

# Trabalho 2

Universidade de Brasília  
Departamento de Ciência da Computação  
Bancos de Dados (2024.1)

**Grupo:**

Carlos Cauã Rocha da Silva      Mat: 231034304

Ricardo Martins Fortes      Mat: 190044179

## 1 Introdução

Este relatório busca explicar e estudar o trabalho realizado na disciplina Banco de Dados, onde o objetivo principal foi construir um aplicativo funcional com as características CRUD (Create, Read, Update e Delete) e capaz de prover persistência aos objetos por meio de um banco de dados relacional, de acordo com as especificações definidas pelo professor.

O tema escolhido para o projeto foi um aplicativo de música, capaz de cadastrar novas músicas e salvar playlist criadas pelo usuário, além de outras funcionalidades como criação e edição de diferentes contas de usuário.

## 2 Diagrama Entidade-Relacionamento

O banco de dados na qual o projeto se baseia foi idealizado a partir do diagrama apresentado na figura 1. O diagrama foi criado com a ferramenta brmodelo, e apresenta todas as entidades e seus atributos e relacionamentos.

## 3 Modelo Relacional

A figura 2 apresenta o modelo relacional do projeto, baseado no diagrama anterior, e também gerado com a ferramenta brmodelo.

## 4 Consultas em álgebra relacional

Como especificado, foram feitas cinco consultas ao banco com álgebra relacional

### 4.1 Consulta 1

Esta consulta Seleciona o nome das músicas salvas pelo usuário especificado ( $CodUser = X$ ), e o nome do artista responsável pela música, combinando as tabelas de Musica, MusicasSalvas, ArtistaMusica e Artista.

```
SELECT Musica.Nome AS NomeMusica, Artista.Nome AS NomeArtista
FROM MusicasSalvas
JOIN Musica ON MusicasSalvas.CodMusica = Musica.CodMusica
JOIN ArtistaMusica ON Musica.CodMusica = ArtistaMusica.CodMusica
```

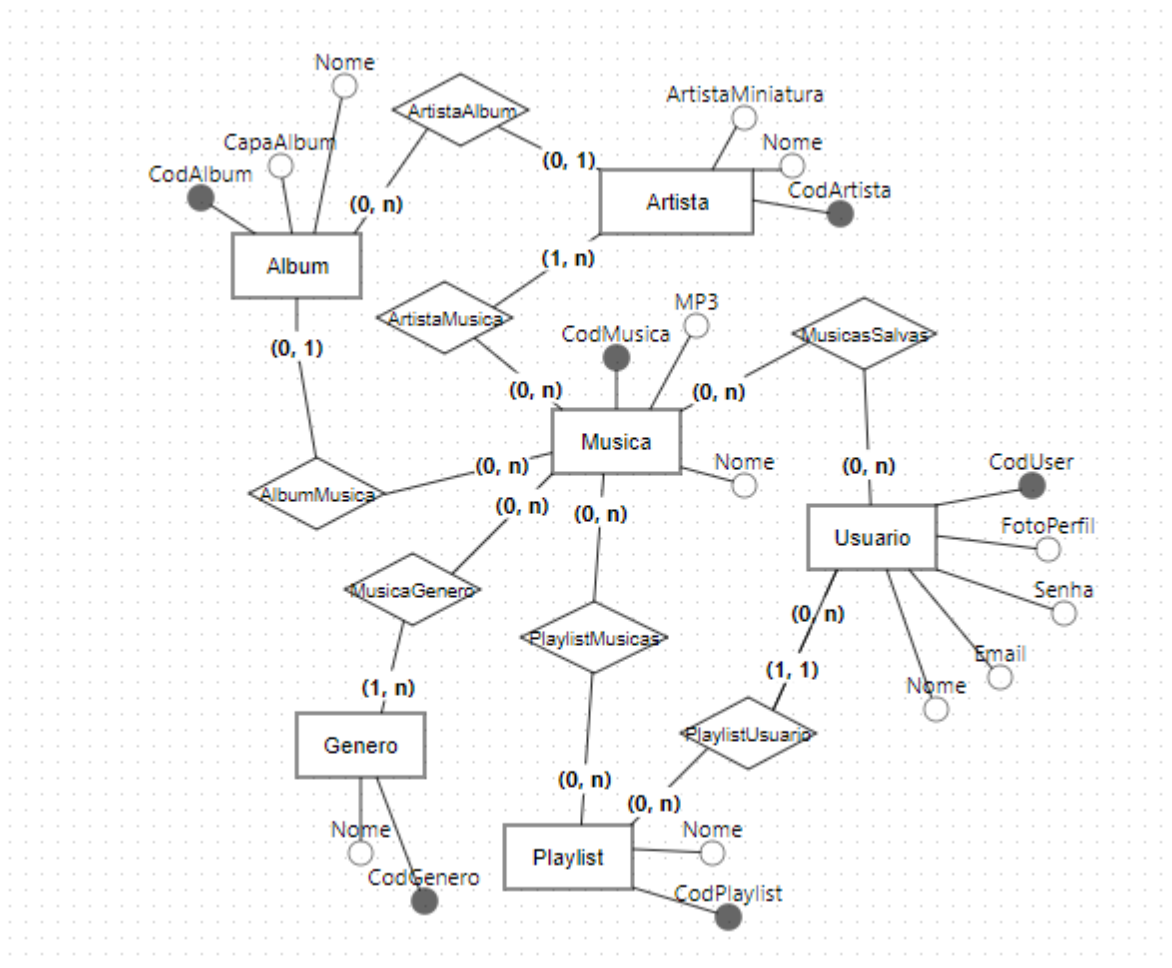


Figura 1: Diagrama Entidade-Relacionamento

```
JOIN Artista ON ArtistaMusica.CodArtista = Artista.CodArtista
WHERE MusicasSalvas.CodUser = X;
```

#### 4.2 Consulta 2

Esta consulta seleciona todas as playlists de um usuário específico ( $\text{CodUser} = X$ ), assim como as músicas contidas nelas.

```
SELECT Playlist.Nome AS NomePlaylist, Musica.Nome AS NomeMusica
FROM PlaylistUsuario
JOIN Playlist ON PlaylistUsuario.CodPlaylist = Playlist.CodPlaylist
JOIN PlaylistMusica ON Playlist.CodPlaylist = PlaylistMusica.CodPlaylist
JOIN Musica ON PlaylistMusica.CodMusica = Musica.CodMusica
WHERE PlaylistUsuario.CodUser = X;
```

#### 4.3 Consulta 3

Esta consulta lista todas as músicas no banco de um determinado gênero ( $\text{CodGenero} = X$ ), assim como o nome do artista relacionado.

```
SELECT Musica.Nome AS NomeMusica, Artista.Nome AS NomeArtista
FROM MusicaGenero
JOIN Musica ON MusicaGenero.CodMusica = Musica.CodMusica
```

```

JOIN ArtistaMusica ON Musica.CodMusica = ArtistaMusica.CodMusica
JOIN Artista ON ArtistaMusica.CodArtista = Artista.CodArtista
WHERE MusicaGenero.CodGenero = X;

```

#### 4.4 Consulta 4

Esta consulta lista os álbuns relacionados ao artista selecionado (CodArtista = X), juntamente com as musicas pertencentes a estes álbuns.

```

SELECT Album.Nome AS NomeAlbum, Musica.Nome AS NomeMusica
FROM Artista
JOIN Album ON Artista.CodArtista = Album.CodArtista
JOIN Musica ON Album.CodAlbum = Musica.CodAlbum
WHERE Artista.CodArtista = X;

```

#### 4.5 Consulta 5

Esta consulta lista todas as músicas contidas em uma playliost específica (CodPlaylist = X), assim como as informações associadas, nome do artista, código da música e nome da Playlist listada.

```

SELECT M.Nome,A.Nome,M.CodMusica,P.Nome
FROM PlaylistMusica as PM
INNER JOIN Musica as M ON PM.CodMusica = M.CodMusica
INNER JOIN ArtistaMusica as AM ON AM.CodMusica = M.CodMusica
INNER JOIN Artista as A ON A.CodArtista = AM.CodArtista
INNER JOIN Playlist as P ON PM.CodPlaylist = P.CodPlaylist
WHERE PM.CodPlaylist = X;

```

### 5 Avaliação de formas normais

Para manter o banco de dados bem estruturado, todas as tabelas presentes no projeto estão na 3ª forma normal, segundo as definições:

- 1ª Forma Normal: Todos os atributos são valores atômicos.
- 2ª Forma Normal: Todos os atributos devem depender inteiramente da chave primária.
- 3ª Forma Normal: Não podem haver dependências transitivas entre atributos não chave.

Como solicitado, as seções 5.1-5.5 apresentam exemplos de tabelas presentes no projeto.

#### 5.1 Tabela 1: Artista

Forma Normal: 3F. (Figura: 3)

#### 5.2 Tabela 2: Genero

Forma Normal: 3F. (Figura: 4)

#### 5.3 Tabela 3: Musica

Forma Normal: 3F. (Figura: 5)

#### 5.4 Tabela 4: Playlist

Forma Normal: 3F. (Figura: 6)

## 5.5 Tabela 5: Usuario

Forma Normal: 3F. (Figura: 7)

## 6 Camada de persistência

O Projeto foi implementado utilizando Flask e html para realizar a interface gráfica com o usuário, e o SGBD utilizado foi o MySQL.

Para realizar a integração entre o banco de dados e o projeto, foram criados os módulos **modArtistas.py**, **modContas.py**, **modMusica.py** e **modPlaylist.py**, que são responsáveis por receber requisições, acessar o banco de dados, e retornar as informações lidas, ou modifica-las.

Esses módulos fazem uso da função de controle de requisições denominada **executeQuery**, definida no módulo **bancoDef.py**, que recebe como parâmetros o texto SQL a ser executado, e seus possíveis parâmetros, como nome de artista ou id de usuário, por exemplo. Por sua vez, o módulo **mysql.connector** é invocado em toda chamada dessa função, que abre a conexão com o banco de dados, a partir das suas credenciais (DB\_HOST, DB\_USER, DB\_PASSWORD, DB\_NAME).

Todos os arquivos utilizados no projeto, assim como os scripts SQL utilizados para montar o banco estão disponíveis em: <https://github.com/Schatten900/TrabalhoBD/tree/main>.

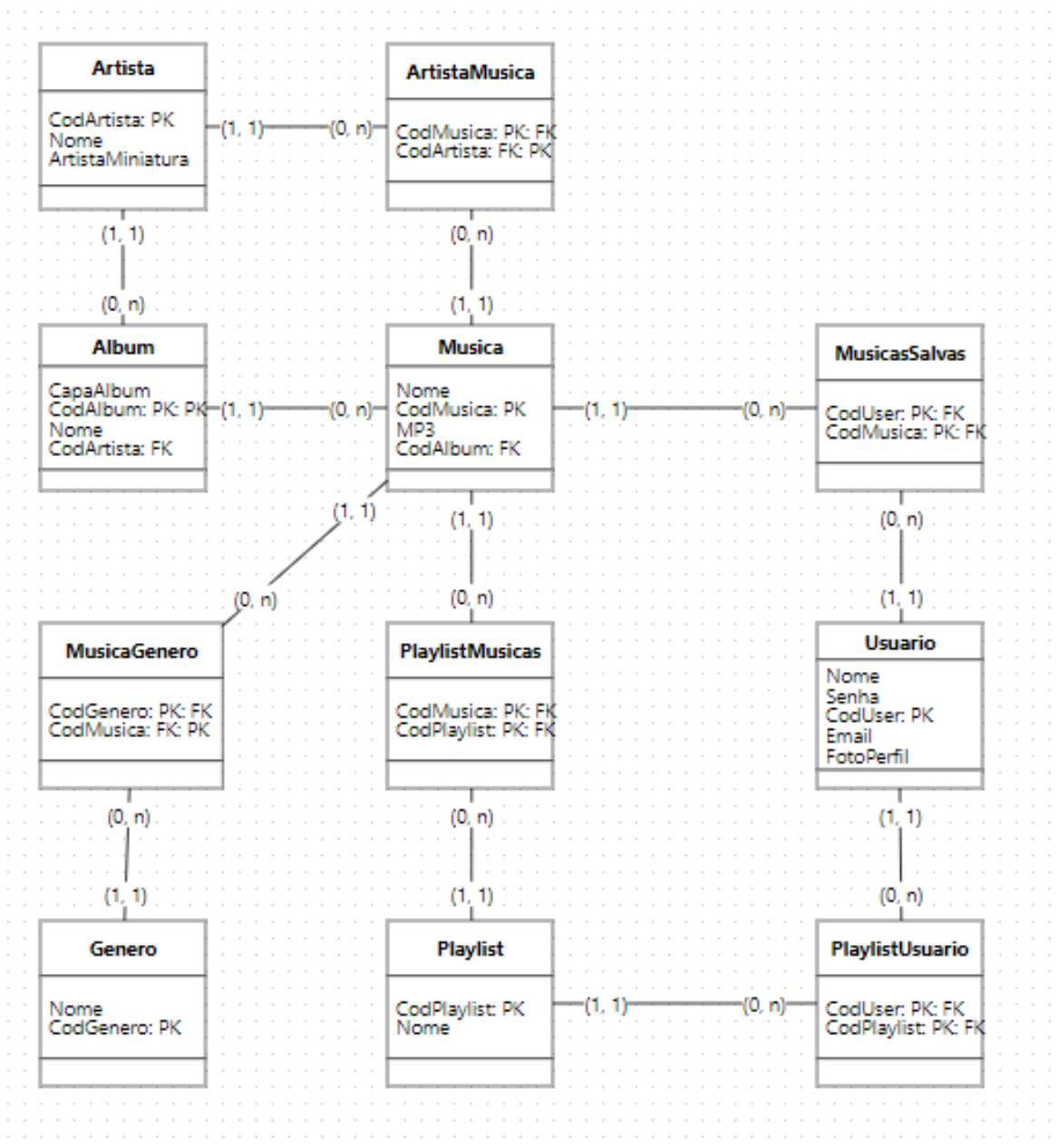


Figura 2: Modelo Relacional

|   | CodArtista | Nome            | ArtistaMiniatura |
|---|------------|-----------------|------------------|
| ▶ | 1          | Imagine Dragons | BLOB             |
|   | 2          | Pentakill       | BLOB             |
|   | 3          | K/DA            | BLOB             |
|   | 4          | Daft Punk       | BLOB             |
|   | 5          | Toby Fox        | BLOB             |
|   | 6          | Bon Jovi        | BLOB             |
| * | NULL       | NULL            | NULL             |

Figura 3: Instâncias de artistas no banco

|   | CodGenero | Nome       |
|---|-----------|------------|
| ▶ | 1         | Rock       |
|   | 2         | Pop        |
|   | 3         | Rap        |
|   | 4         | Metal      |
|   | 5         | Eletronica |
| * | NULL      | NULL       |

Figura 4: Instâncias de generos no banco

|   | CodMusica | CodAlbum | Nome                     | MP3  |
|---|-----------|----------|--------------------------|------|
| ▶ | 1         | 1        | Bones                    | BLOB |
|   | 2         | 2        | Deathfire Grasp          | BLOB |
|   | 3         | 3        | Drum Go Dum              | BLOB |
|   | 5         | 4        | Ruins                    | BLOB |
|   | 6         | 5        | You Give Love A Bad Name | BLOB |
| * | NULL      | NULL     | NULL                     | NULL |

Figura 5: Instâncias de musicas no banco

|   | CodPlaylist | Nome              |
|---|-------------|-------------------|
| ▶ | 1           | Teste 1           |
|   | 2           | Teste2            |
|   | 3           | Teste3            |
|   | 4           | Teste4            |
|   | 5           | Teste5            |
|   | 6           | Best of Pentakill |
| * | NULL        | NULL              |

Figura 6: Instâncias de playlists no banco

|   | CodUser | Nome           | Senha  | Email                 | FotoPerfil |
|---|---------|----------------|--------|-----------------------|------------|
| ▶ | 1       | Nome           | 135790 | nome@email.com        | BLOB       |
|   | 2       | Fulano         | 135790 | fulano@email.com      | NULL       |
|   | 3       | Bob Esponja    | 097531 | bob@fendadobikini.com | NULL       |
|   | 4       | Patrick        | 097531 | pat@pedra.com         | NULL       |
|   | 5       | Seu Sirigueijo | 135790 | siri@gmail.com        | NULL       |
| * | NULL    | NULL           | NULL   | NULL                  | NULL       |

Figura 7: Instâncias de usuarios no banco