



SERVIÇO NACIONAL DE APRENDIZAGEM INDUSTRIAL

SENAI “GASPAR RICARDO JUNIOR”

Curso

**TÉCNICO EM DESENVOLVIMENTO
DE SISTEMAS**

Tema do Trabalho

João Alexandre da Silva Pereira

Sorocaba
Novembro – 2024



SERVIÇO NACIONAL DE APRENDIZAGEM INDUSTRIAL

SENAI “GASPAR RICARDO JUNIOR”

João Alexandre da Silva Pereira

Plataforma de Streaming de Música

Breve relatório sobre Plataforma
de Streaming de Música

Sorocaba
Novembro – 2024

HISTÓRICO DE VERSÕES

[illegible]

SUMÁRIO

INTRODUÇÃO.....	4
1. Bancos de Dados Relacionais (MySQL)	5
1.1. Usuários	5
1.2. Playlists e Músicas:.....	5
1.3. Transições financeiras:	5
2. Bancos de Dados Não-Relacionais (MongoDB)	6
Os bancos de dados não-relacionais, como MongoDB, são ideais para dados semiestruturados e não estruturados, com uma abordagem mais flexível que dispensa a necessidade de um esquema fixo. MongoDB é orientado a documentos, o que facilita a adaptação a grandes volumes de dados complexos e variados. Na plataforma, o	
MongoDB será usado para:	6
2.1. Histórico de reprodução:.....	6
2.2. Feedbacks dos usuários:.....	6
2.3. Interações de suporte:	6
Criação de tabelas no mySQL	6
CONCLUSÃO.....	7
3. Diagramas	Erro! Indicador não definido.
LISTA DE TABELAS	9



Plataforma de Streaming de Música

INTRODUÇÃO

A seleção de bancos de dados é essencial para plataformas de streaming de música que lidam com grandes quantidades de dados estruturados e não estruturados , como informações do usuário , listas de reprodução, histórico de produção e feedback. As características dos bancos de dados relacionais e não relacionais são comparadas neste relatório, que dá suporte à escolha do MySQL para dados estruturados e do MongoDB para dados não estruturados , oferecendo uma solução confiável e adaptável .

Bancos de Dados Relacionais (MySQL)

Bancos de dados bancos relacionais são recomendados para informações estruturadas que exigem alta consistência e transações seguras. são recomendados para informações estruturadas que exigem alta consistência e transações seguras. O MySQL adere ao o modelo relacional que é baseado no uso de tabelas conectadas por chaves primárias e secundárias. É perfeito para armazenar dados que exigem integridade e relacionamento entre entidades. modelo, que é baseado no uso de tabelas conectadas por chaves primárias e secundárias. É perfeito para armazenar dados que exigem integridade e relacionamento entre entidades. Na plataforma, plataforma de streaming, MySQL serão usado para:

Usuários

Informações detalhadas como nome, e-mail e data de nascimento, que exigem consistência e integridade de dados.

Playlists e Músicas:

Relacionamentos entre playlists e músicas, armazenados de forma organizada para fácil consulta e controle.

Transições financeiras:

Dados sobre assinaturas e pagamentos, que precisam de segurança e consistência em operações financeiras.

A escolha do MySQL é justificada pela sua aderência ao modelo ACID (Atomicidade, Consistência, Isolamento, Durabilidade), que garante segurança e confiabilidade em transações, especialmente importante para dados financeiros.

Bancos de Dados Não-Relacionais (MongoDB)

Os bancos de dados não-relacionais, como MongoDB, são ideais para dados semiestruturados e não estruturados, com uma abordagem mais flexível que dispensa a necessidade de um esquema fixo. MongoDB é orientado a documentos, o que facilita a adaptação a grandes volumes de dados complexos e variados. Na plataforma, o MongoDB será usado para:

Histórico de reprodução:

Com registros frequentes de reprodução, é necessário um banco de dados que permita flexibilidade no armazenamento e consulta de dados com diferentes atributos e estruturas.

Feedbacks dos usuários:

Avaliações e comentários dos usuários sobre músicas e playlists são armazenados de forma ágil, possibilitando consultas rápidas.

Interações de suporte:

Dados das interações com o suporte ao cliente, como tickets e chats, variam em estrutura e exigem um armazenamento mais flexível.

O MongoDB permite a escalabilidade horizontal, sendo adequado para grandes volumes de dados e acessos intensos. Além disso, oferece flexibilidade para armazenar e consultar dados variados, sendo a escolha ideal para o histórico de uso e interações não estruturadas.

Criação de tabelas no mySQL

```
create table Usuarios (  
  id_usuario int primary key,  
  nome varchar(100),  
  email varchar(100) unique,  
  data_nascimento date  
);
```

```
create table Playlists (  
  id_playlist int primary key,
```

```

        id_usuario int,
        nome_playlist varchar(100),
        FOREIGN KEY (id_usuario) REFERENCES Usuarios(id_usuario)
    );

create table Musicas (
    id_musica int primary key,
    nome_musica varchar(100),
    artista varchar(100)
);

create table Playlists_Musicas (
    id_playlist int,
    id_musica int,
    primary key (id_playlist, id_musica),
    foreign key (id_playlist) references Playlists(id_playlist),
    foreign key (id_musica) references Musicas(id_musica)
);

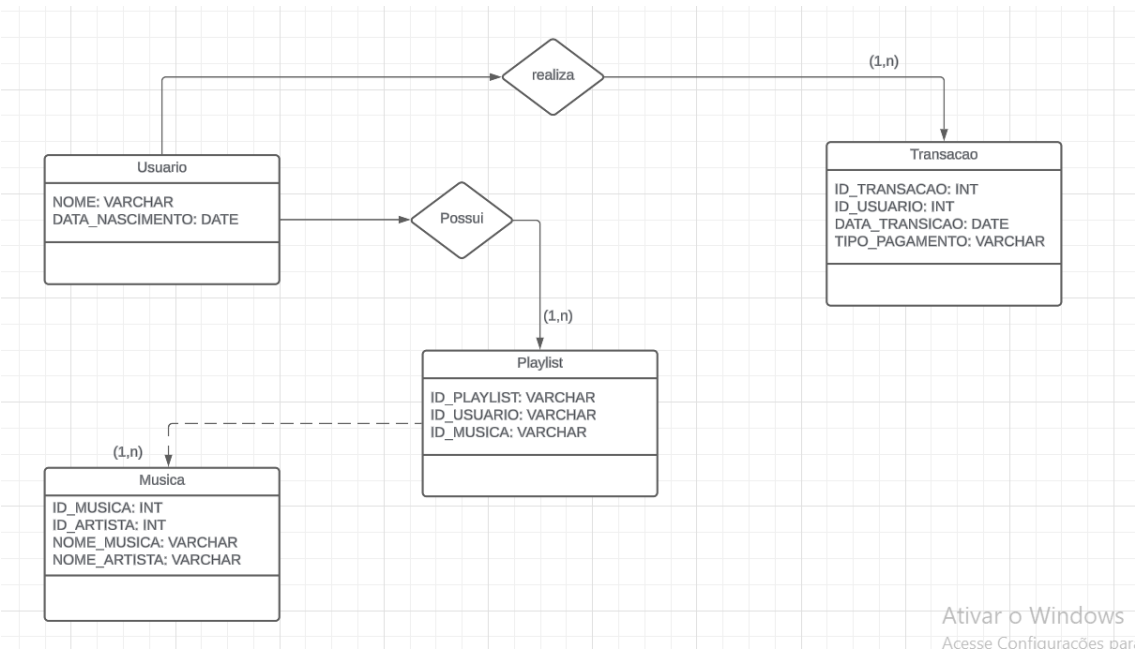
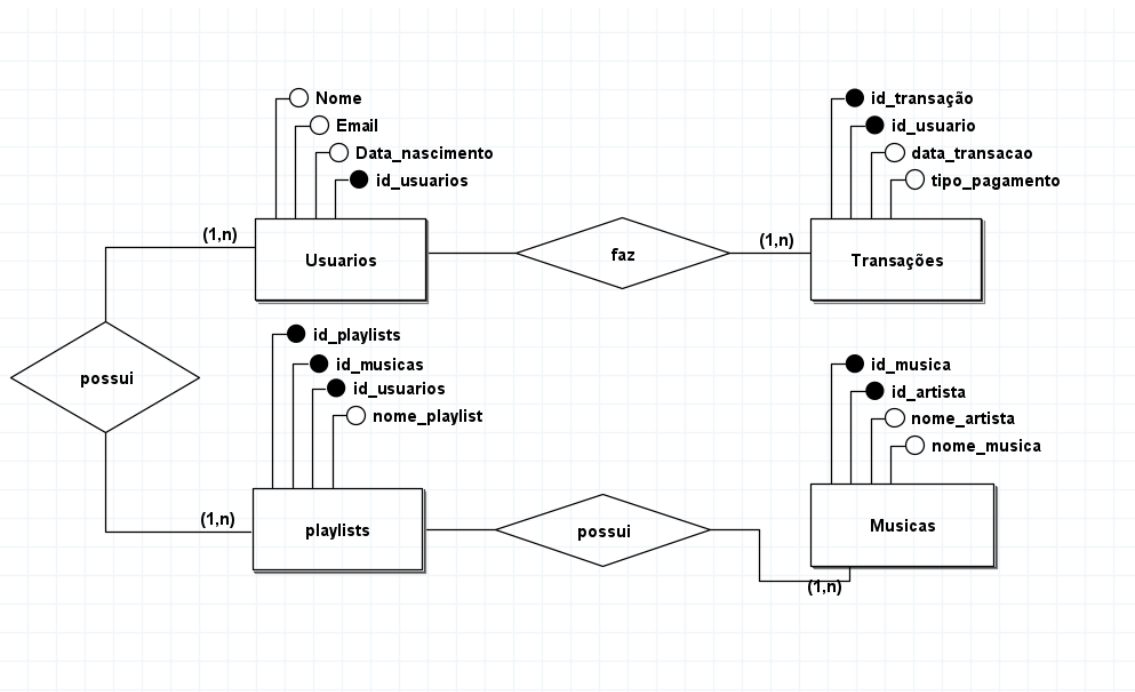
create table Transacoes (
    id_transacao int primary key,
    id_usuario int,
    valor decimal(10, 2),
    data_transacao date,
    foreign key (id_usuario) references Usuarios(id_usuario)
);

```

CONCLUSÃO

A integração entre MySQL e MongoDB permite que a plataforma de streaming de música atenda tanto aos requisitos de integridade e consistência dos dados estruturados quanto à flexibilidade e escalabilidade para dados não estruturados. Esta solução híbrida garante que o sistema consiga lidar com o aumento da complexidade e volume de dados, otimizando a experiência dos usuários e viabilizando análises mais detalhadas sobre comportamento e preferências musicais.

DIAGRAMAS



Ativar o Windows
Acesse Configurações para

LISTA DE TABELAS

TABELA 1 – Tabelas sobre o app de musicas

K18

✕

✓

f_x

|

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	usuarios							Musicas						
2	descrição		TIPO		CARACTERES			descrição		TIPO		CARACTERES		
3	nome:		varchar		100			nome_musica		VARCHAR		100		
4	email:		varchar		100			nome_artista		VARCHAR		100		
5	data_nascimento:		DATE		0000/00/00									
6														
7														
8														
9														
10	Playlists							Playlist_Musicas						
11	descrição		tipo		caracteres			descrição		tipo		caracteres		
12	ID_USUARIOS		INT		-			ID_playlist		INT		-		
13	NOME_PLAYLIST		VARCHAR		100			ID_artista		INT		-		
14														
15														
16			tranações											
17			descrição		tipo									
18			id_usuario		INT									
19			valor		VARCHAR									
20			data_transacao		DATE									
21			tipo_pagamento		VARCHAR									
22														
23														
24														
25														