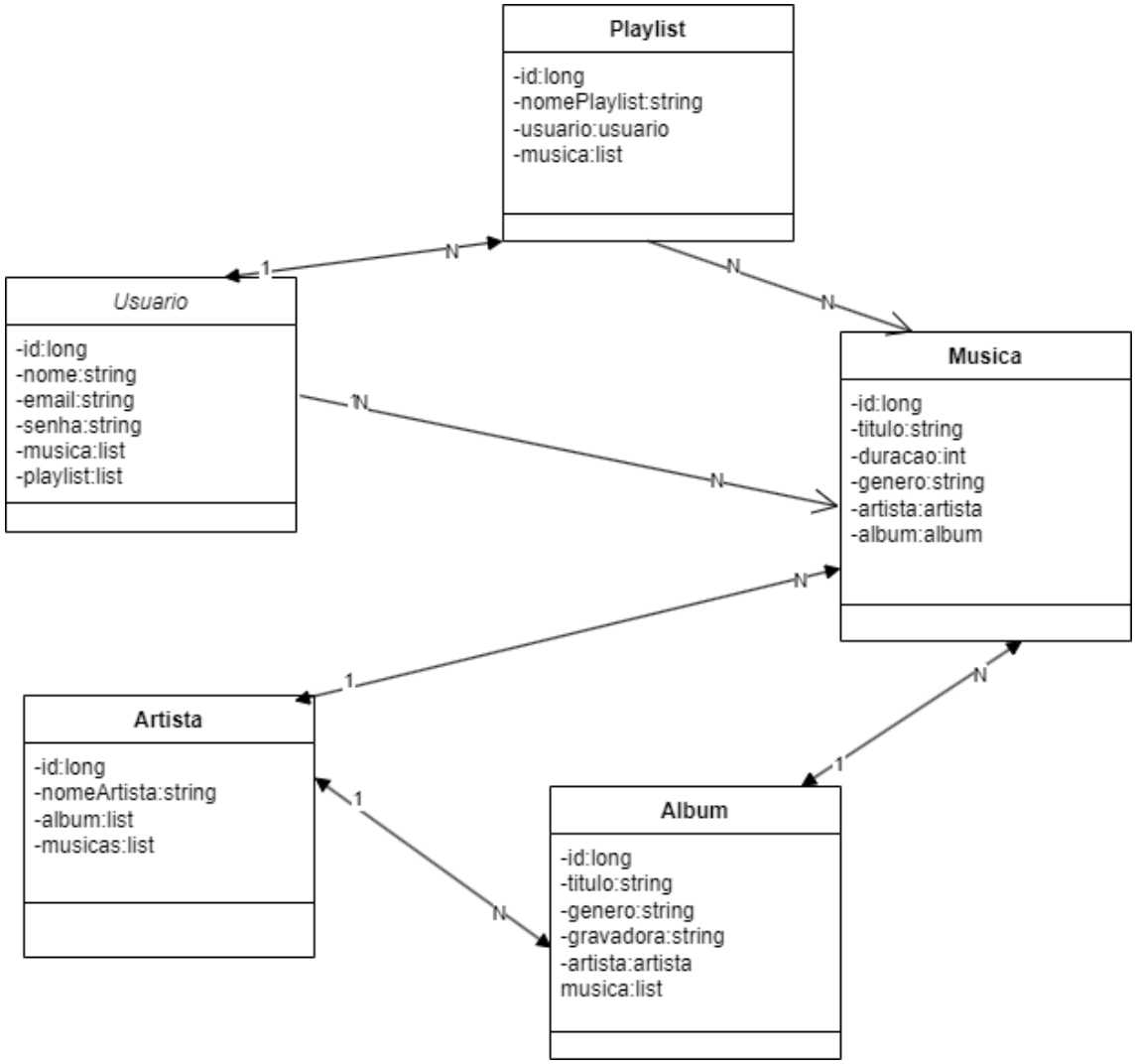

Figura 1. Diagrama de classe do Sistema de música iTunes

3.1. Entidade

Apresentar os códigos das entidades Usuário,Playlist,Música,Artista e Álbum .

```
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")
12:    private List<Playlist>playlists;
13:    @ManyToMany
14:    @JoinTable(name = "usuario_Musica",
15:      joinColumns = @JoinColumn(name = 15:"usuario_id"),
16:      inverseJoinColumns =  @JoinColumn(name = "musica_id"))
17:    private List<Musica>musicas;
18: }
```

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(name = "playlist_musica",
14:      joinColumns = @JoinColumn(name = "playlist_id"),
15:      inverseJoinColumns = @JoinColumn(name = "musica_id"))
16:   private List<Musica> musicas;
17: }
```

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:     private int duracao;
10:    private String genero;
11:    @ManyToOne
12:    @JoinColumn(name = "artista_id")
13:    private Artista artista;
14:    @ManyToOne
15:    @JoinColumn(name = "album_id")
16:    private Album album;
17: }
```

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")
10:    private List<Album> albuns;
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 titulo;
9:    private String genero;
10:    private String gravadora;
11:    @ManyToOne
12:    @JoinColumn(name = "artista_id")
13:    private Artista artista;
14:    @OneToMany(mappedBy = "album")
15:    private List<Musica> musicas;
16: }
```

Figura 5. Código da entidade Álbum.

4. Banco de dados

Apresentar as tabelas e os relacionamentos MER.

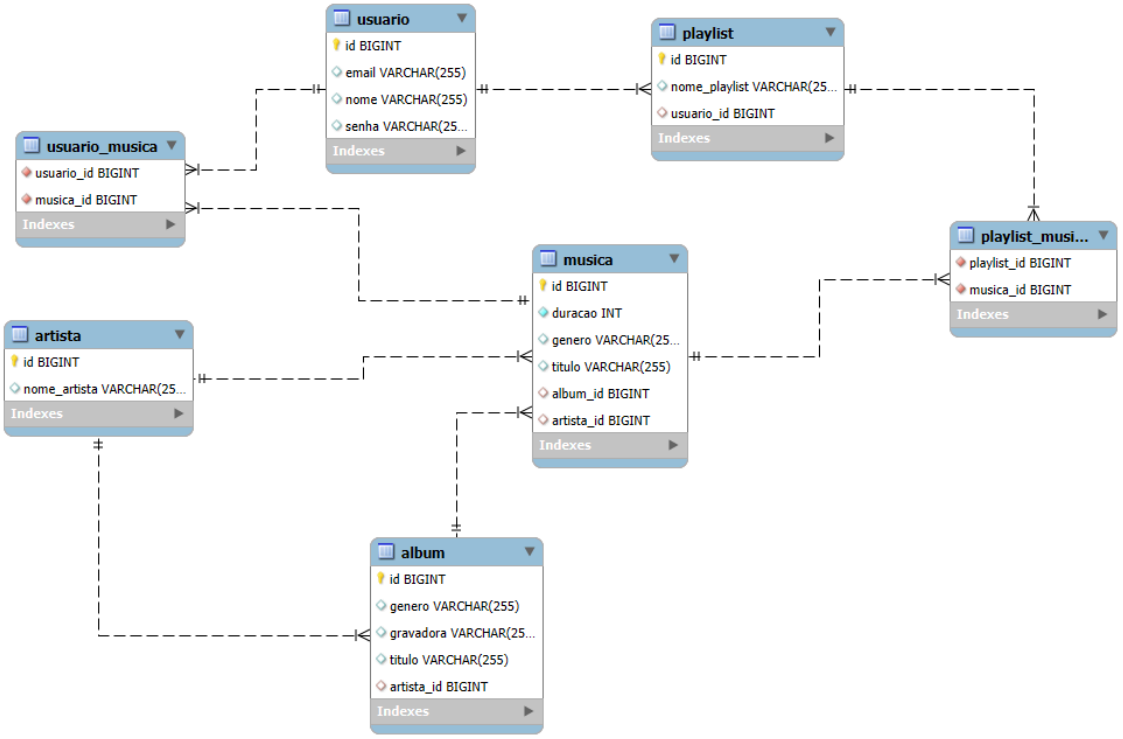

Figura 2. Modelo Entidade Relacionamento do Sistema de Música

4. Conclusão

O Sistema de Música iTunes foi projetado para atender às necessidades dos usuários que desejam uma experiência musical interativa e personalizada. As histórias de usuário apresentadas evidenciam funcionalidades essenciais, enquanto os diagramas fornecem uma visão clara da estrutura do sistema e dos relacionamentos entre suas entidades. A implementação de um banco de dados robusto e as relações adequadas entre as entidades garantem que o sistema será escalável e eficiente.

Referências

Boulic, R. and Renault, O. (1991) "3D Hierarchies for Animation", In: New Trends in Animation and Visualization, Edited by Nadia Magnenat-Thalmann and Daniel Thalmann, John Wiley & Sons ltd., England.

Dyer, S., Martin, J. and Zulauf, J. (1995) "Motion Capture White Paper", http://reality.sgi.com/employees/jam_sb/mocap/MoCapWP_v2.0.html, December.

Holton, M. and Alexander, S. (1995) "Soft Cellular Modeling: A Technique for the Simulation of Non-rigid Materials", Computer Graphics: Developments in Virtual Environments, R. A. Earnshaw and J. A. Vince, England, Academic Press Ltd., p. 449-460.

Knuth, D. E. (1984), The TeXbook, Addison Wesley, 15th edition.

Smith, A. and Jones, B. (1999). On the complexity of computing. In *Advances in Computer Science*, pages 555–566. Publishing Press.