

Sistema de Gestão de Consultas Médicas - MediX

Marcelo Pottier, Pedro Hideki

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

marcelo.pottier@univille.br, pedro.morikawa@univille.br

1. Introdução

O Sistema de Música Melody é uma plataforma desenvolvida para fornecer aos usuários uma experiência completa de streaming musical. Com o objetivo de facilitar a descoberta, organização e compartilhamento de músicas, o Melody se propõe a ser o principal aplicativo de música para usuários que desejam explorar novos sons e interagir com amigos.

Destinado a amantes da música de todas as idades, o sistema possui uma interface amigável que permite aos usuários acessar suas faixas favoritas, criar playlists personalizadas e descobrir novos artistas com base em suas preferências. A principal finalidade do Melody é otimizar a experiência de escuta, integrando recursos como recomendações inteligentes, funcionalidades sociais e acesso offline.

2. Requisitos Funcionais

Os requisitos funcionais do Sistema Melody serão apresentados em forma de histórias de usuário..

2.1. História de Usuário 01

Como usuário, eu quero receber recomendações de músicas baseadas nas minhas preferências, para que eu possa descobrir novos artistas e faixas.

Critérios de Aceitação:

- O sistema deve analisar o histórico de escuta do usuário para gerar recomendações personalizadas.
- As recomendações devem ser atualizadas regularmente com base nas novas músicas ou alterações nas preferências do usuário.
- O usuário deve ter a opção de "Curtir" ou "Descurtir" as músicas recomendadas.

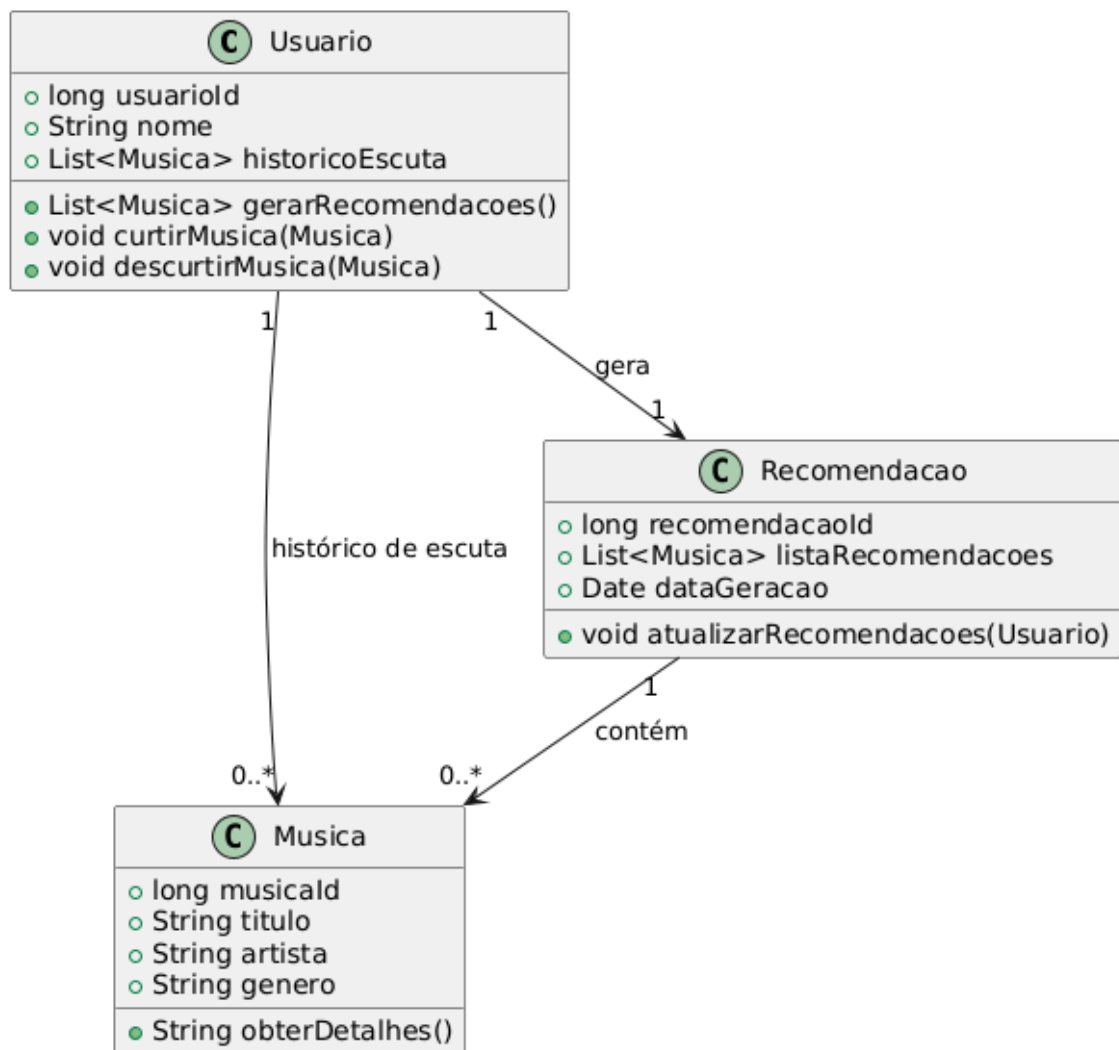


Figura 1. Diagrama de classe das entidades da História de Usuário 01.

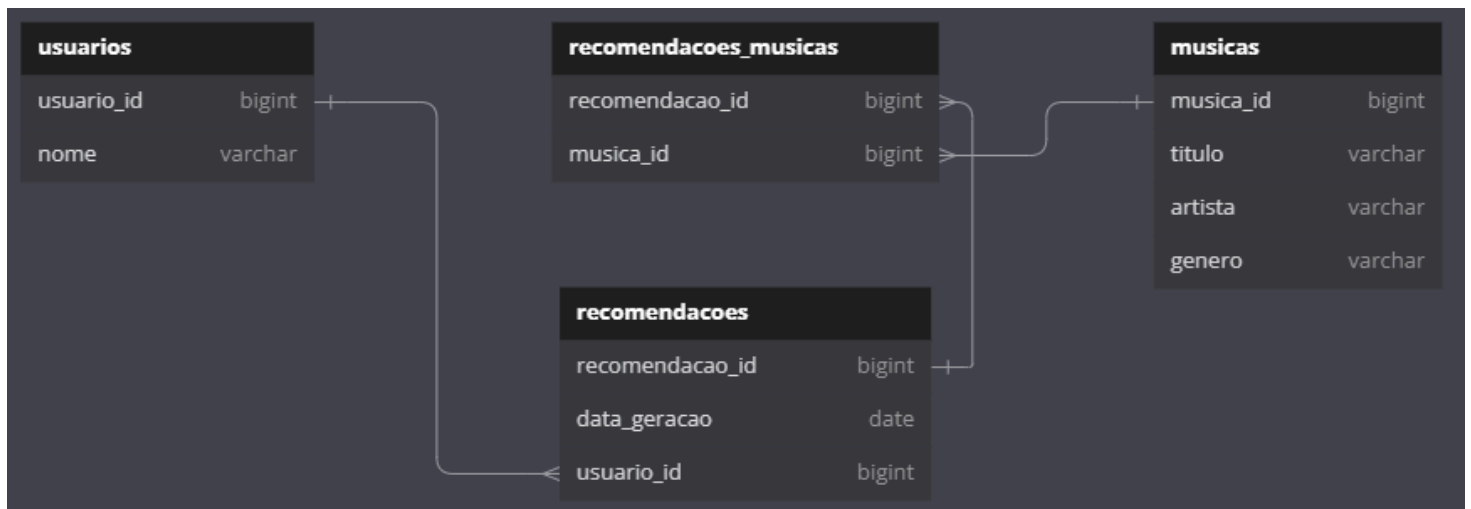


Figura 2. Modelo Entidade Relacionamento da História de Usuário 01.

A Figura 02 é o MER da história de usuário 01. O diagrama ER da História de Usuário 01 ilustra as tabelas que compõem a estrutura de recomendação de músicas do sistema Melody. A tabela RECOMENDACOES_MUSICAS contém as chaves estrangeiras das tabelas Recomendacoes e Musicas, estabelecendo o relacionamento de muitos para muitos entre recomendações e músicas.

A tabela Usuarios contém informações sobre os usuários do sistema, enquanto a tabela Musicas armazena os dados das faixas musicais disponíveis. A tabela Recomendacoes é associada diretamente a cada usuário, contendo as sugestões de músicas baseadas nas preferências do usuário. A tabela RECOMENDACOES_MUSICAS realiza a ligação entre as músicas recomendadas e as recomendações geradas, garantindo a integridade dos dados por meio das chaves estrangeiras.

Outros pontos importantes:

- As chaves primárias garantem a unicidade dos registros em todas as tabelas.
- As chaves estrangeiras asseguram a integridade referencial, ligando uma recomendação a um usuário específico e uma música específica.
- A tabela de relacionamento RECOMENDACOES_MUSICAS implementa uma relação de muitos para muitos entre Recomendacoes e Musicas.

2.2. História de Usuário 02

Como usuário, eu quero criar e compartilhar playlists personalizadas, para que eu possa organizar minhas músicas favoritas e dividir com amigos.

Critérios de Aceitação:

- O usuário deve ser capaz de criar playlists e adicionar ou remover músicas facilmente.
- O sistema deve permitir ao usuário compartilhar playlists com amigos através de links ou nas redes sociais.
- As playlists podem ser configuradas como públicas ou privadas.

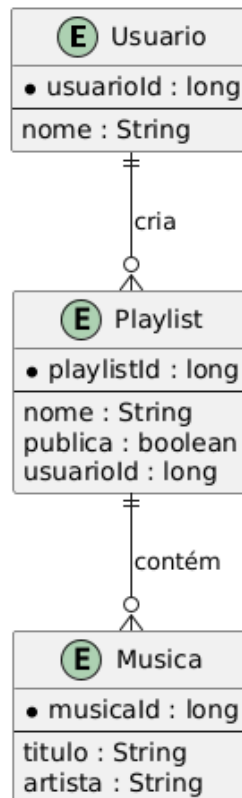


Figura 03. Diagrama de classe das entidades da História de Usuário 02.

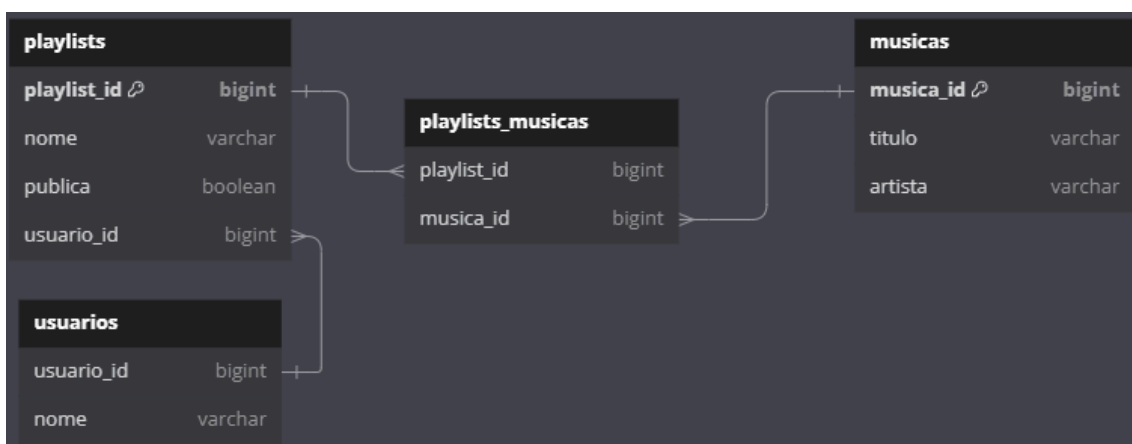


Figura 04: O MER da História de Usuário 02 exibe as tabelas relacionadas à criação e compartilhamento de playlists. A tabela PLAYLISTS_MUSICAS

contém as chaves estrangeiras das tabelas Playlists e Musicas, criando uma relação de muitos para muitos entre playlists e músicas.

- A tabela Usuarios armazena os usuários que criam e gerenciam playlists.
- A tabela Playlists contém as informações sobre cada playlist.
- A tabela Musicas armazena as faixas que podem ser adicionadas às playlists.
- A tabela PLAYLISTS_MUSICAS liga músicas a playlists, permitindo que cada playlist tenha várias músicas e que uma música faça parte de várias playlists.

As chaves primárias e estrangeiras garantem a integridade dos dados e dos relacionamentos.

2.3. História de Usuário 03

Como usuário que viaja, eu quero baixar músicas e playlists para ouvir offline, para que eu possa desfrutar da música sem depender de internet.

Critérios de Aceitação:

- **O sistema deve oferecer a opção de download para músicas e playlists.**
- **O usuário deve ser notificado sobre o espaço disponível no dispositivo antes de realizar downloads.**
- **As músicas baixadas devem ser acessíveis mesmo sem conexão à internet.**

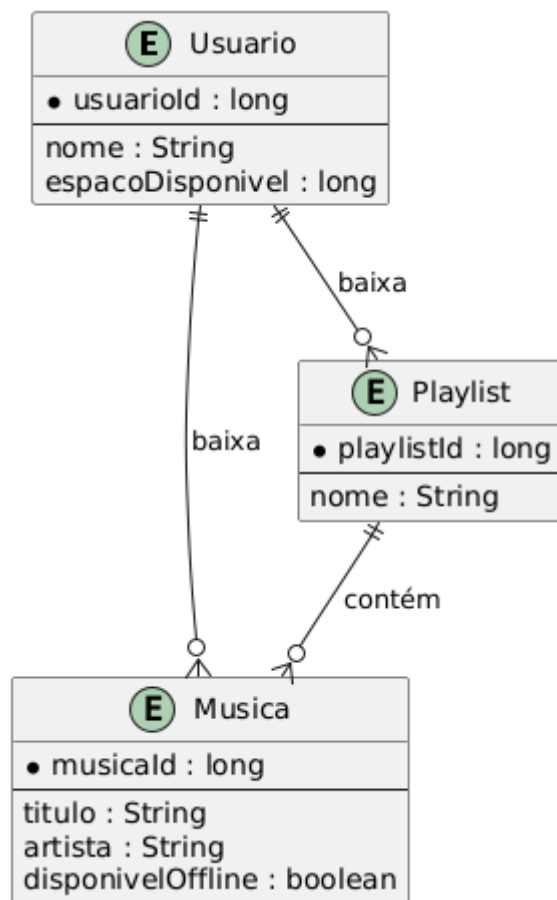


Figura 5. Diagrama de classe das entidades da História de Usuário 03.



Figura 06: O MER da História de Usuário 03 mostra as tabelas responsáveis pelo download de músicas e playlists. As tabelas **USUARIOS_MUSICAS_BAIXADAS** e **USUARIOS_PLAYLISTS_BAIXADAS** contêm as chaves estrangeiras das tabelas **Usuarios**, **Musicas**, e **Playlists**.

- A tabela Usuarios armazena as informações dos usuários e o espaço disponível para download.
- A tabela Musicas contém as músicas disponíveis para download.
- A tabela Playlists armazena as playlists que podem ser baixadas.
- As tabelas USUARIOS_MUSICAS_BAIXADAS e USUARIOS_PLAYLISTS_BAIXADAS estabelecem o relacionamento de muitos para muitos, ligando os usuários às músicas e playlists baixadas.

As chaves estrangeiras garantem que cada música ou playlist baixada esteja associada a um usuário específico.

2.4. História de Usuário 04

Como fã de música, eu quero ver as letras das músicas enquanto ouço, para que eu possa acompanhar a canção.

Critérios de Aceitação:

- **O sistema deve exibir as letras em tempo real durante a reprodução da música.**
- **O usuário deve ter a opção de ocultar ou mostrar as letras conforme preferir**

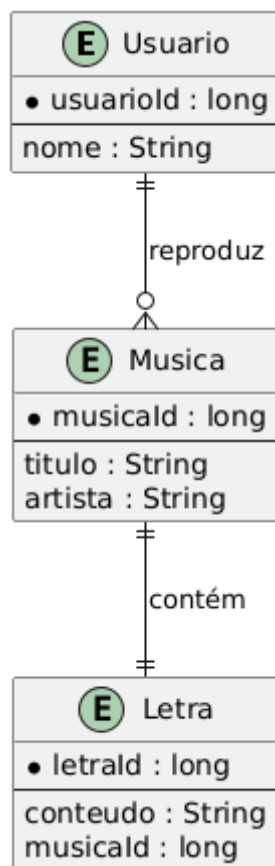


Figura 06. Diagrama de classe das entidades da História de Usuário 04.

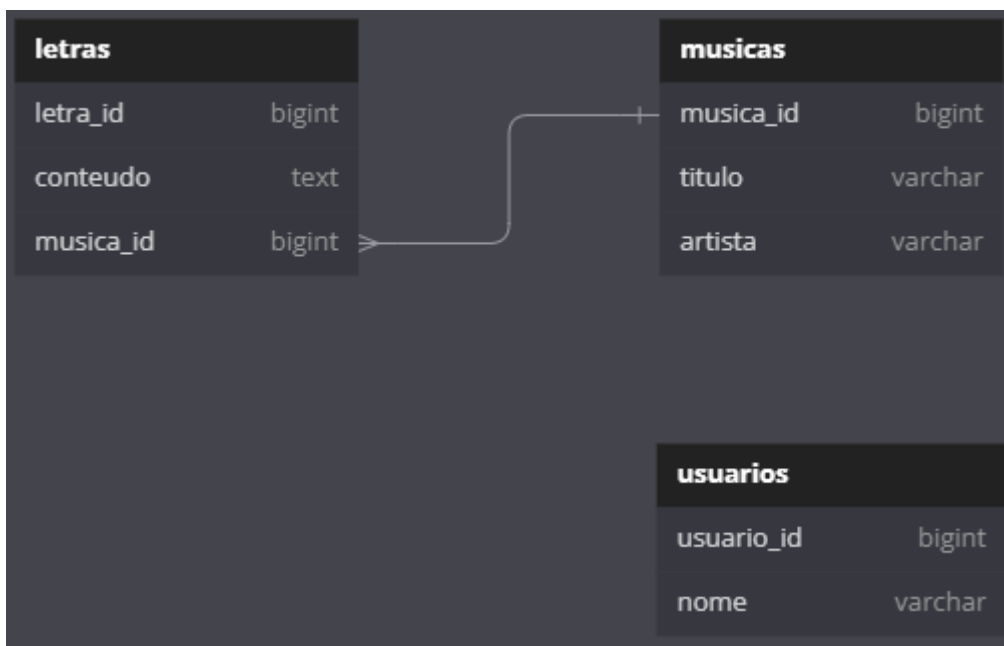


Figura 07: O MER da História de Usuário 04 exibe a relação entre músicas e suas letras. A tabela LETRAS contém uma chave estrangeira que referencia a tabela Musicas.

- A tabela Musicas armazena as informações das faixas musicais.
- A tabela LETRAS armazena as letras das músicas, associadas diretamente a cada música.

Esse relacionamento é de um para um, com cada música tendo uma letra correspondente.

2.5. História de Usuário 05

Como usuário social, eu quero seguir meus amigos e ver o que eles estão ouvindo, para que eu possa descobrir novas músicas e compartilhar experiências.

Critérios de Aceitação:

- O sistema deve permitir que os usuários sigam uns aos outros.
- Um feed de atividades deve ser criado para mostrar o que os amigos estão ouvindo e suas playlists.
- O usuário deve ter a opção de compartilhar músicas diretamente com amigos.

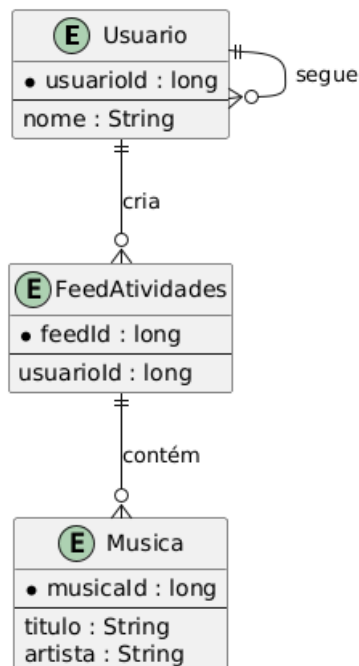


Figura 8. Diagrama de classe das entidades da História de Usuário 05.

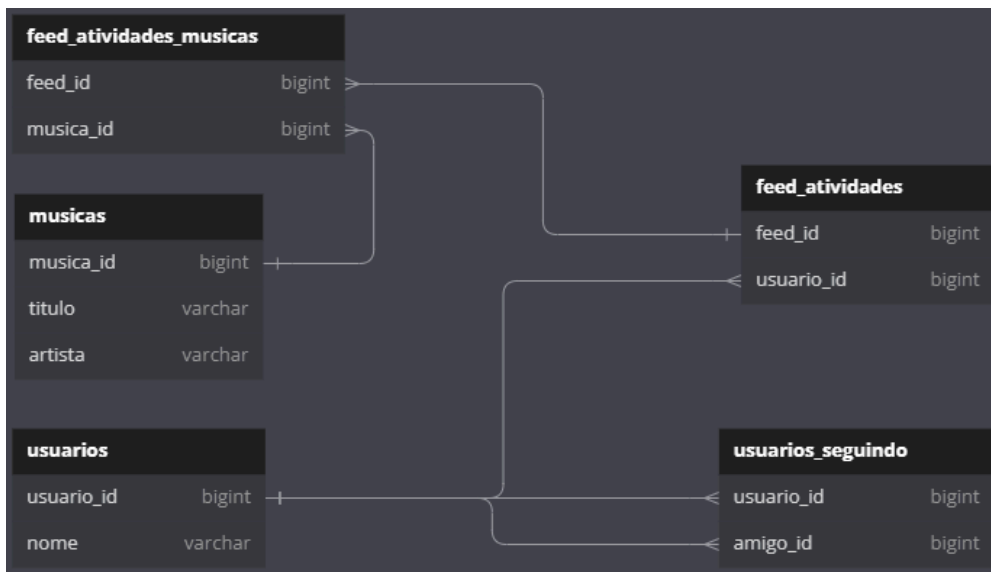


Figura 09: O MER da História de Usuário 05 apresenta as tabelas para seguir amigos e visualizar atividades de escuta. A tabela USUARIOS_SEGUINDO contém as chaves estrangeiras da tabela Usuarios, formando uma relação de muitos para muitos entre usuários.

- A tabela Usuarios armazena os dados dos usuários.
- A tabela FeedAtividades contém as atividades dos usuários, como músicas que estão ouvindo.
- A tabela USUARIOS_SEGUINDO liga usuários entre si para permitir o seguimento de amigos.
- A tabela FEEDATIVIDADES_MUSICAS associa atividades com músicas específicas.

As chaves estrangeiras garantem que as atividades e o seguimento de usuários sejam consistentes e bem estruturados.

3. Codificação

3.1. Entidade Recomendacao

A entidade Recomendação representa o conjunto de músicas sugeridas a um usuário, baseado em seu histórico de escuta e preferências. Esta entidade possui um relacionamento de muitos para um com a entidade Usuario, indicando que cada recomendação é gerada para um único usuário. Além disso, a Recomendação possui um relacionamento de muitos para muitos com a entidade Música, por meio da tabela associativa recomendacao_musicas,

permitindo que uma recomendação contenha várias músicas e que uma música possa aparecer em várias recomendações.

```
@Data
@NoArgsConstructor
@Entity
public class Recomendacao {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private long id;

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

    private Date dataGeracao;

    @ManyToMany
    @JoinTable(
        name = "recomendacao_musicas",
        joinColumns = @JoinColumn(name = "recomendacao_id"),
        inverseJoinColumns = @JoinColumn(name = "musica_id")
    )
    private List<Musica> musicasRecomendadas;
}
```

Figura 10. Código da entidade Recomendacao

3.2. Entidade Playlist

A entidade Playlist representa uma coleção de músicas organizada por um usuário, que pode ser compartilhada com amigos. A Playlist está associada a um relacionamento de muitos para um com a entidade Usuario, indicando que cada playlist pertence a um único usuário. Além disso, a Playlist possui um relacionamento de muitos para muitos com a entidade Musica, por meio da

tabela associativa `playlist_musicas`, permitindo que uma `playlist` contenha várias músicas e que uma música possa ser adicionada a várias `playlists`.

```
@Data
@NoArgsConstructor
@Entity
public class Playlist {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private long id;

    private String nome;

    private boolean publica;

    @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;
}
```

Figura 11. Código da entidade `Playlist`

3.3. Entidade `Usuario`

A entidade `Usuário` representa os indivíduos que utilizam o sistema, sendo capazes de criar `playlists`, receber recomendações e seguir outros usuários. O `Usuário` possui um relacionamento de um para muitos com a entidade `Playlist`, indicando que um usuário pode ter várias `playlists`. Além disso, a entidade `Usuário` possui um relacionamento de muitos para muitos com ela mesma, por

meio da tabela associativa usuários seguindo, permitindo que os usuários sigam uns aos outros e construam uma rede social dentro do sistema.

```
@Data
@NoArgsConstructor
@Entity
public class Usuario {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private long id;

    private String nome;

    private long espacoDisponivel;

    @ManyToMany
    @JoinTable(
        name = "usuarios_seguindo",
        joinColumns = @JoinColumn(name = "usuario_id"),
        inverseJoinColumns = @JoinColumn(name = "amigo_id")
    )
    private List<Usuario> amigos;

    @OneToMany(mappedBy = "usuario")
    private List<Playlist> playlists;
}
```

Figura 12. Código da entidade Usuario

3.4. Entidade Musica

A entidade Musica representa as faixas musicais disponíveis no sistema. Cada música possui um relacionamento de um para um com a entidade Letra, indicando que cada música tem uma letra correspondente. A Musica também tem um relacionamento de muitos para muitos com a entidade Recomendacao, permitindo que uma música seja incluída em várias recomendações e que cada

recomendação possa conter várias músicas. Além disso, a Musica é parte de um relacionamento de muitos para muitos com a Playlist, permitindo que uma música seja adicionada a várias playlists.

```
@Data
@NoArgsConstructor
@Entity
public class Musica {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private long id;

    private String titulo;

    private String artista;

    private String genero;

    private boolean disponivelOffline;

    @OneToOne(mappedBy = "musica")
    private Letra letra;
}
```

Figura 13. Código da entidade Musica

3.5. Entidade Letra

A entidade Letra representa as letras das músicas, que podem ser exibidas durante a reprodução. A Letra possui um relacionamento de um para um com a entidade Musica, indicando que cada letra está associada a uma única música. Essa relação permite que o sistema forneça as letras correspondentes às músicas que estão sendo reproduzidas, enriquecendo a experiência do usuário.

```
@Data
@NoArgsConstructor
```

```
@Entity
public class Letra {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private long id;

    @OneToOne
    @JoinColumn(name = "musica_id")
    private Musica musica;

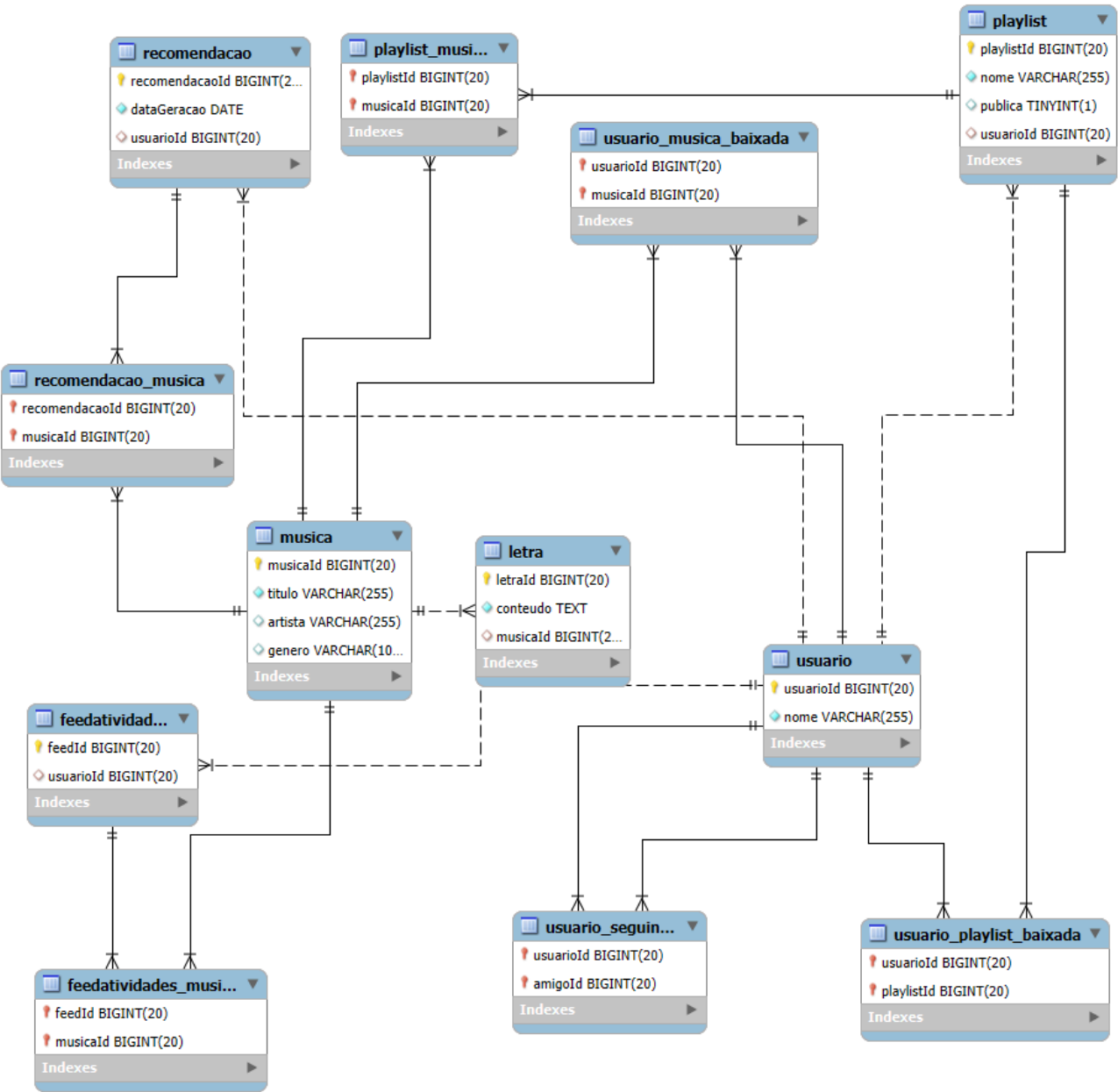
    private String conteudo;
}
```

Figura 14. Código da entidade Letra

4. Banco de dados

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

Figura 15. Modelo Entidade Relacionamento do Sistema de Melody



4. Conclusão

Em conclusão, o Sistema de Música Melody se destaca como uma solução abrangente para os amantes da música, combinando uma interface intuitiva com recursos inovadores que facilitam a descoberta e o compartilhamento musical. Ao oferecer recomendações personalizadas e funcionalidades sociais, o Melody não apenas enriquece a experiência de escuta, mas também promove a interação entre os usuários. Com a promessa de atender a todas as faixas etárias e preferências musicais, o Melody se posiciona como a escolha ideal para quem busca uma plataforma dinâmica e acessível, consolidando-se como uma referência no universo do streaming musical.