

Sistema de Música XPO

Vinicius Aron

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

vinicius.prado@univille.br

1. Introdução

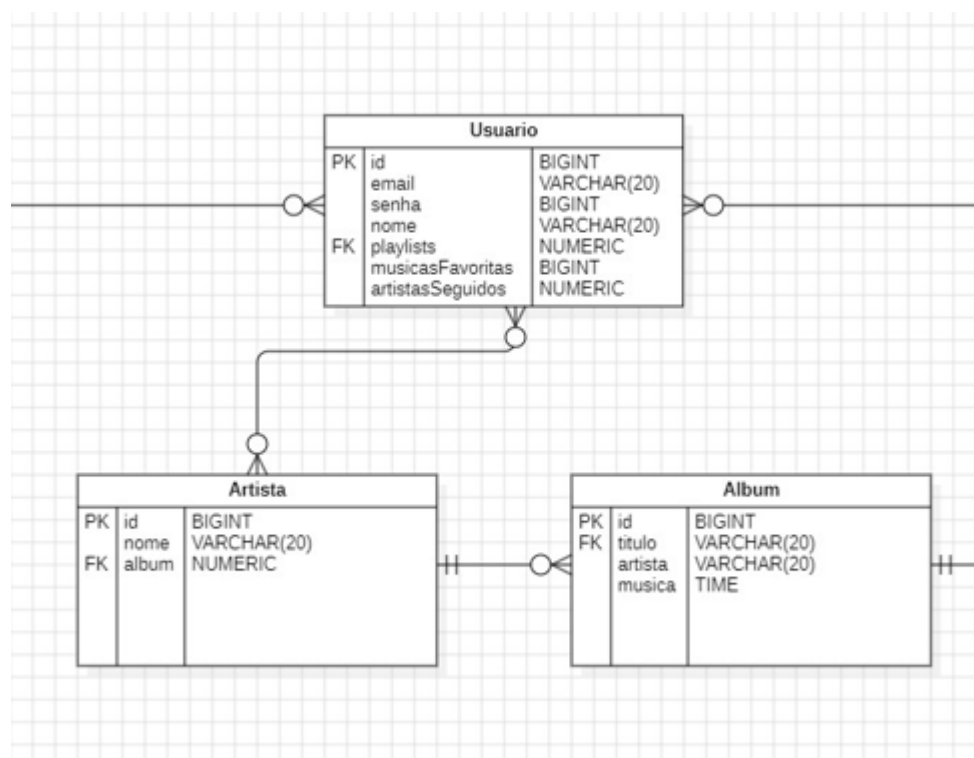
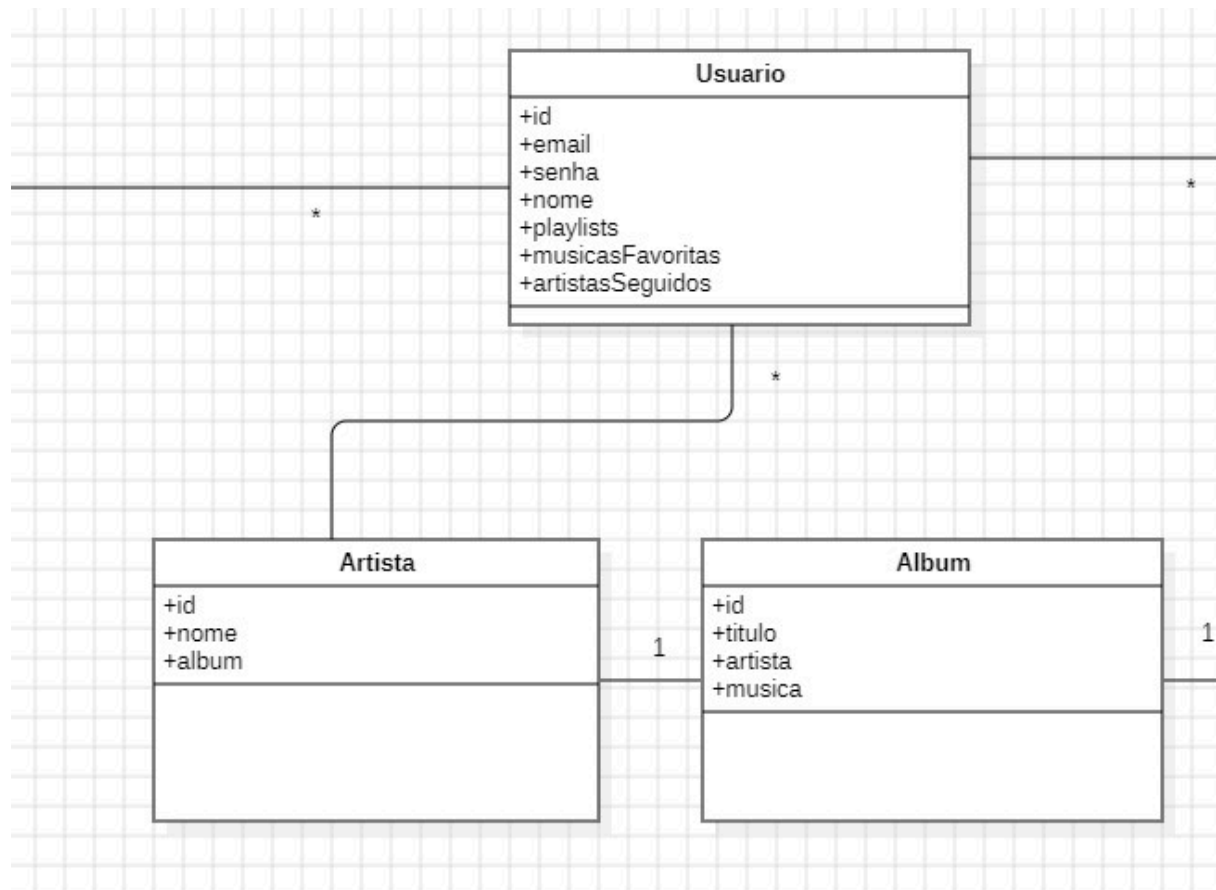
O sistema permite que os usuários explorem músicas, criem playlists personalizadas, sigam artistas, e acessem seu histórico de reprodução. O sistema será implementado sobre um banco de dados relacional baseado em servidor MySQL.

2. Requisitos Funcionais

Este sistema terá várias funcionalidades relacionadas ao gerenciamento de usuários, músicas, playlists e artistas. Cada funcionalidade é descrita sob a forma de histórias de usuários.

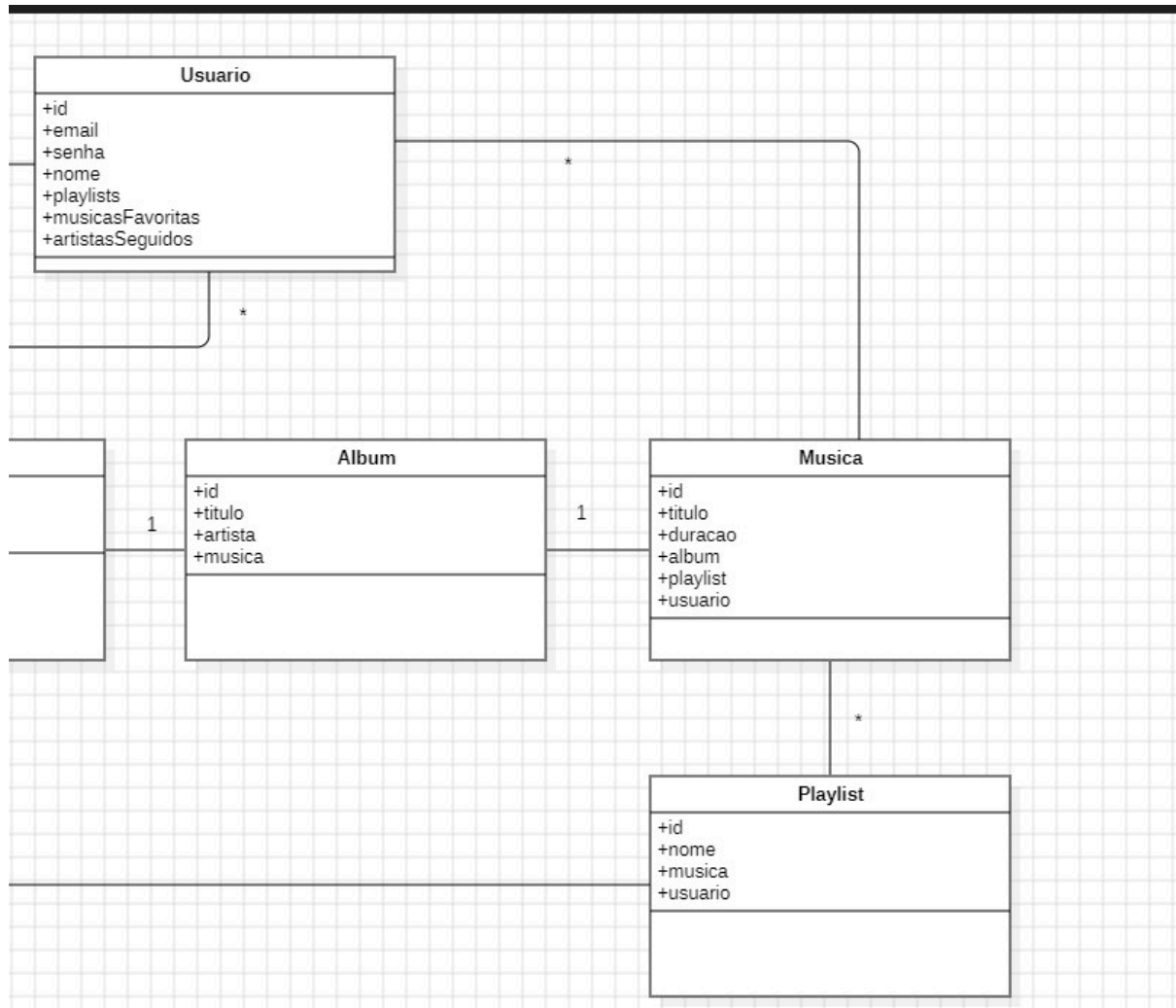
2.1. História de Usuário 01

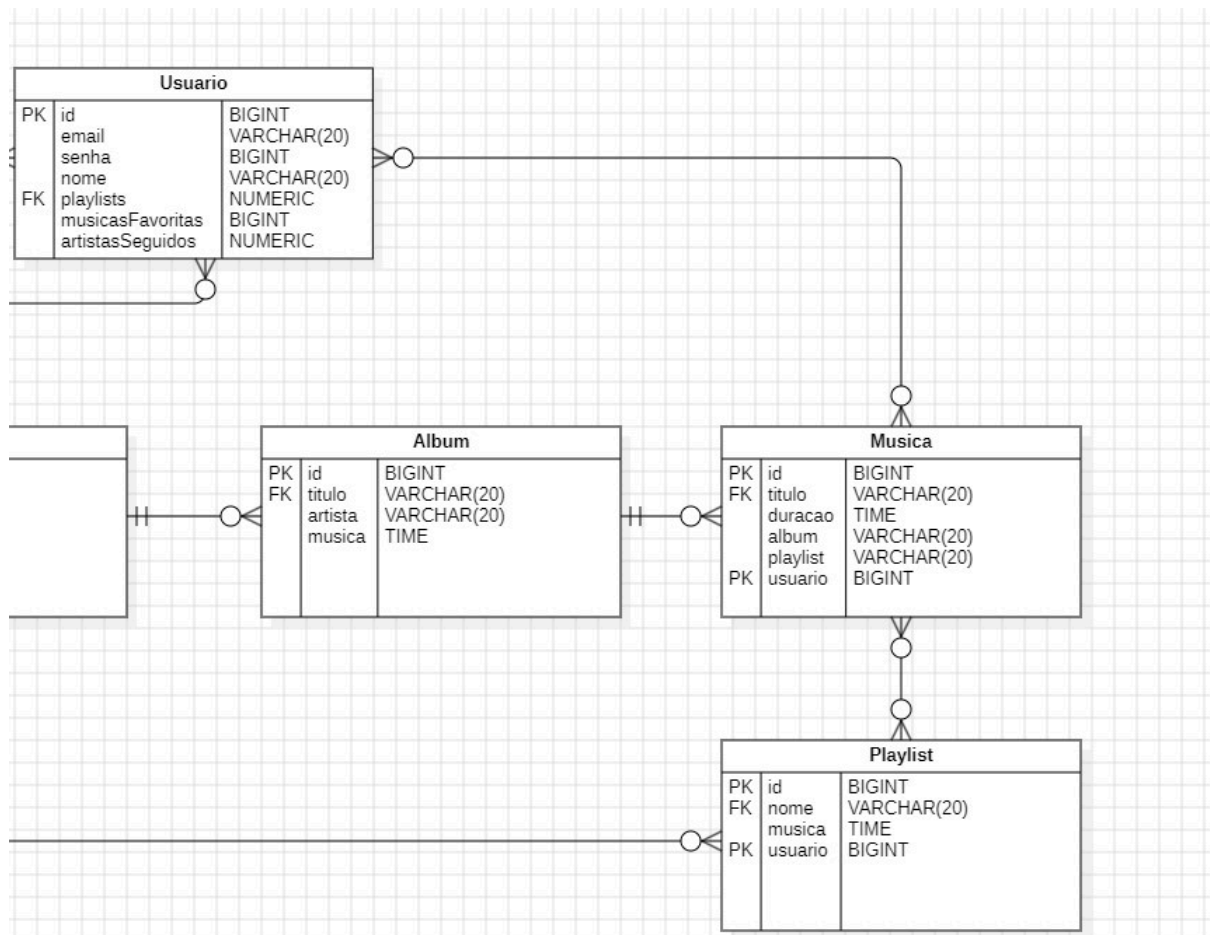
Como usuário, desejo poder acessar o perfil de artistas, acessar álbuns e salvar minhas músicas favoritas.



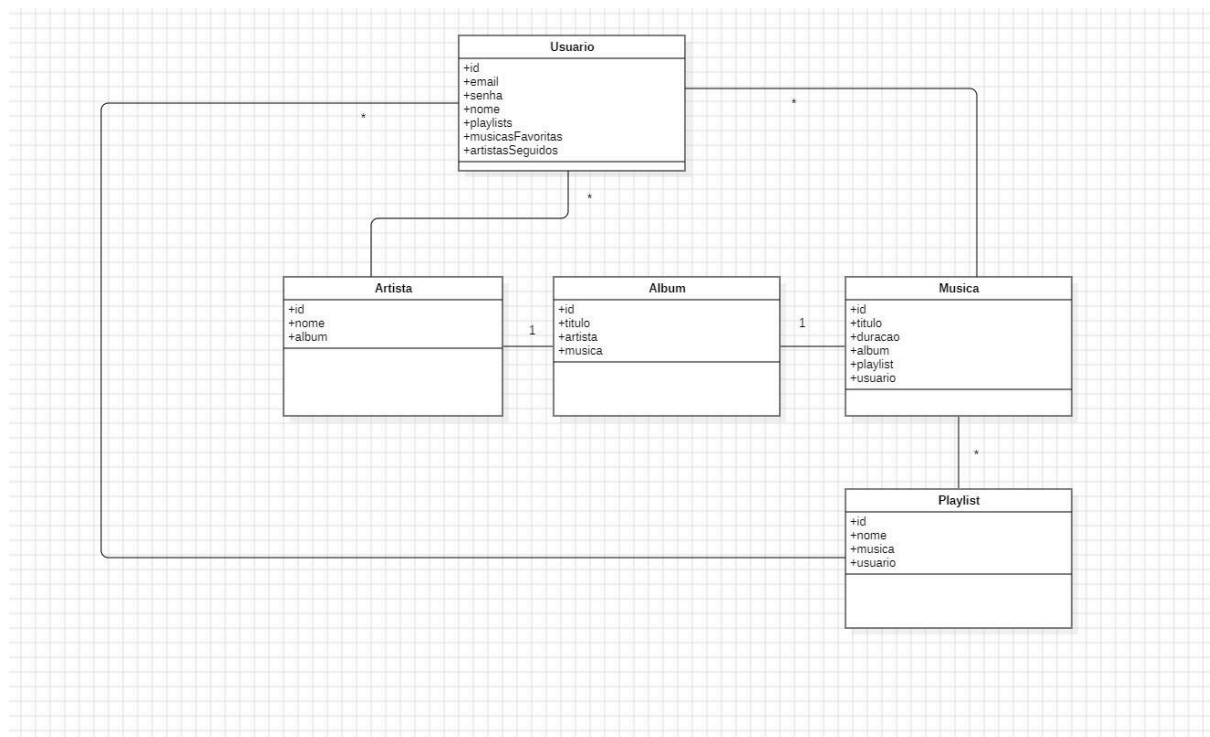
2.2. História de Usuário 02

Como usuário, desejo criar, editar e excluir minhas playlists para organizar minhas músicas e álbuns favoritos.





3. Codificação



3.1. Entidade Usuario

```
package br.edu.univille.br.Relacionamentos.entity;

import br.edu.univille.br.Relacionamentos.entity.Artista;
import br.edu.univille.br.Relacionamentos.entity.Musica;
import jakarta.persistence.*;

import java.util.List;

@Entity 2 usages
public class Usuario {
    @Id no usages
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String email; no usages
    private String senha; no usages
    private String nome; no usages
}
```

3.2. Entidade Playlist

```
package br.edu.univille.br.Relacionamentos.entity;

import br.edu.univille.br.Relacionamentos.entity.Musica;
import jakarta.persistence.*;

import java.util.List;

@Entity 2 usages
public class Playlist {
    @Id no usages
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String nome; no usages

    @ManyToMany no usages
    private List<Musica> musica;

    @ManyToMany(mappedBy = "playlist") no usages
    private List<Usuario> usuario;
}
```

3.3. Entidade Musica

```
import br.edu.univille.br.relacionamentos.entity.Album;
import jakarta.persistence.*;

import java.util.List;

@Entity 5 usages
public class Musica {
    @Id no usages
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String titulo; no usages
    private Integer duracao; // em segundos no usages

    @ManyToOne no usages
    private Album album;

    @ManyToMany(mappedBy = "musicas") no usages
    private List<Playlist> playlists;

    @ManyToMany(mappedBy = "musicasFavoritas") no usages
    private List<Usuario> usuario;
}
```

3.4. Entidade Artista

```

package br.edu.univille.br.relacionamentos.entity;

import br.edu.univille.br.relacionamentos.entity.Album;
import jakarta.persistence.*;

import java.util.List;

@Entity 3 usages
public class Artista {
    @Id no usages
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String nome; no usages

    @OneToMany(mappedBy = "artista") no usages
    private List<Album> albums;
}

```

3.4. Entidade Album

```


package br.edu.univille.br.relacionamentos.entity;

import jakarta.persistence.*;

import java.util.List;

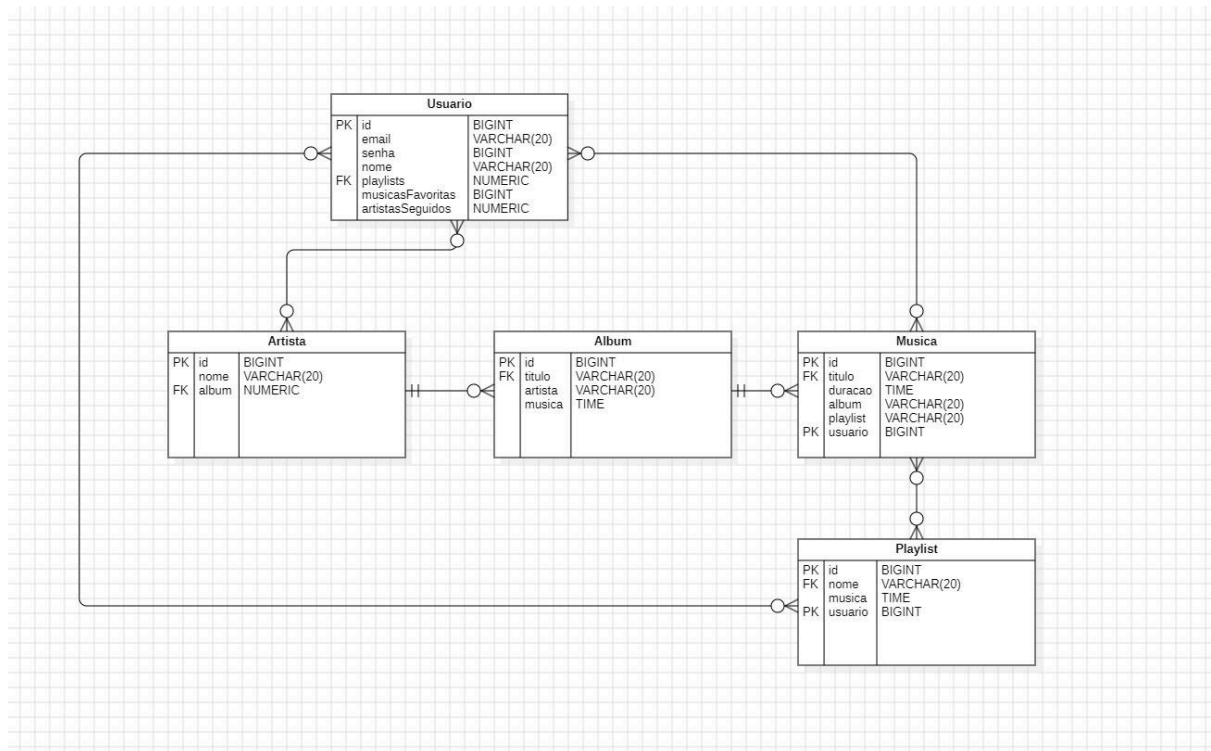
@Entity 4 usages
public class Album {
    @Id no usages
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String titulo; no usages

    @ManyToOne no usages
    private Artista artista;

     @OneToMany(mappedBy = "album") no usages
    private List<Musica> musicas;
}

```

4. Banco de dados



5. Conclusão

O desenvolvimento deste sistema de música oferece uma base sólida para a criação de um serviço similar ao Spotify, onde os usuários podem interagir com músicas, artistas e playlists.