

**UNIVERSIDADE DO VALE DO ITAJAÍ**  
**CIÊNCIAS DA COMPUTAÇÃO**

**LAURA ERIKA KLUGE SCHETINGER**

**TRABALHO M3 DE BANCO DE DADOS**

**SISTEMA DE STREAMING: UNIFY**

**PROFESSOR: MAURÍCIO PASETTO DE FREITAS**

**ITAJAÍ / SC**

**2025**

# SUMÁRIO

## 1. DEFINIÇÃO

.....	3
1.1. FORMULAÇÃO DO PROBLEMA .....	3
1.2. SOLUÇÃO PROPOSTA .....	3

## 2. PROJETO CONCEITUAL

.....	5
-------	---

## 3. PROJETO LÓGICO

.....	6
3.1. PROJETO LÓGICO TEXTUAL NORMALIZADO .....	6
3.2. DIAGRAMA RELACIONAL NORMALIZADO .....	6

## 4. PROJETO FÍSICO

.....	7
4.1. DIAGRAMA RELACIONAL (WORKBENCH) .....	7
4.2. CÓDIGO SQL (DDL) .....	8
4.3. INSERÇÃO DE DADOS (DML) .....	9

## 5. REPOSITÓRIO GITHUB

.....	10
-------	----

## 1. DEFINIÇÃO

## 1.1. FORMULAÇÃO DO PROBLEMA

Atualmente, os ouvintes de músicas enfrentam dificuldades em organizar e centralizar suas bibliotecas de áudio, no qual com a transição das mídias físicas para o formato digital, muitos usuários acabaram com coleções fragmentadas, dispersas entre arquivos locais desorganizados, dispositivos com armazenamento limitado e mídias obsoletas. Essa descentralização impede o acesso rápido e onipresente às faixas desejadas, criando uma experiência de usuário frustrante onde a busca por um álbum ou artista específico torna-se uma tarefa manual e ineficiente, sem a possibilidade de criar listas de reprodução personalizadas ou obter informações detalhadas sobre as obras.

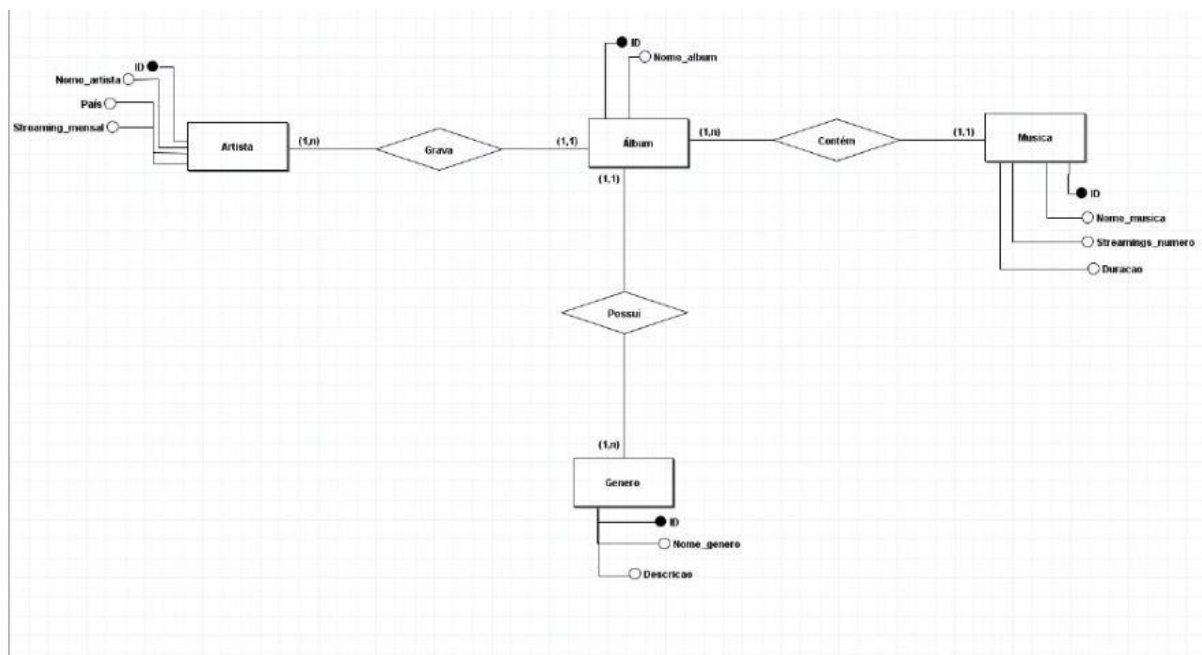
Além disso, existe outro problema no qual desvincula o artista de sua obra pela falta de integridade de dados. Neste caso, sem um sistema relacional estruturado, o cadastro de artistas, lançamentos de álbuns e faixas individuais sofre com redundância de dados e inconsistências, dificultando a catalogação correta de discografias. Esse cenário prejudica tanto o usuário que consome informações erradas, quanto os artistas, que necessitam de uma plataforma confiável para que suas produções sejam armazenadas, catalogadas e recuperadas de forma lógica e hierárquica.

## 1.2. SOLUÇÃO PROPOSTA

Para mitigar os problemas de gestão e acesso ao conteúdo musical, propõe-se o desenvolvimento do **Unify**, um sistema de streaming de áudio suportado por um banco de dados relacional. A solução consiste numa plataforma centralizada que permite o registo estruturado e hierárquico das três entidades fundamentais do domínio: Artistas, Álbuns, Músicas e Gênero. Através desta arquitetura, o sistema garante a integridade referencial dos dados, assegurando que cada faixa musical esteja inequivocamente ligada ao seu álbum de origem e, consequentemente, ao seu artista criador, eliminando redundâncias e inconsistências comuns em armazenamentos amadores.

Além da organização lógica, o Unify implementará funcionalidades de manipulação de dados (CRUD) que permitirão aos administradores e usuários gerir o catálogo de forma dinâmica. A aplicação oferecerá interfaces para cadastrar novos lançamentos, consultar discografias completas, atualizar metadados de faixas e remover conteúdos obsoletos. Desta forma, a solução não apenas resolve o problema do armazenamento disperso, mas também agrega valor ao permitir a recuperação rápida e precisa de informações, servindo como base sólida para futuras expansões, como a criação de playlists personalizadas e sistemas de recomendação.

## **2. PROJETO CONCEITUAL**



### 3. PROJETO LÓGICO

### 3.1. PROJETO LÓGICO TEXTUAL NORMALIZADO

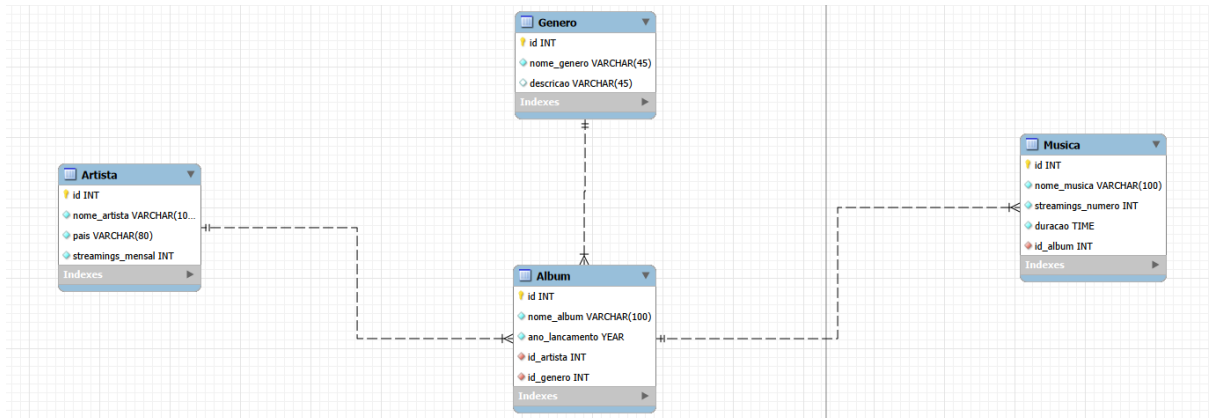
GENERO (id, nome\_genero, descricao)

ARTISTA (id, nome\_artista, pais, streaming\_mensal)

ALBUM (id, nome\_album, ano\_lancamento, id\_artista, id\_genero)

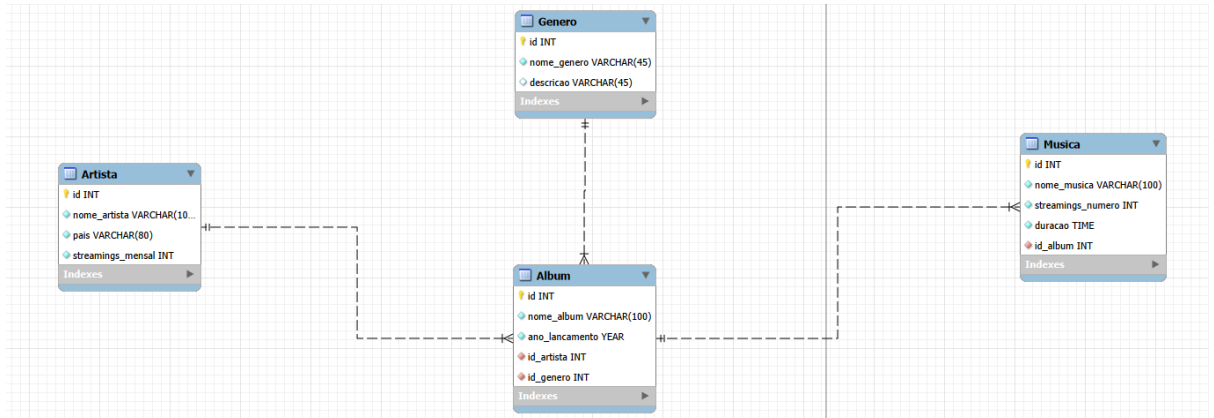
MUSICA (id, nome\_musica, streamings\_numero, duracao, id\_album)

### 3.2. DIAGRAMA RELACIONAL NORMALIZADO



## 4. PROJETO FÍSICO

### 4.1. DIAGRAMA RELACIONAL



### 4.2. CÓDIGO SQL

```

1 • SET @OLD_UNIQUE_CHECKS=@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
2 • SET @OLD_FOREIGN_KEY_CHECKS=@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
3 • SET @OLD_SQL_MODE=@SQL_MODE, SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';
4
5 • CREATE SCHEMA IF NOT EXISTS `unify` DEFAULT CHARACTER SET utf8 ;
6 • USE `unify` ;
7
8 • CREATE TABLE IF NOT EXISTS `unify`.`Artista` (
9     `id` INT NOT NULL AUTO_INCREMENT,
10     `nome_artista` VARCHAR(100) NOT NULL,
11     `pais` VARCHAR(50) NOT NULL,
12     `streamings_mensal` INT NOT NULL,
13     PRIMARY KEY (`id`))
14     ENGINE = InnoDB;
15
16 • CREATE TABLE IF NOT EXISTS `unify`.`Genero` (
17     `id` INT NOT NULL AUTO_INCREMENT,
18     `nome_genero` VARCHAR(45) NOT NULL,
19     `descricao` VARCHAR(45) NULL,
20     PRIMARY KEY (`id`))
21     ENGINE = InnoDB;
22
23 • CREATE TABLE IF NOT EXISTS `unify`.`Album` (
24     `id` INT NOT NULL AUTO_INCREMENT,
25     `nome_album` VARCHAR(100) NOT NULL,
26     `ano_lancamento` YEAR NOT NULL,
27     `id_artista` INT NOT NULL,
28     `id_genero` INT NOT NULL,
29     PRIMARY KEY (`id`),
30     INDEX `album_artista_idx` (`id_artista` ASC) VISIBLE,
31     INDEX `album_genero_idx` (`id_genero` ASC) VISIBLE,
32     CONSTRAINT `album_artista`
33     FOREIGN KEY (`id_artista`)
34     REFERENCES `unify`.`Artista` (`id`)
35     ON DELETE NO ACTION
36     ON UPDATE NO ACTION,
37     CONSTRAINT `album_genero`
38     FOREIGN KEY (`id_genero`)
39     REFERENCES `unify`.`Genero` (`id`)
40     ON DELETE NO ACTION
41     ON UPDATE NO ACTION)
42     ENGINE = InnoDB;
43
44 • CREATE TABLE IF NOT EXISTS `unify`.`Musica` (
45     `id` INT NOT NULL AUTO_INCREMENT,
46     `nome_musica` VARCHAR(100) NOT NULL,
47     `streamings_numero` INT NOT NULL,
48     `duracao` TIME NOT NULL,
49     `id_album` INT NOT NULL,
50     PRIMARY KEY (`id`),
51     INDEX `musica_album_idx` (`id_album` ASC) VISIBLE,
52     CONSTRAINT `musica_album`
53     FOREIGN KEY (`id_album`)
54     REFERENCES `unify`.`Album` (`id`)
55     ON DELETE NO ACTION
56     ON UPDATE NO ACTION)
57     ENGINE = InnoDB;
58
59 • ALTER TABLE `unify`.`Musica`
60     MODIFY COLUMN `streamings_numero` BIGINT;
61
62 • SET SQL_MODE=@OLD_SQL_MODE;
63 • SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
64 • SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
65

```

## 4.3. INSERÇÃO DE DADOS



```

1 • USE `unify`;
2
3 • INSERT INTO `unify`.`Artista` (nome_artista, pais, streamings_mensal) VALUES
4   ('Pink Floyd', 'Inglaterra', 245000000),
5   ('Charli XCX', 'Inglaterra', 327000000),
6   ('Raul Seixas', 'Brasil', 32000000),
7   ('Linkin Park', 'Estados Unidos', 534000000);
8
9 • INSERT INTO `unify`.`Genero` (nome_genero, descricao) VALUES
10  ('Rock', 'Rock clássico, alternativo e pop rock'),
11  ('Pop', 'Música popular comercial e hits'),
12  ('Heavy Metal', 'Numetal e Death Metal'),
13  ('MPB', 'Música Popular Brasileira clássica');
14
15 • INSERT INTO `unify`.`Album` (nome_album, ano_lancamento, id_artista, id_genero) VALUES
16  ('The Wall', 1979, 1, 1),
17  ('BRAT', 2024, 2, 2),
18  ('"Kring-Ha, Bando"', 1973, 3, 4),
19  ('Metora', 2003, 4, 3);
20
21 • INSERT INTO `unify`.`Musica` (nome_musica, streamings_numero, duracao, id_album) VALUES
22  ('Comfortably Numb', 788019458, '00:06:22', 1),
23  ('365', 245568747, '00:03:22', 2),
24  ('Metamorfose Ambulante', 132144934, '00:03:50', 3),
25  ('Numb', 2514936885, '00:03:06', 4);
26

```

## 5. REPOSITÓRIO GITHUB