

# Sistema de Música Boombox

Vladimir Pinto de Lima, Thales Natan Abreu Pedri

Engenharia de Software

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

vladimir.lima@univille.br, thales.pedri@univille.br

## 1. Introdução

O sistema a ser desenvolvido será uma plataforma de música similar ao Spotify. O objetivo é permitir que os usuários possam acessar e gerenciar playlists, músicas, álbuns e artistas. O sistema deverá permitir a interação entre usuários e suas bibliotecas musicais, além de oferecer a capacidade de explorar novos conteúdos, criar e compartilhar playlists, bem como ouvir seus artistas favoritos.

## 2. Requisitos Funcionais

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

### 2.1. História de Usuário 01

Como um usuário, eu quero me cadastrar no sistema de música, fornecendo meu nome, e-mail e senha, para poder acessar minha biblioteca de músicas e playlists personalizadas.

Diagrama de Classe (UML):

- **Entidades:**
  - **Usuário:** Contém atributos como **id**, **nome**, **email**, **senha**.

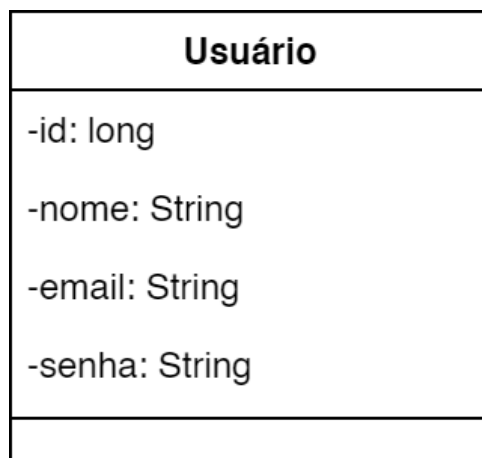


Figura 1. Diagrama de classe das entidades da História de Usuário 01.

Usuário		
PK	id	BIGINT
	nome	VARCHAR(255)
	email	VARCHAR(255)
	senha	VARCHAR(255)

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

## 2.2. História de Usuário 02

Como um usuário, eu quero criar e organizar minhas playlists pessoais, para poder gerenciar minhas músicas e artistas favoritos de forma personalizada.

**Modelo Entidade Relacionamento (MER):**

- **Entidades:**
  - **Música:** Contém *id*, *nome*, *artista*.
  - **Playlist:** Contém *id*, *nome*, *descrição*.
  - **Artista:** Contém *id*, *nome*, *gênero*.

**Diagrama UML:**

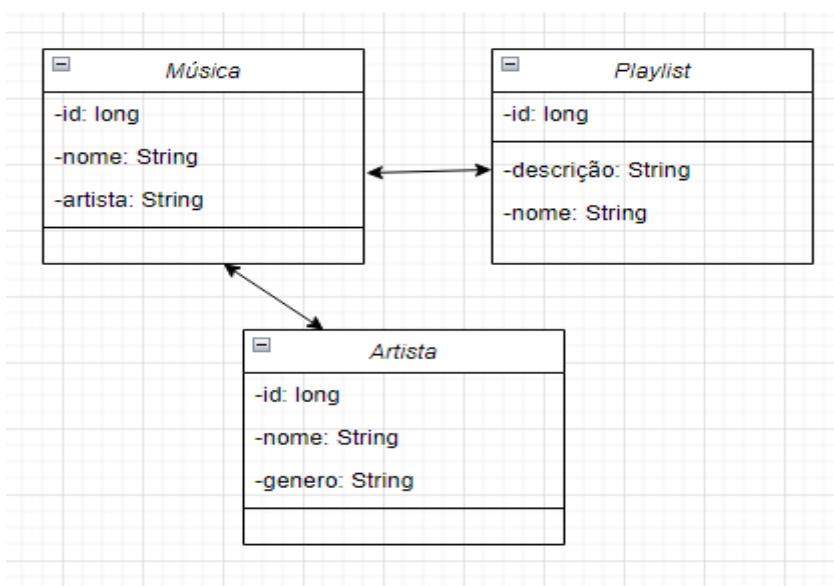


Figura 3. Diagrama de classe das entidades da História de Usuário 02.

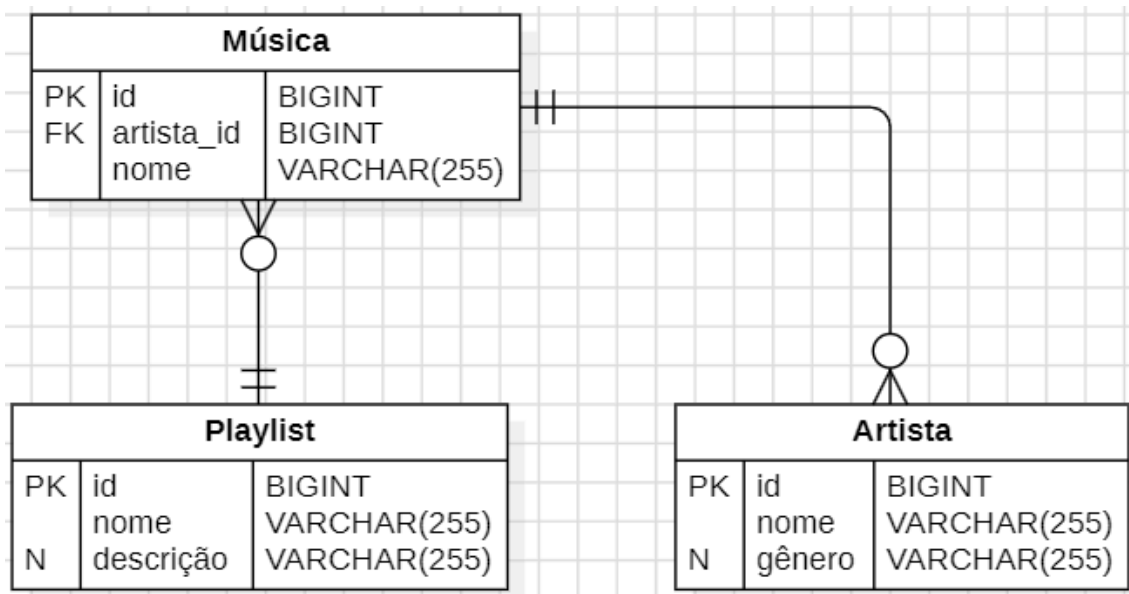


Figura 4. Modelo Entidade Relacionamento da História de Usuário 02.

### 3. Codificação

Para a construção de um sistema de música similar ao Spotify, iremos estruturar o banco de dados e os relacionamentos da seguinte forma:

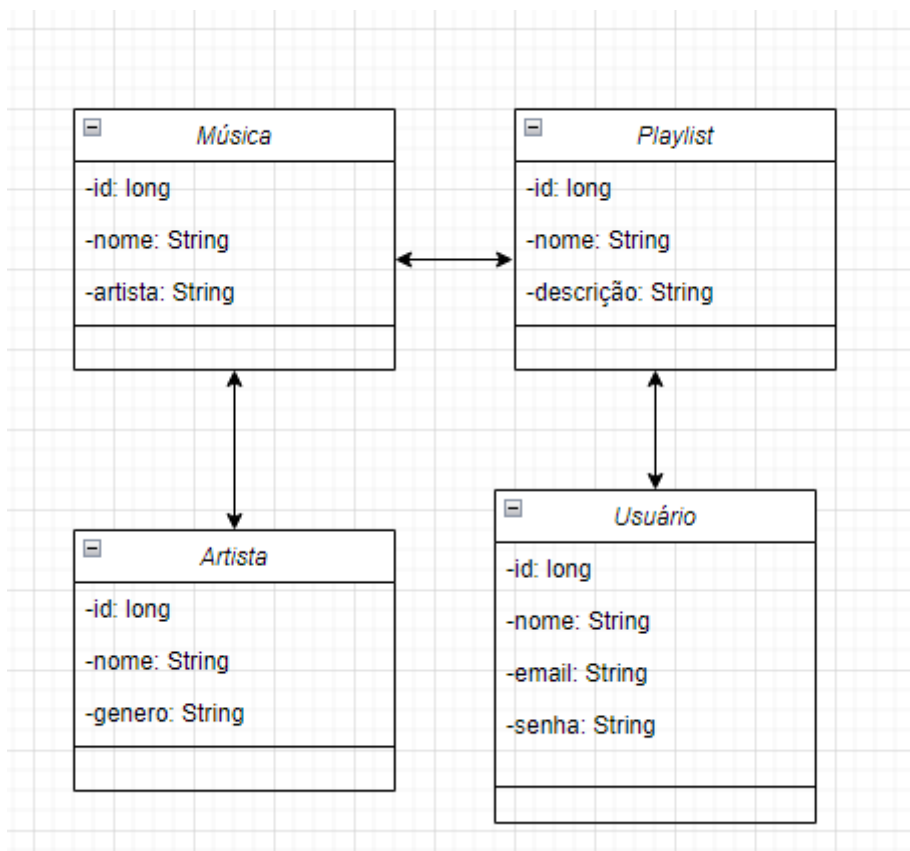


Figura 5. Diagrama de classe do Sistema de música Boombox

### 3.1. Entidade Música

Abaixo, mostramos um exemplo de como uma entidade pode ser codificada utilizando o framework JPA (Java Persistence API) para criar uma classe que represente o conceito de "Música" no sistema.

```
@Data
@NoArgsConstructor
@Entity
public class Musica {
    @Id
    @GeneratedValue
    private long id;
    @Column(nullable = false)
    private String nome;
    @Column(nullable = false)
    private long artista_id;
}
```

**Figura 6. Código da entidade Música**

#### 4. Banco de dados

O banco de dados será estruturado utilizando tabelas que refletem as entidades do sistema e seus relacionamentos. Abaixo, apresentamos o Modelo Entidade Relacionamento (MER) completo do sistema Boombox.

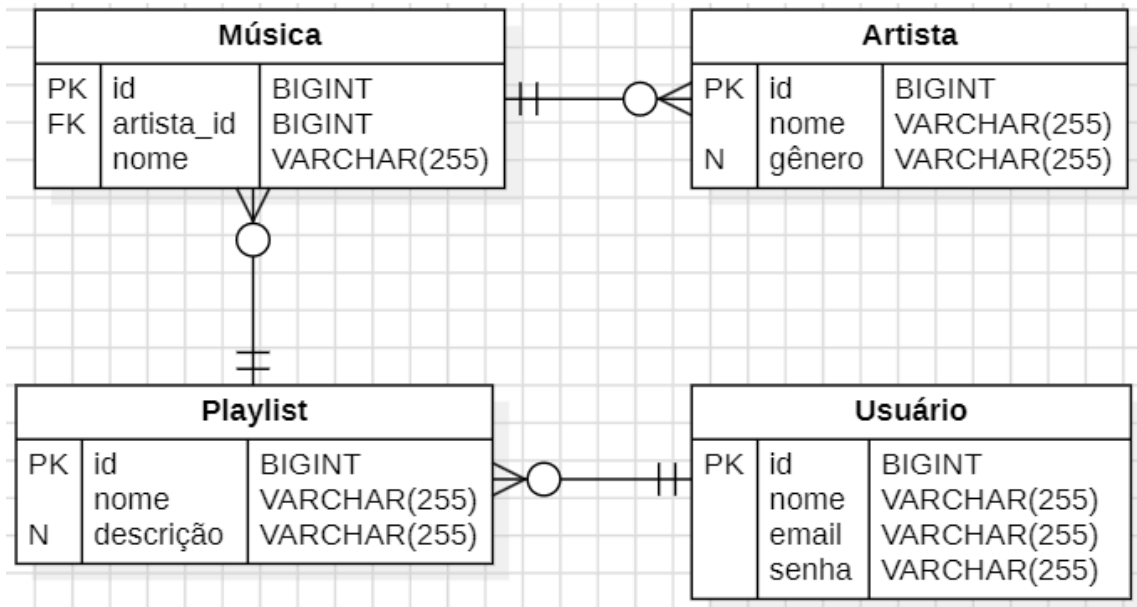


Figura 7. Modelo Entidade Relacionamento do Sistema de Música

#### 5. Conclusão

Neste documento, foram apresentadas as principais funcionalidades e entidades do sistema de música Boombox, um projeto inspirado em plataformas de streaming de música. Ao desenvolver o sistema, buscamos oferecer uma experiência personalizada para o usuário, focando na interação com playlists, músicas e artistas favoritos.