

Sistema de Música XPO

Gabriel Fernandes da Costa, Kaue Gomes Cidral

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

1. Introdução

O sistema criado foi um sistema de músicas que tem planos e pode fazer login para salvar os dados, também conta com playlists como músicas e álbuns curtidos(as) e também é possível o download deles.

2. Requisitos Funcionais

As histórias abaixo apresentarão as funcionalidades do software como cadastro, login, ouvir músicas, download de músicas, compartilhar álbuns e criar álbuns

2.1. História de Usuário 01

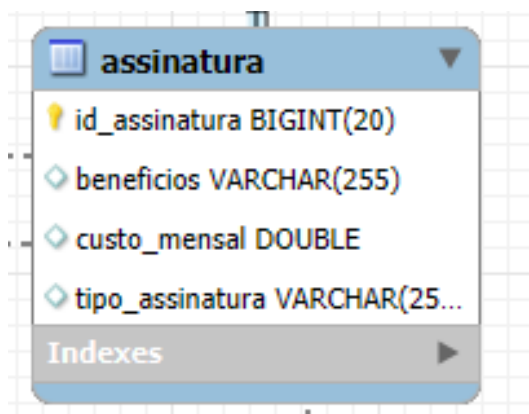
Cenário: A Jornada Musical de Alice no Universo do Streaming

Alice sempre foi apaixonada por música. Desde que ela assinou o serviço de streaming **StreamSound**, sua vida musical nunca mais foi a mesma. Vamos acompanhar um dia típico de Alice explorando o vasto catálogo de músicas, criando playlists, e interagindo com outras funcionalidades do serviço.

1. A Assinatura de Alice

No início, Alice usava a versão gratuita do **StreamSound**, mas logo decidiu fazer um upgrade para o plano **premium**, que oferecia diversas vantagens como a possibilidade de criar **playlists colaborativas**, **baixar músicas offline** e **compartilhar suas playlists** com diferentes permissões.

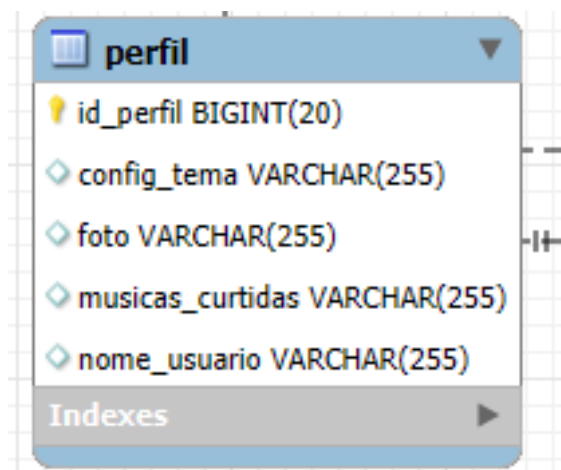
Assinatura
-idAssinatura -tipoAssinatura -custoMensal -beneficios



2. Criando o Perfil de Alice

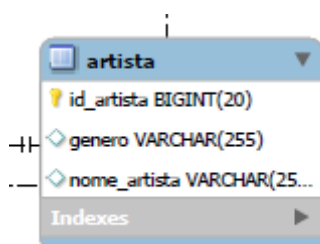
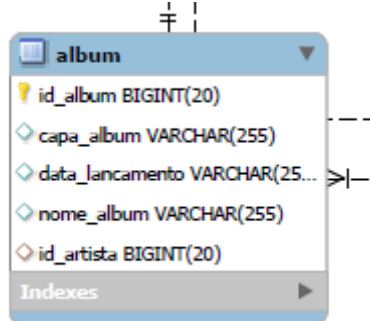
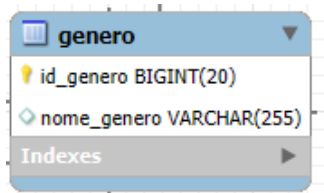
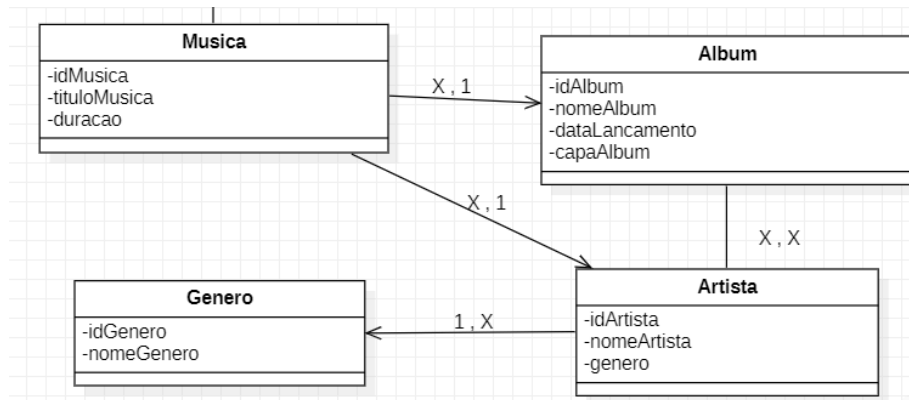
Ao criar sua conta no **StreamSound**, Alice personalizou o seu perfil. Ela escolheu um nome de usuário e uma foto que representassem sua personalidade e configurou o tema do aplicativo para o modo escuro, que ela achava mais agradável.

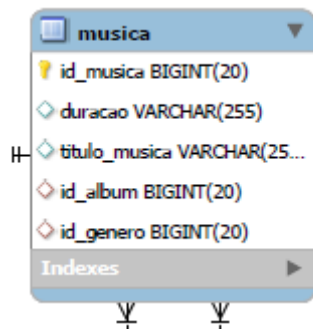
Perfil
-idPerfil -NomeUsuario -Foto -musicasCurtidas -ConfigTema



3. Descobrindo Novos Gêneros e Artistas

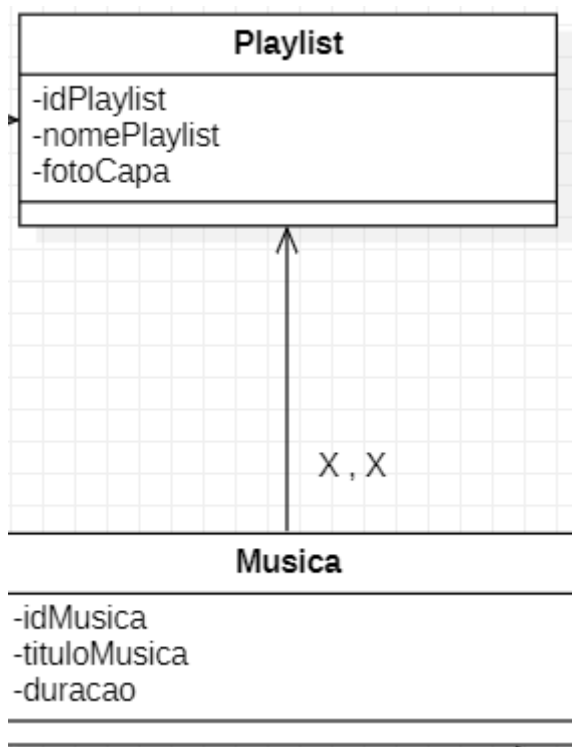
Explorando os **gêneros musicais** disponíveis, Alice descobriu algo diferente: **Lo-fi Hip Hop**, perfeito para momentos de estudo e relaxamento. Essa exploração a levou a um artista chamado **DJ Relax**, que havia lançado um álbum chamado "**Chill Beats for Study**".

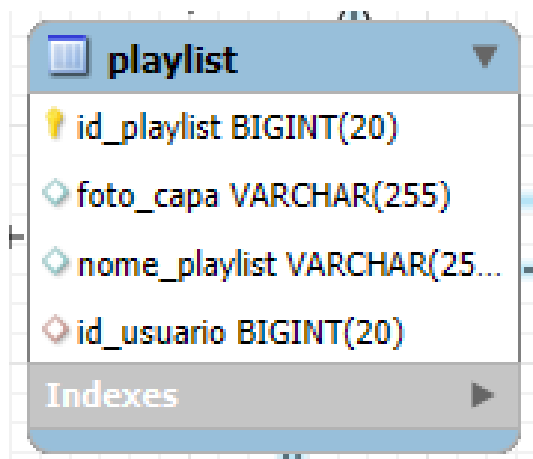




4. Criando uma Playlist para Estudo

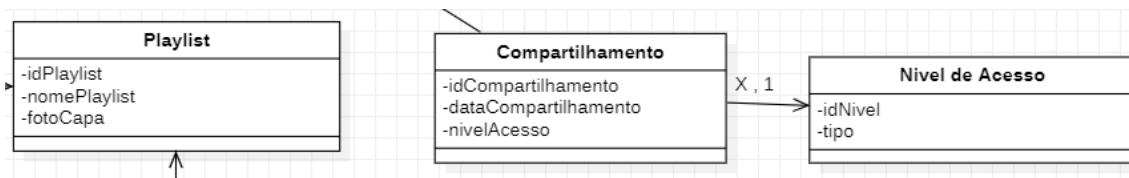
Inspirada por sua nova descoberta, Alice criou uma **playlist** chamada "**Estudo Concentração**". Ela escolheu uma imagem de capa para a playlist e começou a adicionar suas músicas favoritas do álbum recém-descoberto.

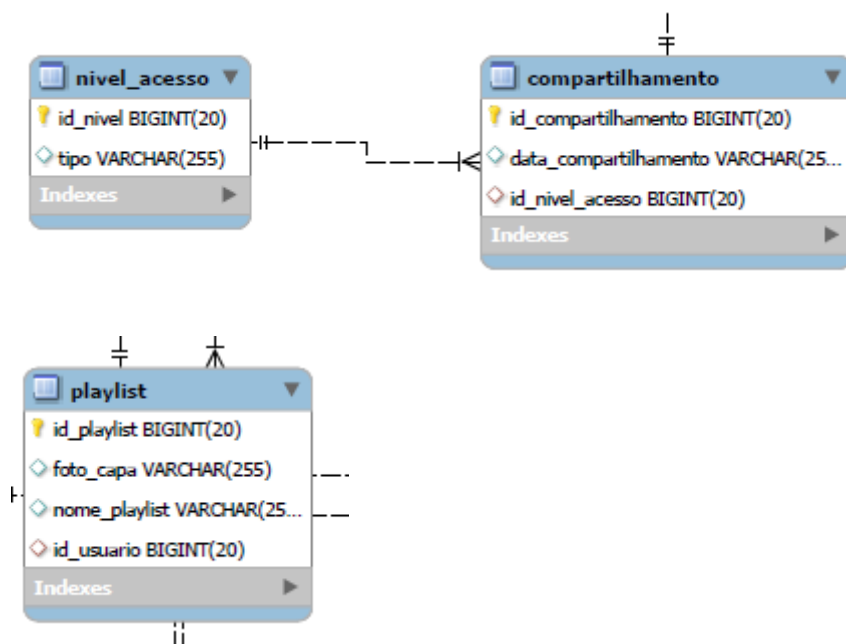




5. Compartilhando a Playlist com Nível de Acesso

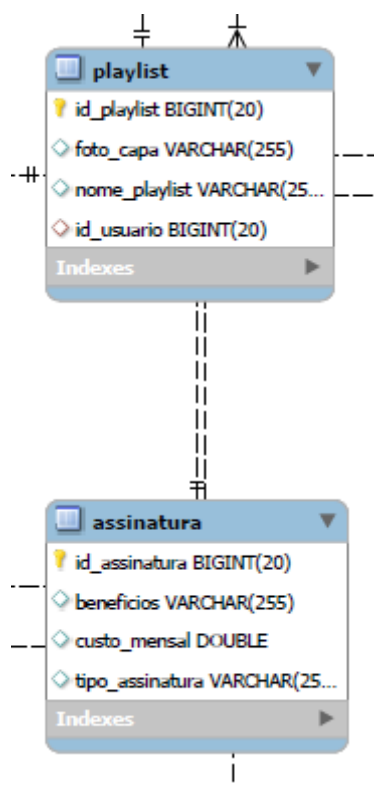
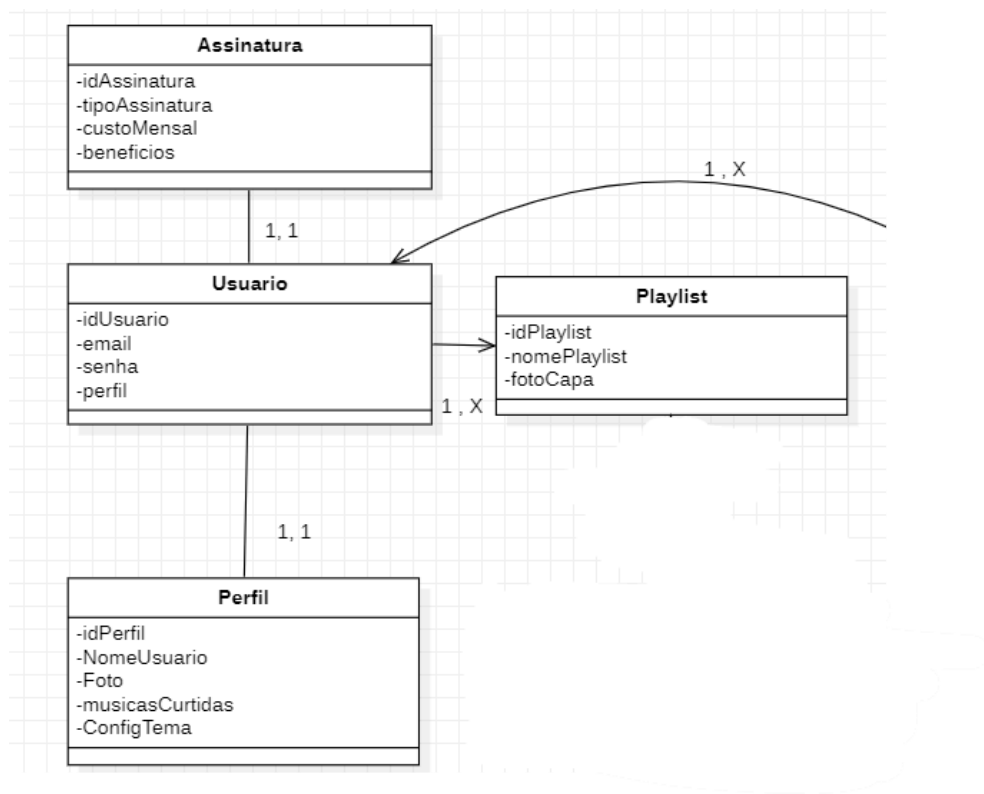
Alice decidiu compartilhar sua nova playlist com seu amigo Lucas, que também estava estudando para provas importantes. Como ela queria que ele colaborasse com mais músicas, Alice permitiu que Lucas também pudesse **editar a playlist**, adicionando ou removendo faixas.

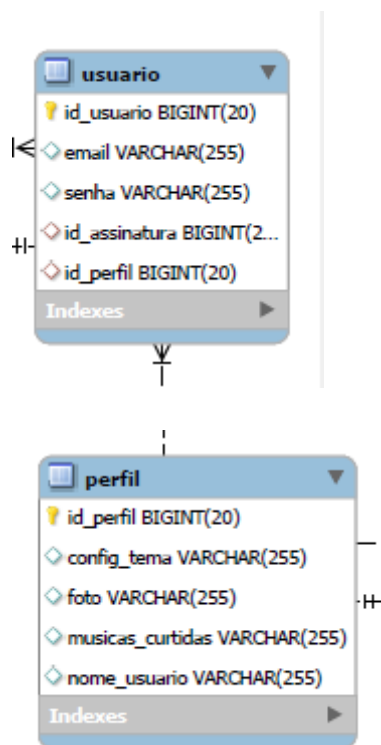




6. Exploração de Outras Funcionalidades

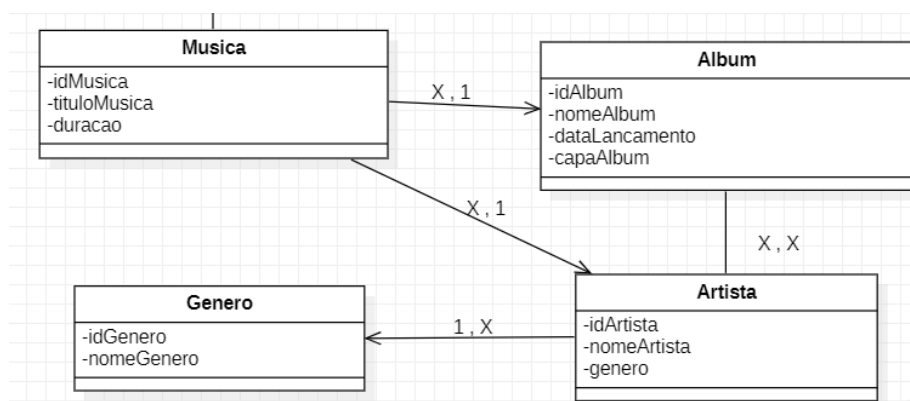
Ao longo dos dias, Alice continuou a explorar mais funcionalidades do **StreamSound**. Ela começou a curtir músicas individuais, que automaticamente foram salvas em uma lista de músicas favoritas. Além disso, Alice baixou suas playlists para escutar offline, uma das grandes vantagens de sua assinatura premium.

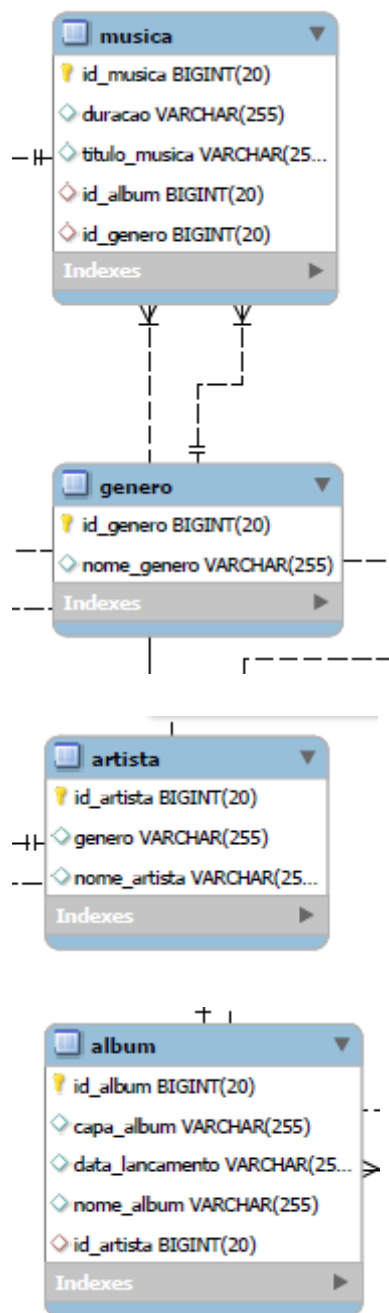




7. Integração com o Álbum e Artistas

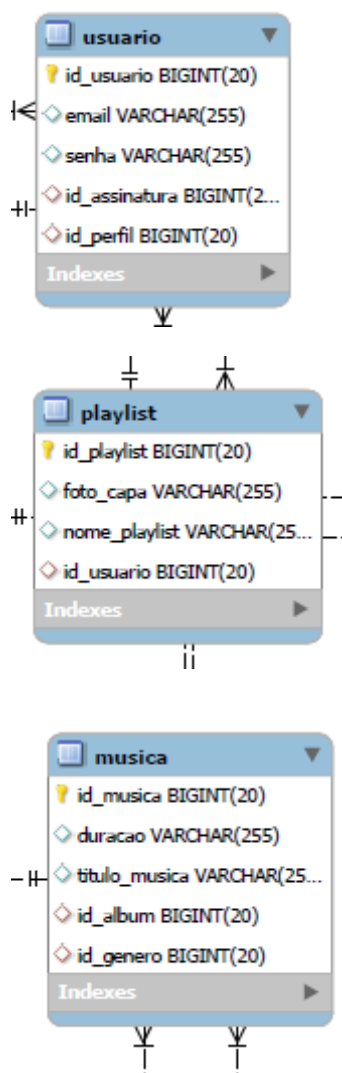
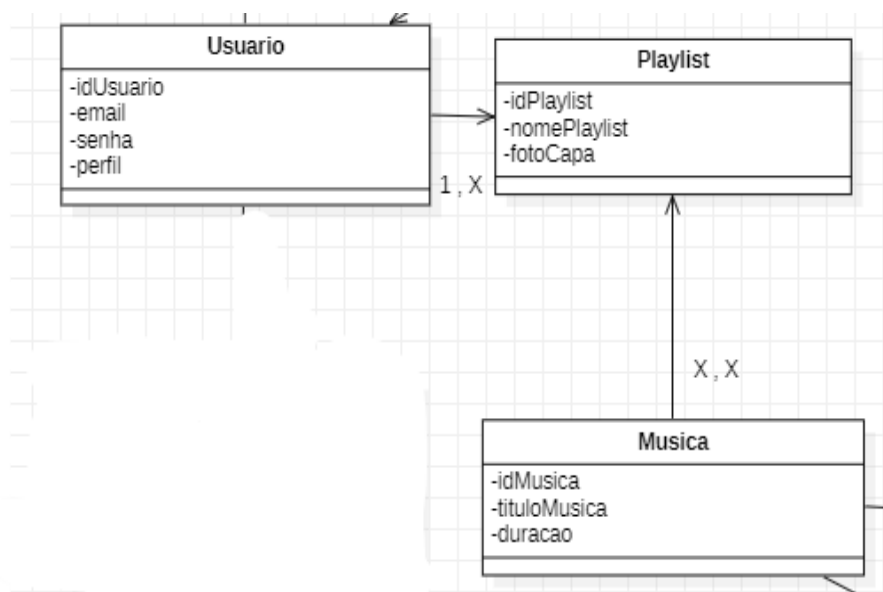
Depois de ouvir o álbum **"Chill Beats for Study"** várias vezes, Alice ficou curiosa sobre o trabalho do **DJ Relax**. Explorando o perfil do artista, ela descobriu outros álbuns e músicas, além de aprender que ele também criava faixas de **Synthwave**, outro gênero que Alice gostava.





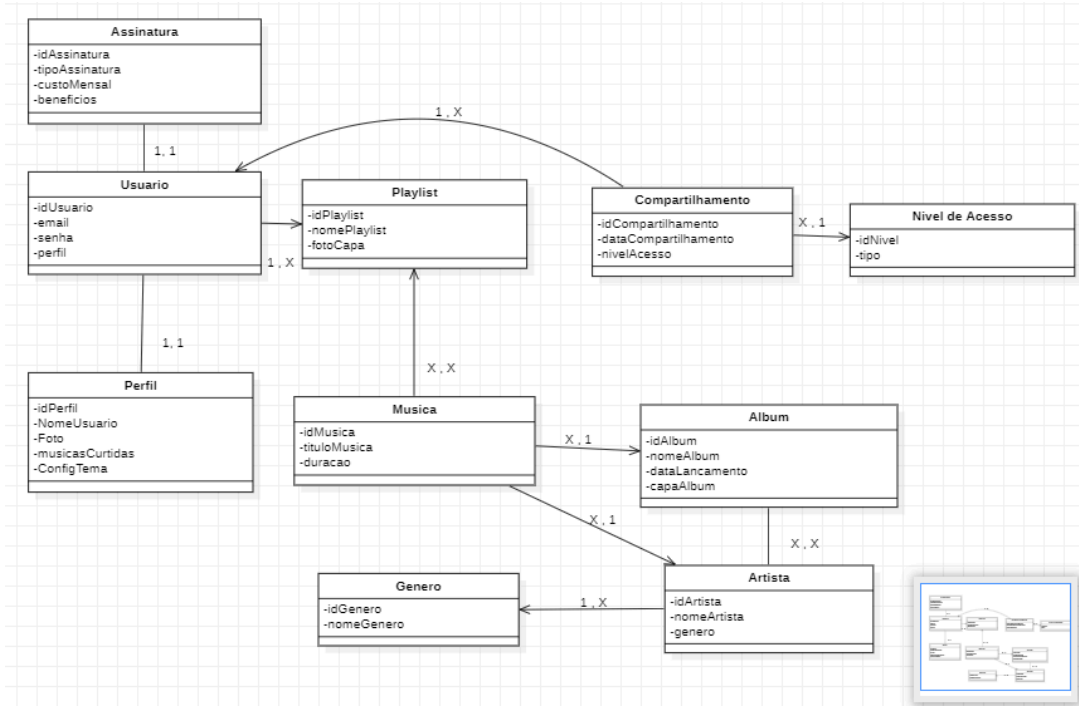
8. Finalizando a Experiência

Alice passou semanas explorando novas músicas, criando playlists personalizadas e colaborando com seus amigos. O **StreamSound** se tornou sua plataforma de streaming favorita. Ao longo do tempo, Alice acumulou uma grande coleção de playlists, cada uma com um tema específico.



3. Codificação

Apresentar as entidades e como realizou os relacionamentos. Apresentar o Diagrama completo em forma de figura XY.



3.1. Entidade XOP

Entidade Album:

```
@Entity
public class Album {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)

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

    @ManyToOne
    @JoinColumn(name = "idArtista")
    private Artista artista;
```

```
}
```

Entidade Artista:

```
@Entity
public class Artista {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)

    @ManyToMany(mappedBy = "artista")
    private List<Album> albums;

}
```

Entidade Assinatura:

```
@Entity
public class Assinatura {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)

    @OneToOne(mappedBy = "assinatura")
    private Usuario usuario;

}
```

Entidade Compartilhamento:

```
@Entity
public class Compartilhamento {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)

    @ManyToOne
```

```

    @JoinColumn(name = "idNivelAcesso")
    private NivelAcesso nivelAcesso;

    @ManyToMany
    @JoinTable(name = "compartilhamento_playlist",
                joinColumns = @JoinColumn(name =
"idCompartilhamento"),
                inverseJoinColumns = @JoinColumn(name =
"idPlaylist"))
    private List<Playlist> playlists;
}

```

Entidade Genero:

```

@Entity
public class Genero {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)

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

}

```

Entidade Musica:

```

@Entity
public class Musica {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)

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

    @ManyToOne

```

```

@JoinColumn(name = "idAlbum")
private Album album;

@ManyToOne
@JoinColumn(name = "idGenero")
private Genero genero;
}

```

Entidade NivelAcesso:

```

@Entity
public class NivelAcesso {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)

    @OneToMany(mappedBy = "nivelAcesso")
    private List<Compartilhamento> compartilhamentos;
}

```

Entidade Perfil:

```

@Entity
public class Perfil {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)

    @OneToOne(mappedBy = "perfil")
    private Usuario usuario;
}

```

Entidade Playlist:

```

@Entity
public class Playlist {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)

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

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

}

```

Entidade Usuario:

```

@Entity
public class Usuario {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)

    @OneToOne(cascade = CascadeType.ALL)
    @JoinColumn(name = "idPerfil")
    private Perfil perfil;

    @OneToMany(mappedBy = "usuario", cascade =
CascadeType.ALL)
    private List<Playlist> playlists;

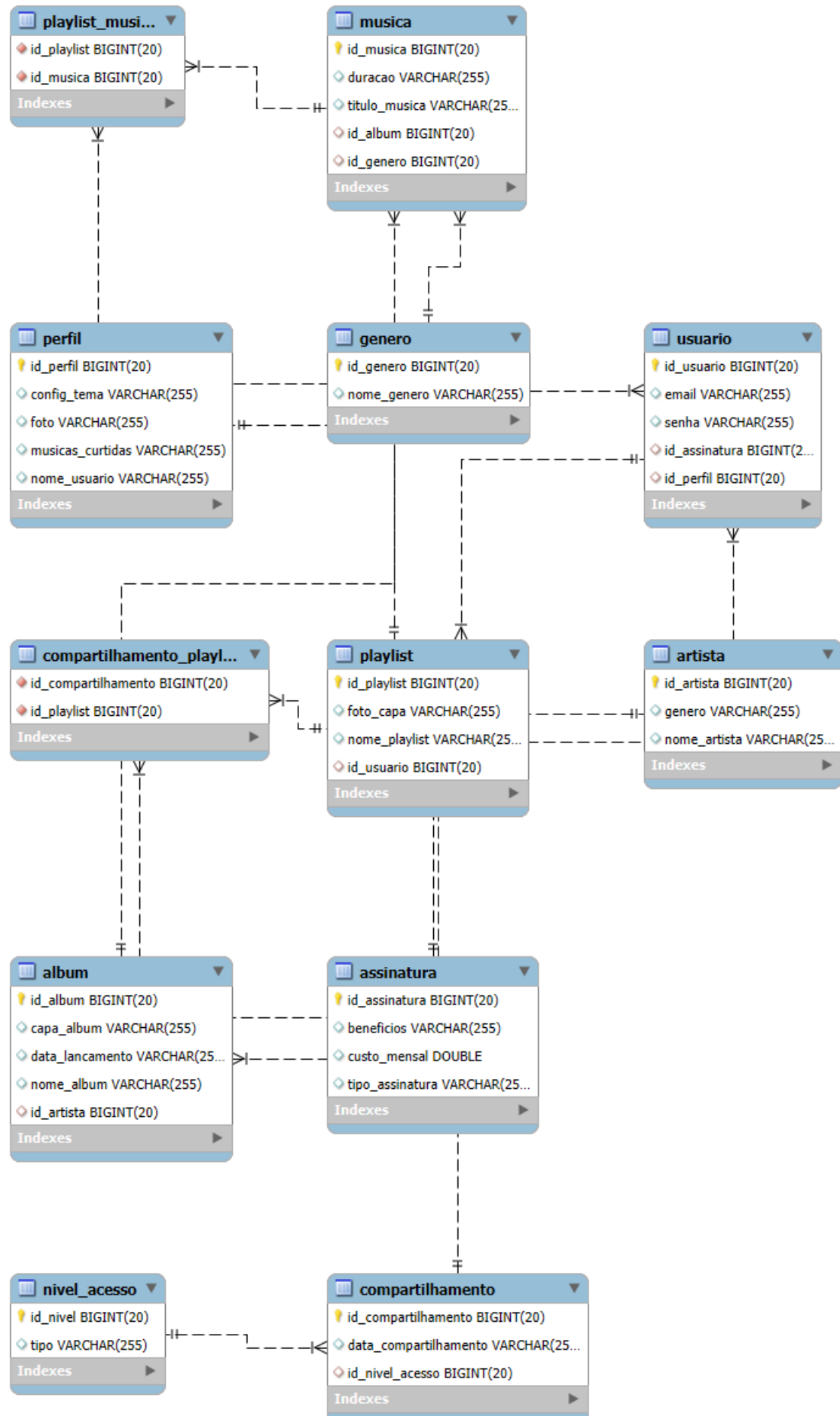
    @OneToOne

```

```
@JoinColumn(name = "idAssinatura")  
private Assinatura assinatura;  
  
}
```

4. Banco de dados

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



4. Conclusão

Através da modelagem desenvolvida, foi possível criar um banco de dados relacional para o sistema de música, abrangendo funcionalidades essenciais como a organização de músicas, playlists, álbuns e perfis de usuários. O diagrama MER apresentado reflete a estrutura de armazenamento e as relações entre entidades fundamentais, como música, artista, gênero e usuário, além de funcionalidades de compartilhamento e assinatura.