



SERVIÇO NACIONAL DE APRENDIZAGEM INDUSTRIAL

SENAI “GASPAR RICARDO JUNIOR”

Curso

**TÉCNICO EM DESENVOLVIMENTO
DE SISTEMAS**

**Relatório Comparativo para Plataforma de
Streaming de Música**

Nycolle Kauany Caetano Barbosa

**Sorocaba
Novembro – 2024**



SERVIÇO NACIONAL DE APRENDIZAGEM INDUSTRIAL

SENAI “GASPAR RICARDO JUNIOR”

Nycolle Kauany Caetano Barbosa

Relatório Comparativo para Plataforma de Streaming de Música

Relatório Comparativo de uma
plataforma de streaming de
música

Prof. Emerson Magalhães

Sorocaba
Novembro – 2024

HISTÓRICO DE VERSÕES

[illegible]

SUMÁRIO

1.	RELATORIO COMPARATIVO	4
1.1.	Relacionais (MySQL).....	4
1.2.	Não-Relacionais (MongoDB).....	4
1.3.	Justificativa da Escolha	4
2.	CONFIGURAÇÃO DO AMBIENTE	6
3.	DIAGRAMA DE MODELAGEM	7
3.1.	(MER).....	7
3.2.	(DER)	7
4.	BANCO DE DADOS NORMALIZADO	8
5.	DICIONÁRIO DE DADOS	9

Relatório Comparativo para Plataforma de Streaming de Música

1. RELATORIO COMPARATIVO

1.1. Relacionais (MySQL)

- Estrutura: Utiliza tabelas para armazenar dados.
- Modelo de dados: Estruturado, com relações definidas por chaves primarias e estrangeiras.
- Integridade: Garante integridade referencial e ACID (Atomicidade, Consistência, Isolamento, Durabilidade).
- Exemplos: Dados transacionais, como registros de usuários, playlists e transições financeiras.

1.2. Não-Relacionais (MongoDB)

- Estrutura: Utiliza coleções de documentos (JSON/BSON).
- Modelo de dados: Flexível, permitindo esquemas variados e aninhados.
- Escalabilidade: Horizontal, ideal para grandes volumes de dados não estruturados.
- Linguagem de Consulta: Utiliza consultas JSON.
- Exemplos: Histórico de reprodução, feedbacks de usuários, interação do suporte.

1.3. Justificativa da Escolha

MySQL para dados estruturados: Ideal para armazenar informações estruturadas como dados de usuários, playlists e transições financeiras, pois fornece forte integridade de dados e suporte a transações ACID.

MongoDB para dados não-estruturados: Adequado para dados não estruturados e flexíveis, como histórico de produção, feedbacks e interações de

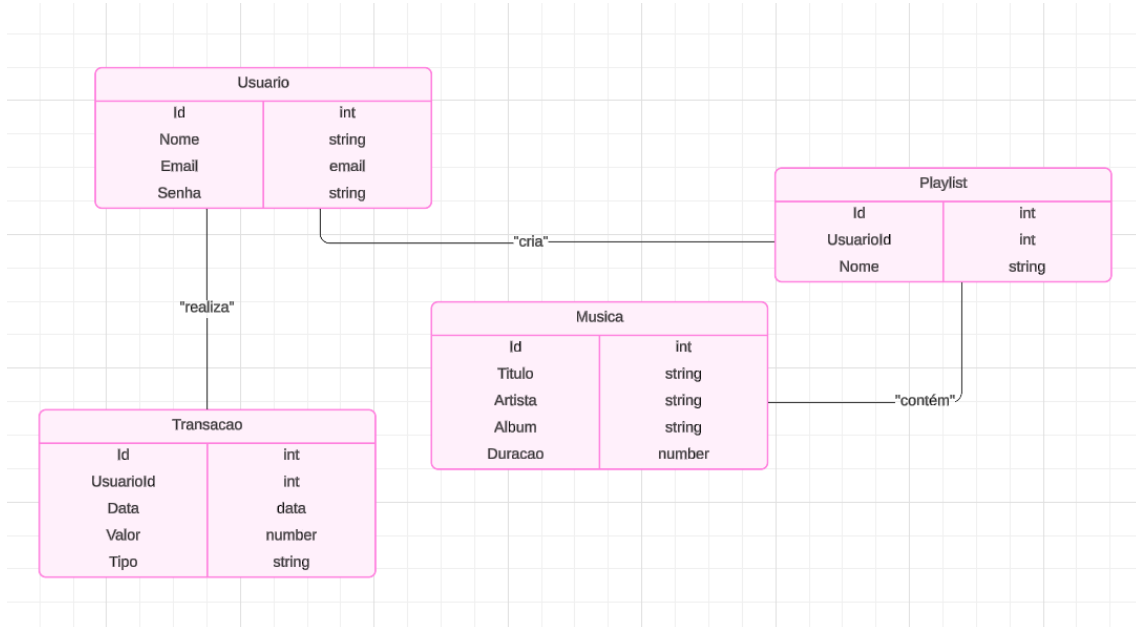
suporte, devido à sua capacidade de armazenar dados em documentos JSON flexíveis e escalar horizontalmente.

2. CONFIGURAÇÃO DO AMBIENTE

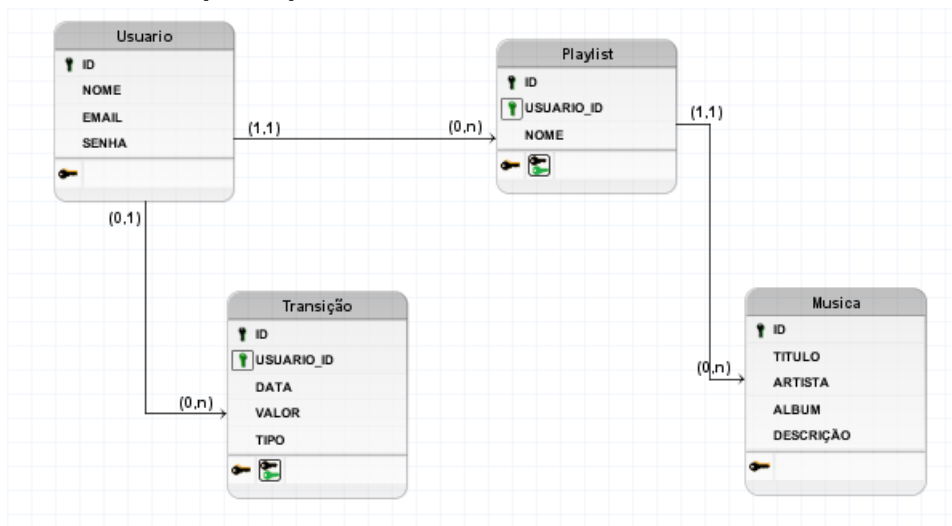
```
1 ● CREATE DATABASE streaming_musica;
2 ● USE streaming_musica;
3
4 ● ○ CREATE TABLE Usuarios (
5     id INT auto_increment primary key,
6     nome VARCHAR(100),
7     email VARCHAR(100),
8     senha VARCHAR(10)
9 );
10
11 ● ○ CREATE TABLE Playlists (
12     id INT auto_increment primary key,
13     nome VARCHAR(100),
14     usuario_id INT,
15     FOREIGN KEY (usuario_id) REFERENCES usuarios(id)
16 );
17
18 ● ○ CREATE TABLE TransiçõesFinanceiras (
19     id INT auto_increment primary key,
20     data_transicao DATE,
21     valor DECIMAL(10,5),
22     usuario_id INT,
23     FOREIGN KEY (usuario_id) REFERENCES usuarios(id)
24 );
25
26 ● select * from Usuarios
```

3. DIAGRAMA DE MODELAGEM

3.1. (MER)



3.2. (DER)



4. BANCO DE DADOS NORMALIZADO

```
1 ● CREATE DATABASE streaming_musica;
2 ● USE streaming_musica;
3
4 ● ○ CREATE TABLE Usuarios (
5     id INT auto_increment primary key,
6     nome VARCHAR(100),
7     email VARCHAR(100),
8     senha VARCHAR(10)
9 );
10
11 ● ○ CREATE TABLE Playlists (
12     id INT auto_increment primary key,
13     nome VARCHAR(100),
14     usuario_id INT,
15     FOREIGN KEY (usuario_id) REFERENCES usuarios(id)
16 );
17
18 ● ○ CREATE TABLE TransiçõesFinanceiras (
19     id INT auto_increment primary key,
20     data_transicao DATE,
21     valor DECIMAL(10,5),
22     usuario_id INT,
23     FOREIGN KEY (usuario_id) REFERENCES usuarios(id)
24 );
25
26 ● select * from Usuarios
```

5. DICIONÁRIO DE DADOS

USUÁRIOS

ATRIBUTOS	TIPO	TAMANHO	CHAVE
ID	INT	AUTO INCREMENT	PK
NOME	VARCHAR	100	X
EMAIL	VARCHAR	UNIQUE	X
SENHA	VARCHAR	10	X

PLAYLISTS

ATRIBUTOS	TIPO	TAMANHO	CHAVE
ID	INT	AUTO INCREMENT	PK
NOME	VARCHAR	100	X
USUARIO_ID	INT	REFERENCES USUARIO(ID)	FK

TRANSIÇÕES

ATRIBUTOS	TIPO	TAMANHO	CHAVE
ID	INT	AUTO INCREMENT	PK
USUARIO_ID	INT	REFERENCES USUARIO(ID)	FK
DATA	DATE	X	X
VALOR	DECIMAL	10, 2	X

6. CONCLUSÃO

A implementação de um novo sistema de banco de dados para a plataforma de streaming de música foi um grande passo para lidar com o aumento de dados e usuários. Escolhemos o MySQL para dados estruturados como usuários e transações financeiras porque ele é super confiável e mantém tudo organizado. Para os dados estruturados e não estruturados, como históricos de reprodução e feedbacks, escolhemos o MongoDB, que é flexível e comporta o necessário.

A configuração foi simplificada, com instruções simples. Os diagramas de modelagem (MER e DER) foram essenciais para entender a estrutura dos dados. A criação do banco normalizado assegurou a eliminação de redundâncias, e o dicionário documentou tabelas, colunas e relacionamentos, destacando a organização.

Essa abordagem integrada permitiu uma experiência de usuário personalizada e eficiente, otimizando análises e recomendações.