

Sistema de música de Guilherme

Guilherme Fuchter dos Santos

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

1.Introdução- Estamos criando uma plataforma musical inovadora, inspirada no Spotify, que vai transformar a experiência dos usuários com suas playlists e artistas. Com ela, será possível explorar um extenso catálogo, criar playlists personalizadas e gerenciar álbuns. A plataforma também promove a interação entre os usuários, permitindo o compartilhamento de playlists e descobertas musicais, além de acompanhar lançamentos em tempo real. Ideal para quem ama música e quer estar sempre por dentro das novidades.

2.Requisitos funcionais-Os requisitos funcionais são apresentados em forma de história de usuário.

2.1.História de usuário: Como usuário, quero me registrar no sistema de música, informando meu nome, e-mail e senha, para acessar minha biblioteca de músicas e playlists personalizadas.

Diagrama de Classe (UML): • Entidades: ○ Usuário: Contém atributos como [id](#), [nome](#), [email](#), [senha](#)

2.2. História de Usuário 02

Como 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](#)

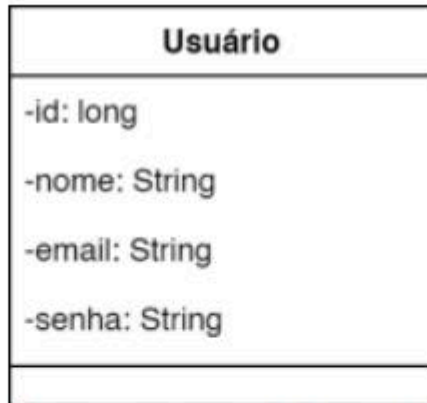


Figura 1. Diagrama de classe das entidades da História de Usuário 01.

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

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

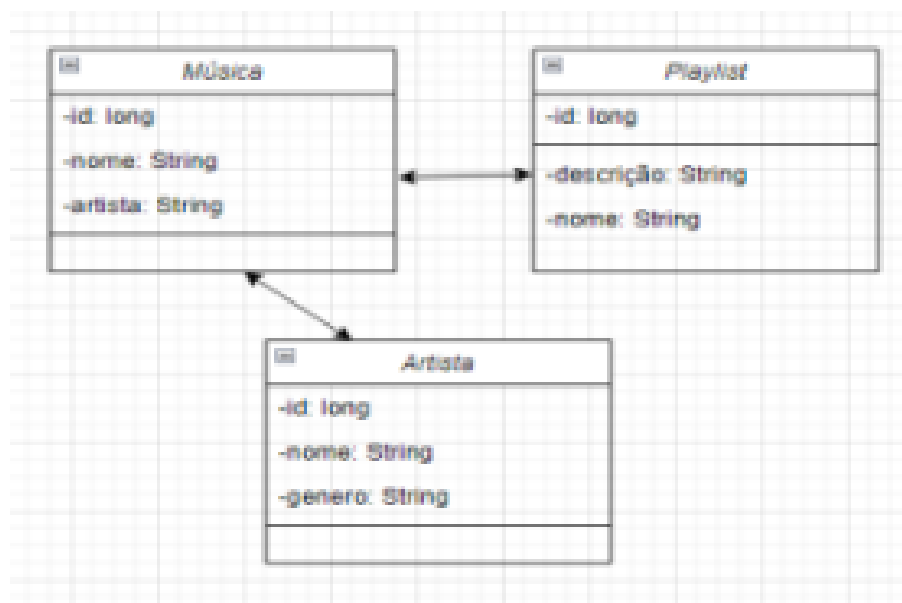


Figura 3. Diagrama de classe das entidades da História de Usuário 02.

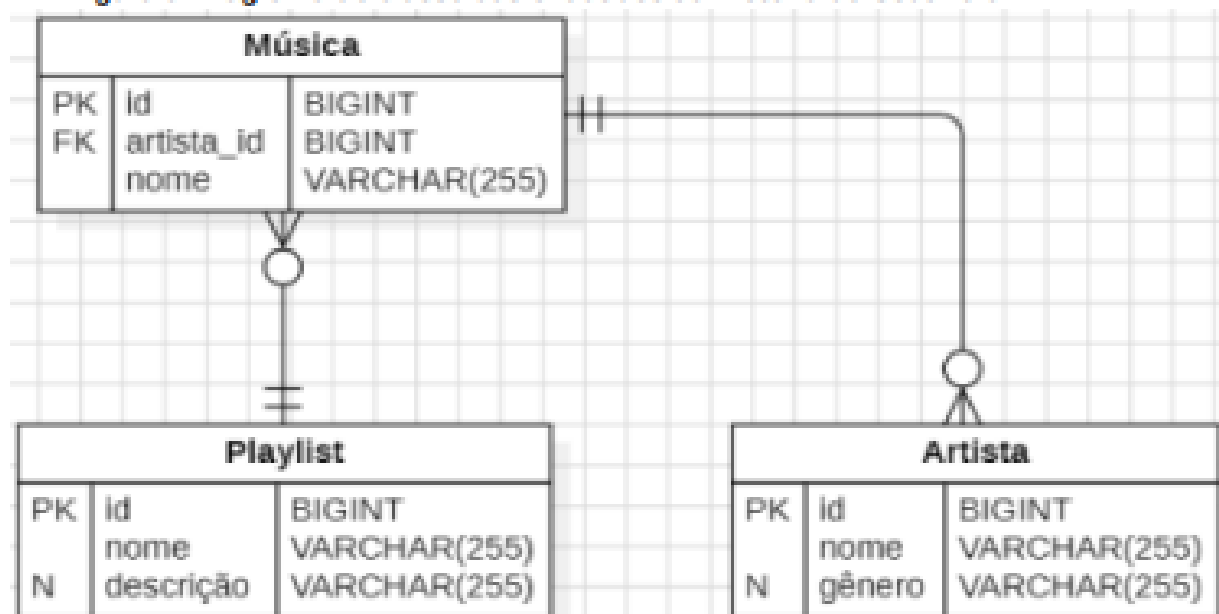
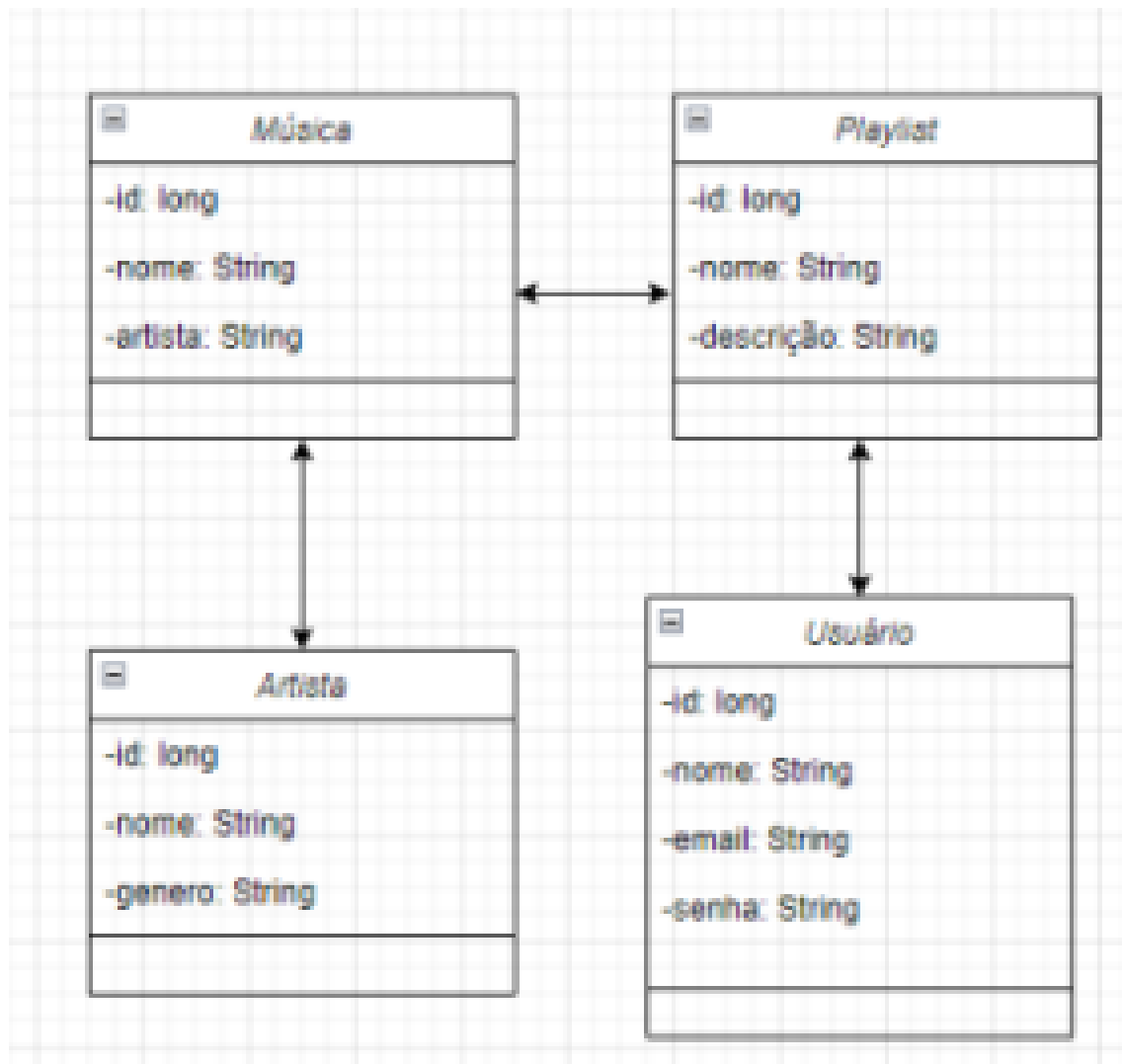


Figura 4. Modelo Entidade Relacionamento da História de Usuário 02.

3.Codificação

Para desenvolver um sistema de música similar ao Spotify, estruturamos o banco de dados e seus relacionamentos da seguinte maneira



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
```

3.2 Banco de Dados

O banco de dados será organizado com tabelas que representam as entidades do sistema e seus respectivos relacionamentos. A seguir, apresentamos o Modelo Entidade-Relacionamento (MER) completo do sistema.

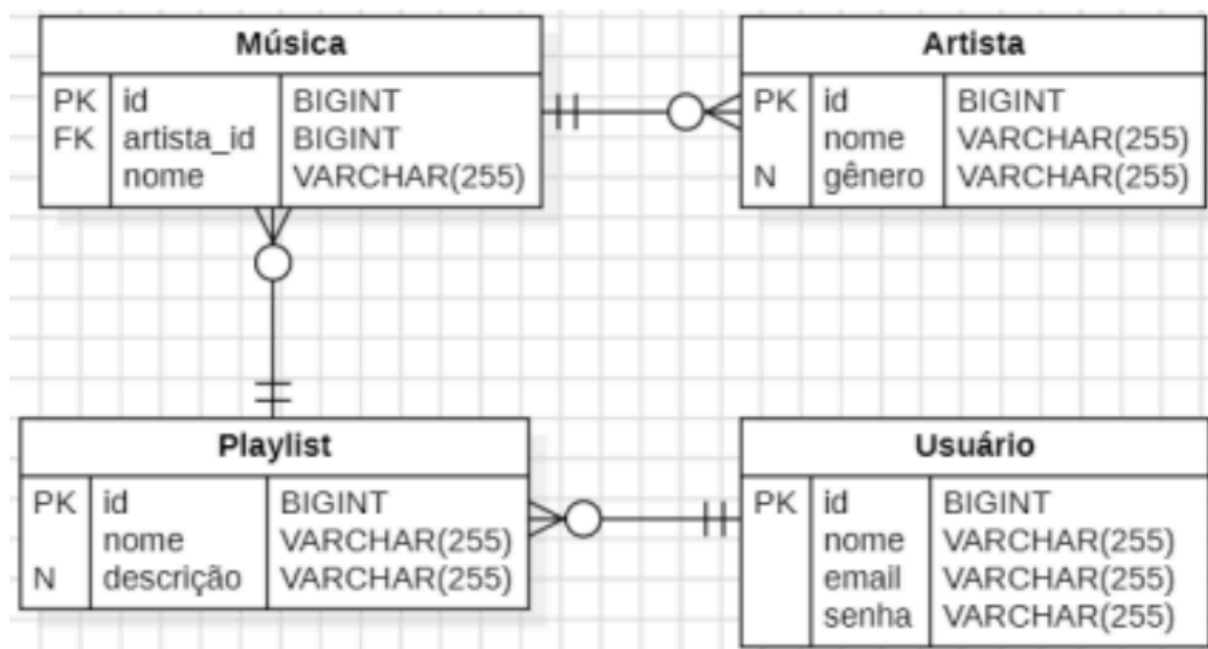


Figura 7. Modelo Entidade Relacionamento do Sistema de Música

5 Conclusão:

Neste documento, apresentamos as principais funcionalidades e entidades do sistema de música, inspirado em plataformas populares de streaming. Nosso objetivo é criar uma experiência envolvente e personalizada, facilitando a conexão dos usuários com suas playlists, músicas e artistas preferidos. Durante o desenvolvimento, priorizamos a criação de uma experiência centrada no usuário, com foco na interação com playlists e artistas favoritos.