Sistema de Música Mscy

Ian Sergio Helfenberger, Nicoly Cristina Ott

Engenharia de Software
Universidade da Região de Joinville (UNIVILLE) – Joinville, SC – Brasil
ian.helfenberger@univille.br, nicoly.ott@univille.br

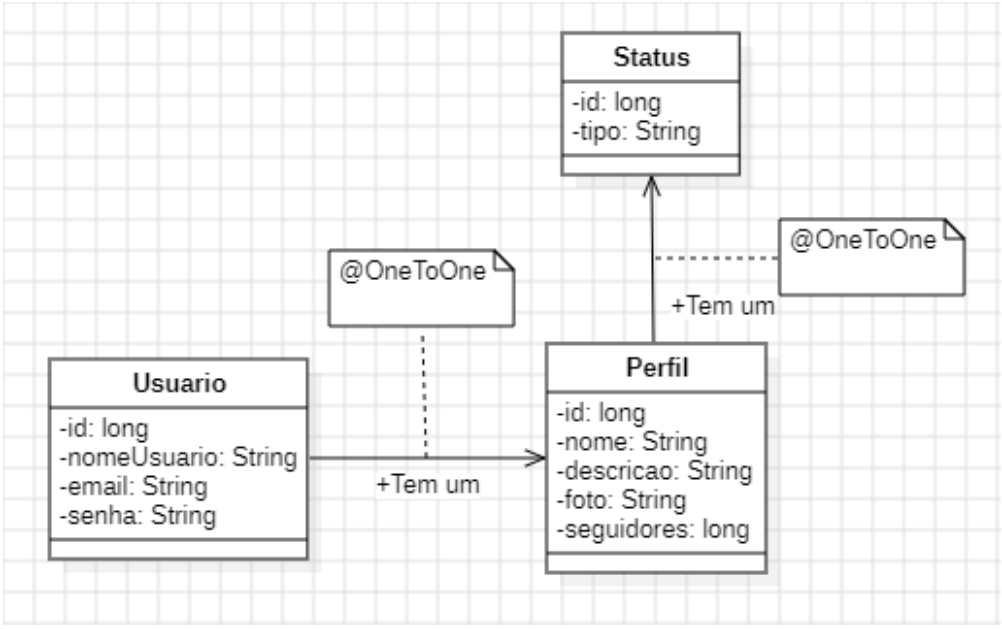
1. Introdução

Aplicativos de música mudaram como as pessoas ouvem músicas, oferecendo acesso fácil e personalizado a milhões de faixas. Com este tipo de streaming em alta, há muitas oportunidades para inovar na forma como os usuários descobrem músicas e criam playlists, melhorando sua experiência. Isso cria espaço para novas funcionalidades, como recomendações inteligentes e compartilhamento social, que podem aumentar o engajamento. Mas inicialmente um aplicativo musical precisa do essencial para seu funcionamento.

2. Requisitos Funcionais

2.1. História de Usuário 01

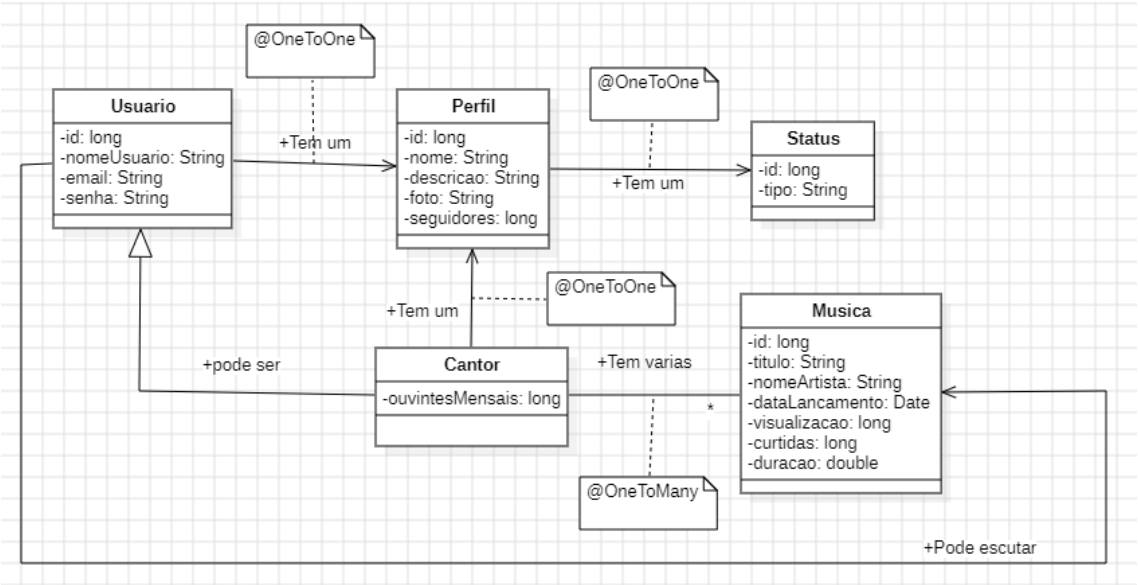
Como um usuário do Sistema de música, eu quero poder me registrar e gerenciar as informações da minha conta, o sistema deve me permitir inserir um nome de usuário para identificação, um e-mail associado à minha conta para realizar o login e uma senha personalizada para acesso a conta, também deve possuir um identificador único e uma personalização de perfil com nome personalizado, uma foto, um status contendo meu tipo de música favorito, seguidores e uma descrição para que as pessoas ao entrarem no perfil saibam meus gostos.



A Figura 01 o diagrama de classe da história de usuário 01. A tabela Usuario tem um Perfil e um Perfil tem um Usuario.

2.2. História de Usuário 02

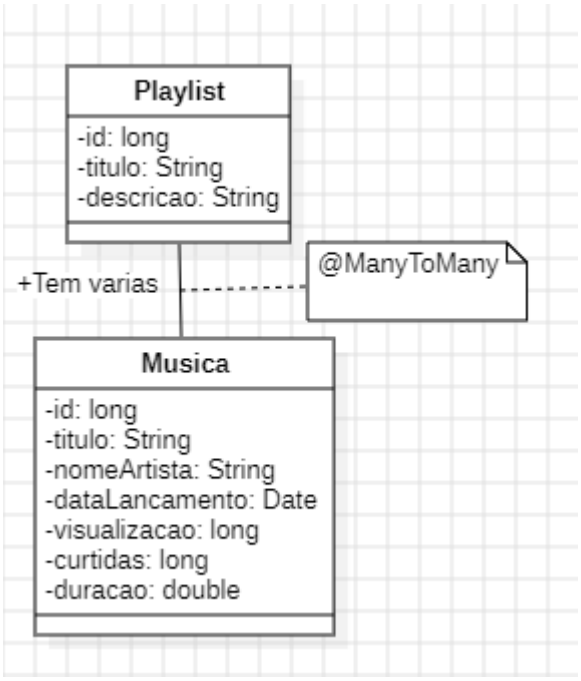
Como um usuário do Sistema de música, eu quero poder navegar pelas músicas e acessar de forma que cada música tenha um identificador único, um título que resume a canção, o nome do artista, a data de lançamento, o número de visualizações, quantidade de curtidas e duração da música, o sistema deve informar o cantor da música e possuir um perfil próprio para os mesmos.



A Figura 02 o diagrama de classe da história de usuário 02. A tabela Cantor tem um Perfil e Cantor tem várias Musica.

2.3. História de Usuário 03

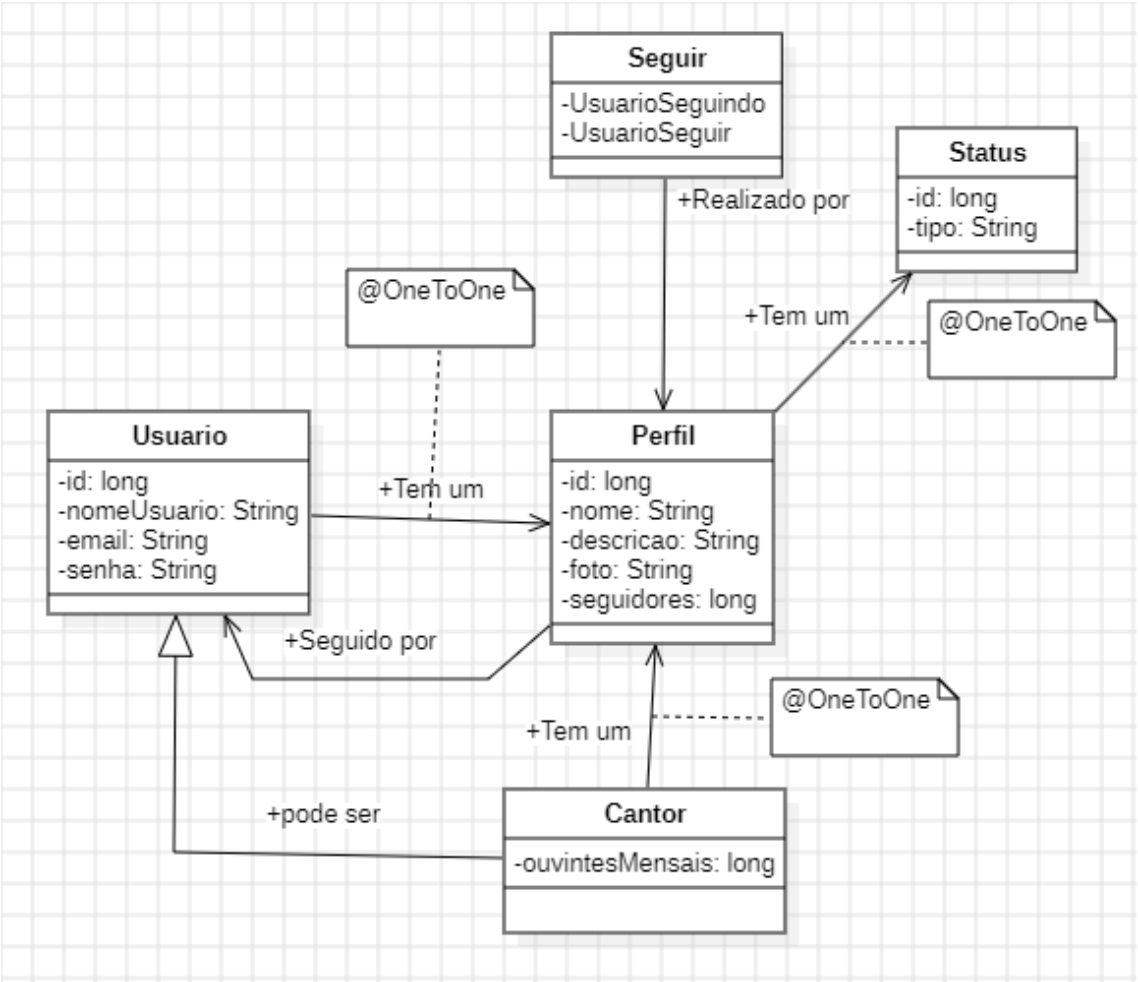
Como um usuário do Sistema de música, eu quero poder criar playlist personalizadas onde cada playlist tenha um identificador único, um título e uma descrição que resume a playlist. Para as playlists quero que elas contenham minhas listas de músicas adicionadas não tendo um limite, quero que quando seja colocado uma música duplicada o sistema confirme que é para adicionar ou retirar não possuindo restrição para duplicidade.



A Figura 03 o diagrama de classe da história de usuário 02. A tabela Playlist tem várias Musica e Musica tem várias Playlist.

2.4. História de Usuário 04

Como um usuário do Sistema de música, eu quero poder seguir outros usuários para ver suas atividades e playlists. E quero poder seguir cantores para acompanhar e receber notificações de lançamentos de músicas novas.



A Figura 04 o diagrama de classe da história de usuário 02. A tabela Seguir Segue um perfil e é seguida por um Usuario.

3. Codificação

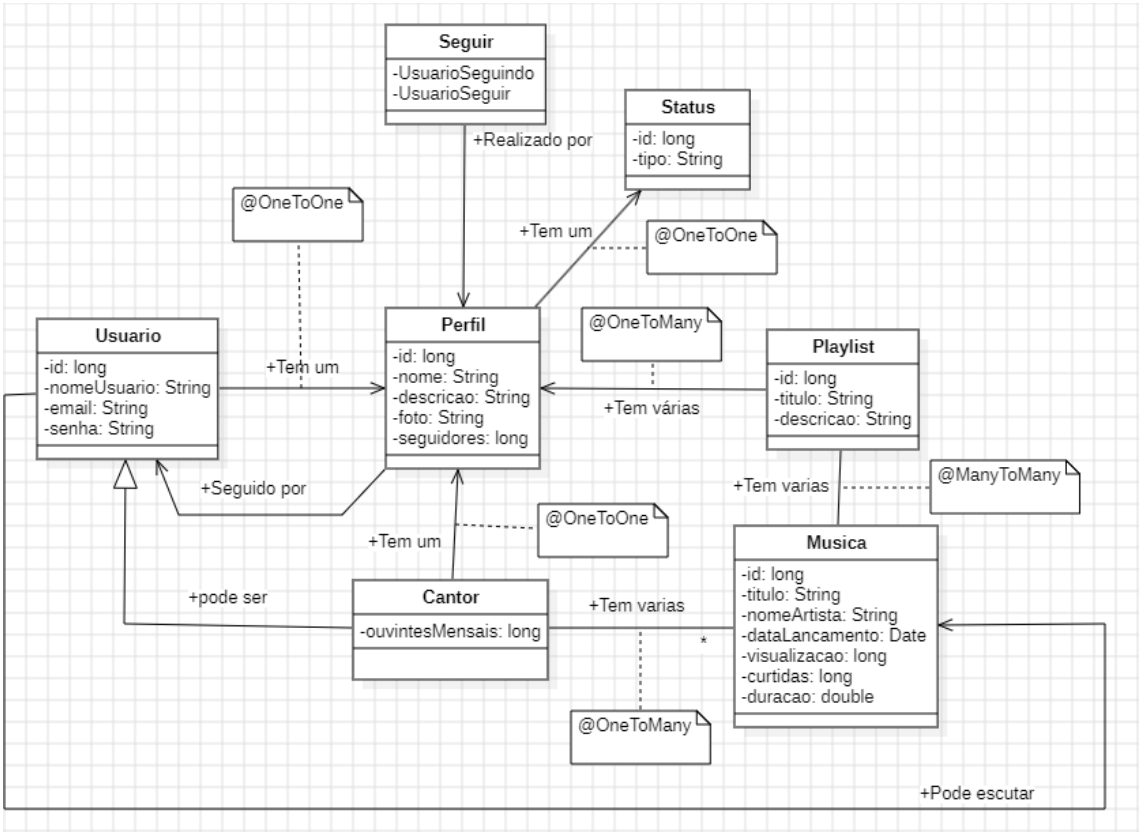


Figura 05. Diagrama de classe do Sistema de música Mscy.

3.1. Entidade Usuario

```java
@Data  1 usage  1 inheritor
@NoArgsConstructor
@Entity
public class Usuario {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private long id;
    private String nome;
    private String email;
    private String senha;

    @OneToOne
    private Perfil perfil;
}
```

Figura 06. Código da entidade Usuario.

3.2. Entidade Cantor

```java
@Data  2 usages
@NoArgsConstructor
@Entity
public class Cantor extends Usuario{

    private long ouvintesMensais;

    @OneToMany(mappedBy = "cantor")
    private List<Musica> musicas;

    @OneToOne
    private Perfil perfil;
}
```

Figura 07. Código da entidade Cantor.

3.3. Entidade Musica

```java
@Data  2 usages
@NoArgsConstructor
@Entity
public class Musica {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private long id;
    private String titulo;
    private long vizualizacao;
    private long curtidas;

    @ManyToOne
    @JoinColumn(name = "cantor_id")
    private Cantor cantor;

    @ManyToMany(mappedBy = "musicas")
    private List<Playlist> playlists;
}
```

Figura 08. Código da entidade Musica.

3.4. Entidade Playlist

```java
@Data  2 usages
@NoArgsConstructor
@Entity
public class Playlist {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private long id;
    private String titulo;
    private String descricao;

    @ManyToMany
    @JoinTable(
            name = "playlist_musica",
            joinColumns = @JoinColumn(name = "playlist_id"),
            inverseJoinColumns = @JoinColumn(name = "musica_id"))
    private List<Musica> musicas;

    @ManyToOne
    @JoinColumn(name = "perfil_id")
    private Perfil perfil;
}
```

Figura 09. Código da entidade Playlist.

3.5. Entidade Perfil

```java
@Data  3 usages
@NoArgsConstructor
@Entity
public class Perfil {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private long id;
    private String nome;
    private String descricao;
    private String foto;
    private String seguidores;

    @OneToOne
    @JoinColumn(name = "status_id")
    private Status status;

    @OneToMany(mappedBy = "perfil")
    private List<Playlist> playlists;

    @OneToOne
    @JoinColumn(name = "cantor_id")
    private Cantor cantor;
}
```

Figura 10. Código da entidade Perfil.

3.6. Entidade Status

```java
@Data  1 usage
@NoArgsConstructor
@Entity
public class Status {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private long id;
    private String tipo;
}
```

Figura 11. Código da entidade Status.

4. Banco de dados

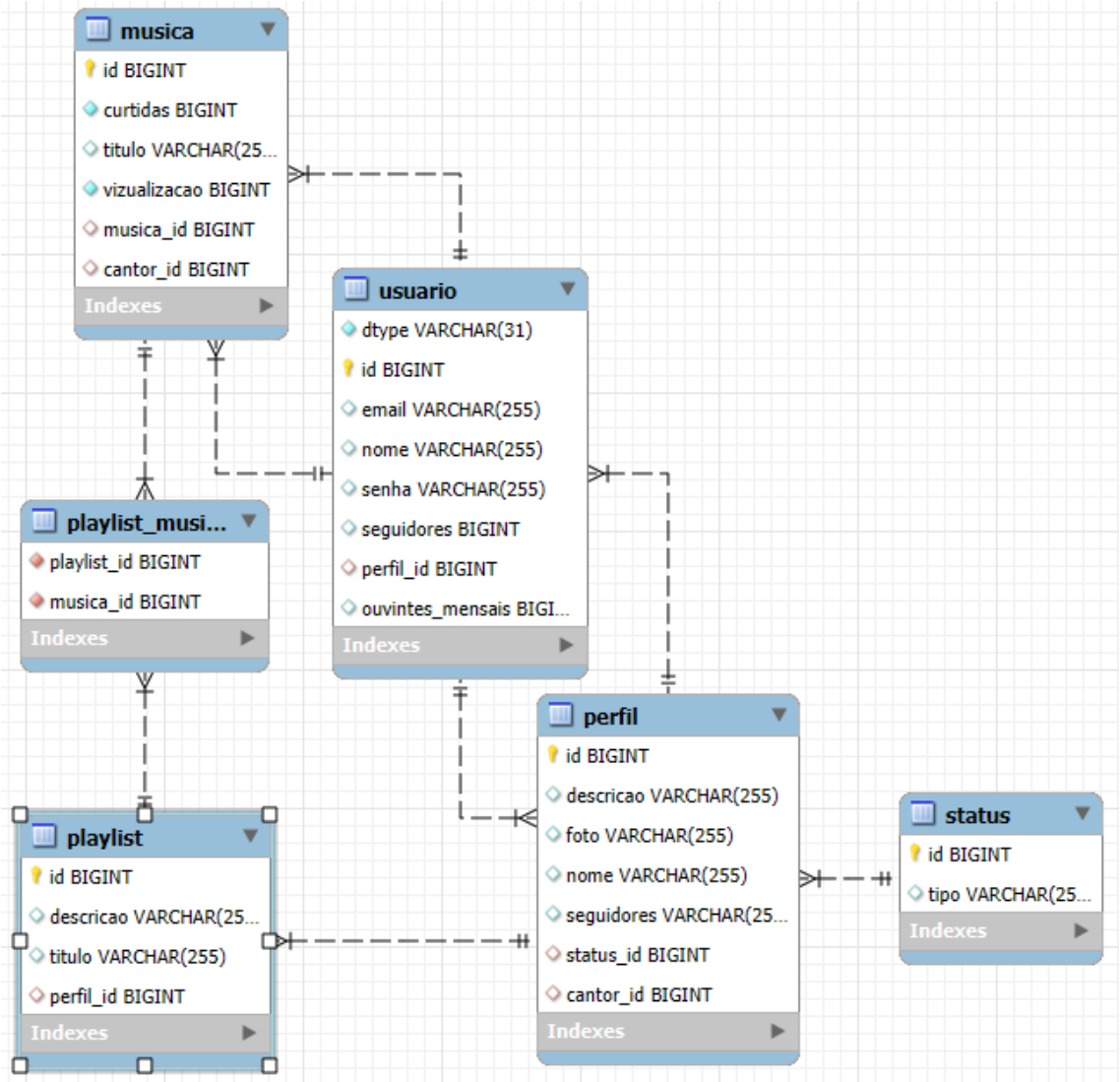


Figura 12. Modelo Entidade Relacionamento do Sistema de Música Mscy.

4. Conclusão

Em resumo, embora inovações como recomendações inteligentes e compartilhamento social possam aumentar o interesse dos usuários, o mais importante para o sucesso de um aplicativo de música é garantir que ele tenha o básico bem-feito. Isso inclui uma interface fácil de usar, acesso rápido a muitas músicas e uma boa ferramenta de busca. A partir dessas funções essenciais, o aplicativo pode crescer e oferecer recursos mais avançados, melhorando a experiência dos usuários.

Referência

BARLOW, J. The Digital Music Revolution: How Streaming Services Changed the Music Industry. Oxford: Oxford University Press, 2020.