****

**Serviço Nacional de Aprendizagem Industrial**

**SENAI “Gaspar Ricardo Junior”**

Curso

TÉCNICO EM DESENVOLVIMENTO

DE SISTEMAS

Avaliação Formativa

Quéren-Hapuque Lopes Vieira

Sorocaba

Novembro – 2024



**Serviço Nacional de Aprendizagem Industrial**

**SENAI “Gaspar Ricardo Junior”**

Quéren-Hapuque Lopes Vieira

Banco de dados

Prof. – Emerson Magalhães

Sorocaba

Novembro – 2024

**Sumário**

[1. Relatório Comparativo: 4](#_Toc182220646)

[2. Configuração do Ambiente: 5](#_Toc182220647)

[3. Diagramas de Modelagem: 6](#_Toc182220648)

[4. Banco de Dados Normalizado: 6](#_Toc182220649)

[5. Dicionário de Dados 6](#_Toc182220650)

# 

# Relatório Comparativo:

Dependendo do tipo de dados gerados, a escolha do modelo de banco de dados adequado pode impactar diretamente a eficiência no processamento, armazenamento e consultas. Em particular, os bancos de dados relacionais (como o MySQL) e os bancos de dados não-relacionais (como o MongoDB) oferecem abordagens distintas para lidar com dados estruturados e não estruturados.

Características de Bancos de Dados Relacionais

Estrutura de Dados: Os bancos de dados relacionais utilizam uma estrutura tabular composta por tabelas com linhas e colunas. Cada linha representa uma entrada ou registro, e cada coluna representa um campo ou atributo desse registro.

SQL: Consultas são feitas por meio da linguagem SQL (Structured Query Language), que oferece uma forma estruturada e eficiente de interagir com o banco de dados.

Integridade dos Dados: Garantem a consistência, precisão e integridade dos dados por meio de chaves primárias, chaves estrangeiras e restrições de dados.

Escalabilidade Vertical: Tradicionalmente, bancos de dados relacionais escalam verticalmente (aumentando a capacidade do servidor onde o banco está hospedado), o que pode ser limitado em grandes volumes de dados.

Exemplo de Banco Relacional: MySQL.

Vantagens do MySQL:

Dados Estruturados: Ideal para dados bem definidos e estruturados, como informações financeiras, cadastro de clientes, inventário e transações.

Integridade: O uso de chaves e relacionamentos entre tabelas garante a integridade referencial dos dados.

3. Características de Bancos de Dados Não-Relacionais

Estrutura de Dados: Os bancos de dados não-relacionais (NoSQL) podem armazenar dados de forma mais flexível. Entre os modelos populares, estão os documentais, chave-valor, colunares e de grafos.

MongoDB: Um exemplo de banco de dados não-relacional orientado a documentos, que armazena dados no formato BSON (uma extensão do JSON). Ele permite armazenar dados semi-estruturados e não-estruturados, como documentos com diferentes campos e tipos de dados.

Escalabilidade Horizontal: Ao contrário dos bancos de dados relacionais, os bancos NoSQL, como o MongoDB, escalam horizontalmente (adicionando mais servidores) para lidar com grandes volumes de dados distribuídos.

Flexibilidade: Não há um esquema fixo de dados, o que permite que os dados sejam armazenados sem uma estrutura rígida.

Vantagens do MongoDB:

Dados Não-Estruturados e Semi-Estruturados: Ideal para armazenar dados que não se encaixam facilmente em tabelas, como textos, imagens, dados de sensores, logs e redes sociais.

Alta Escalabilidade: MongoDB é projetado para ser distribuído e escalável, permitindo a gestão eficiente de grandes volumes de dados e altas taxas de leitura/gravação.

Desempenho e Flexibilidade: A flexibilidade no formato de dados (JSON/BSON) permite a adaptação fácil a mudanças nos requisitos de dados, como adição de novos campos sem a necessidade de reestruturar a base de dados.

4. Justificativa para a Escolha do Banco de Dados

Escolha do Banco de Dados Relacional (MySQL) para Dados Estruturados:

Tipo de Dados: Quando os dados gerados pela empresa são bem definidos, consistentes e com relações claras entre diferentes entidades (exemplo: informações de clientes, pedidos, inventário), o modelo relacional é a escolha ideal. Neste caso o Usuários (dados relacionais).

Consultas Complexas: A necessidade de realizar consultas complexas, como junções entre diferentes tabelas (por exemplo, relacionar dados de clientes com pedidos e produtos), torna o MySQL uma escolha apropriada. Neste caso Playlists e músicas (dados relacionais).

Escolha do Banco de Dados Não-Relacional (MongoDB) para Dados Não-Estruturados:

Tipo de Dados: Quando os dados são menos estruturados ou podem variar ao longo do tempo (como registros de logs, dados de sensores, imagens, vídeos ou posts de redes sociais), o MongoDB oferece a flexibilidade necessária para armazenar e processar esses dados. Neste caso os Feedbacks dos usuários (dados não-relacionais)

Agilidade e Flexibilidade: A natureza esquematizada dos dados (ou a falta de um esquema fixo) torna o MongoDB ideal para empresas que lidam com dados dinâmicos e mudanças frequentes nos requisitos de armazenamento de dados.

# Configuração do Ambiente:

Para configurar um ambiente de banco de dados, pode seguir-se um processo que inclui as seguintes fases:

* Analisar os requisitos, ou seja, identificar o objetivo do banco de dados
* Organizar os dados em tabelas
* Especificar chaves primárias e analisar relações
* Normalizar para padronizar as tabelas

Para configurar o ambiente de banco de dados é necessário primeiramente criar com

CREATE DATABASE plataformaMusica;

Após criar o banco de dados, precisamos avisar ao mysql que vamos usá-lo, para isso basta escrevermos

USE plataformaMusica;

Feito isso o nosso banco de dados está criado, faltando apenas criar a nossa tabela. Para isso vamos usar o comando

CREATE TABLE

CREATE TABLE usuarios(

codigo int(4) AUTO\_INCREMENT,

nome varchar(30) NOT NULL,

email varchar(50),

PRIMARY KEY (codigo)

);

Como podemos ver, nossa tabela está criada.

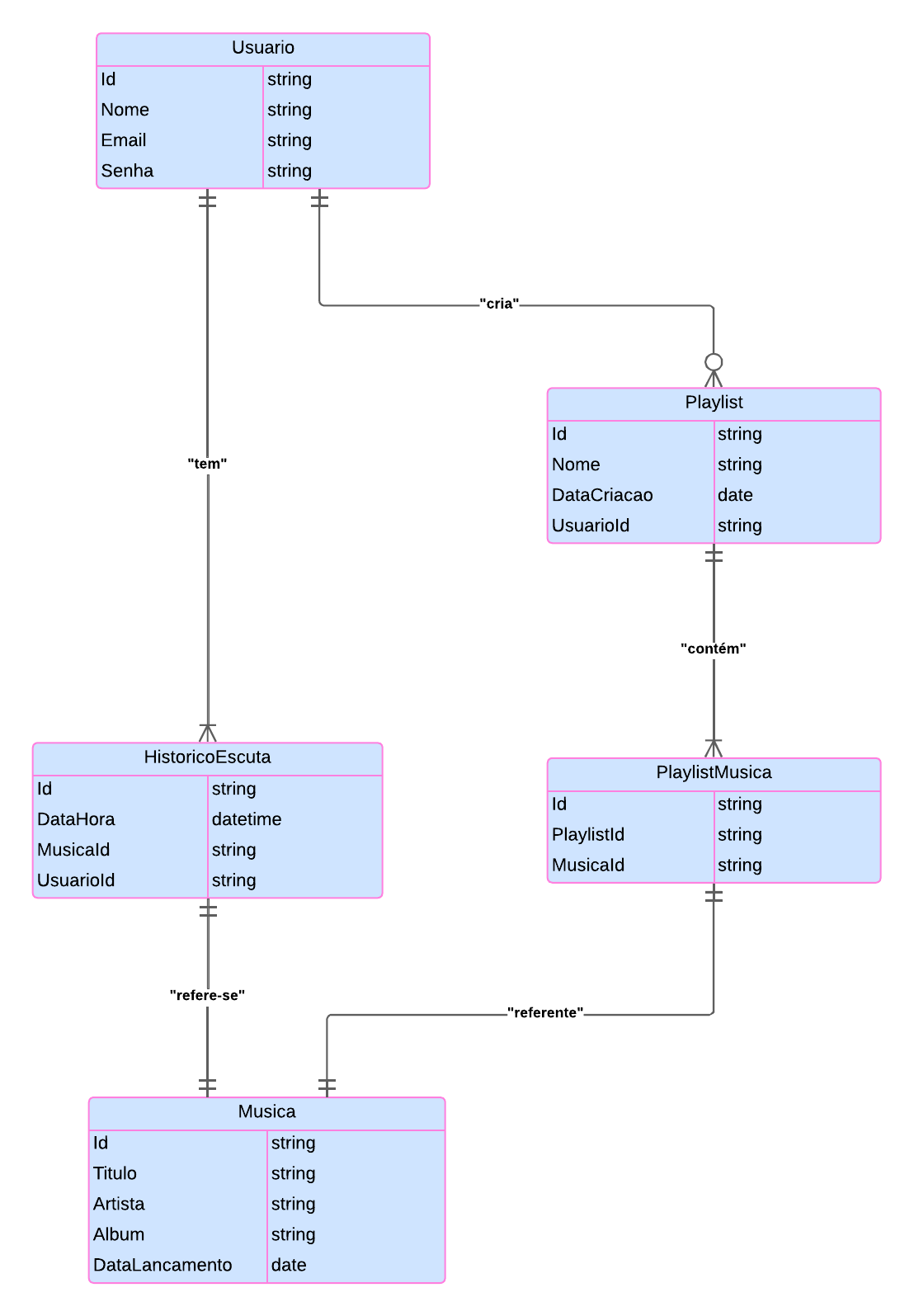
Agora vamos inserir alguma informação na nossa tabela, para isso vamos usar o comando

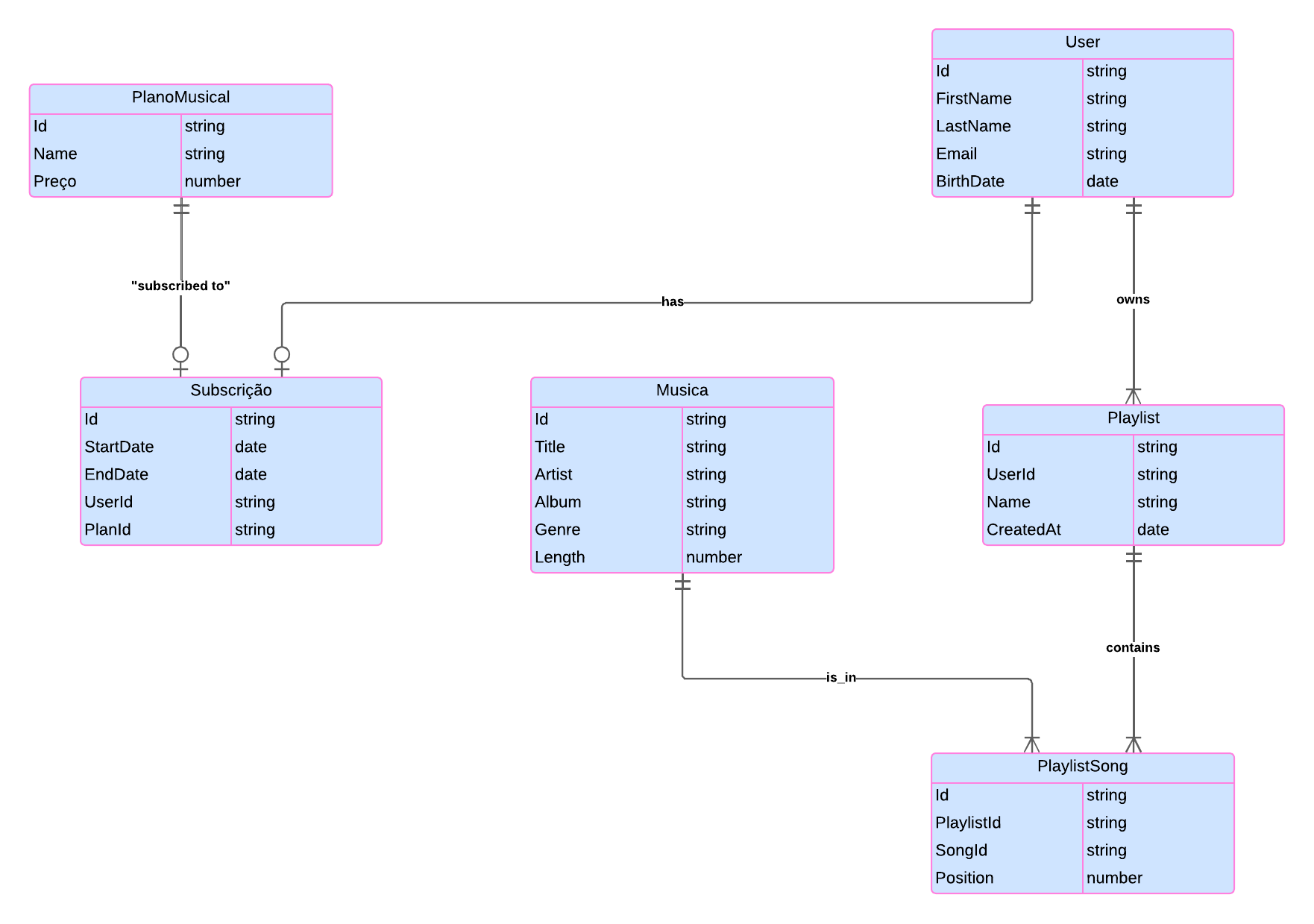
INSERT

INSERT INTO usuarios(codigo, nome, email) VALUES (null, “Ricardo”, “ricoarrigoni@gmail.com”) ;

INSERT INTO usuarios(codigo, nome, email) VALUES (null, “João”, “joao@gmail.com”) ;

# Diagramas de Modelagem:





# Banco de Dados Normalizado:

create database db\_plataforma;

use db\_plataforma;

create table usuarios(

codigo int(4) Auto\_increment,

nome varchar(30) not null,

email varchar(50),

primary key (codigo),

senha varchar(8)

);

insert into usuarios(codigo, nome, email, senha) values (null, "Ricardo", "ricardogoni@gamil.com");

insert into usuarios(codigo, nome, email, senha) values (null, "Maria", "mariaferraz@gamil.com");

insert into usuarios(codigo, nome, email, senha) values (null, "Pedro", "pedrozo@gamil.com");

insert into usuarios(codigo, nome, email, senha) values (null, "Isabela", "Isabelamadeiro@gamil.com");

create table playlistemusic(

codigo int(4) Auto\_increment,

nomeMusica varchar(20),

nomePlaylist varchar(20),

primary key (codigo)

);

insert into playlistmusic(codigo, nomeMusica, nomePlaylist) values (null, "Proibida pra mim", "só charlie jnr");

insert into playlistmusic(codigo, nomeMusica, nomePlaylist) values (null, "Caraca muleke", "Pagodin");

insert into playlistmusic(codigo, nomeMusica, nomePlaylist) values (null, "Modo esquece", "funk pra animar");

create table historicEscuta(

codigo int(4) Auto\_increment,

usuario varchar(20),

email varchar(50),

primary key (codigo),

dataHora datetime(20)

);

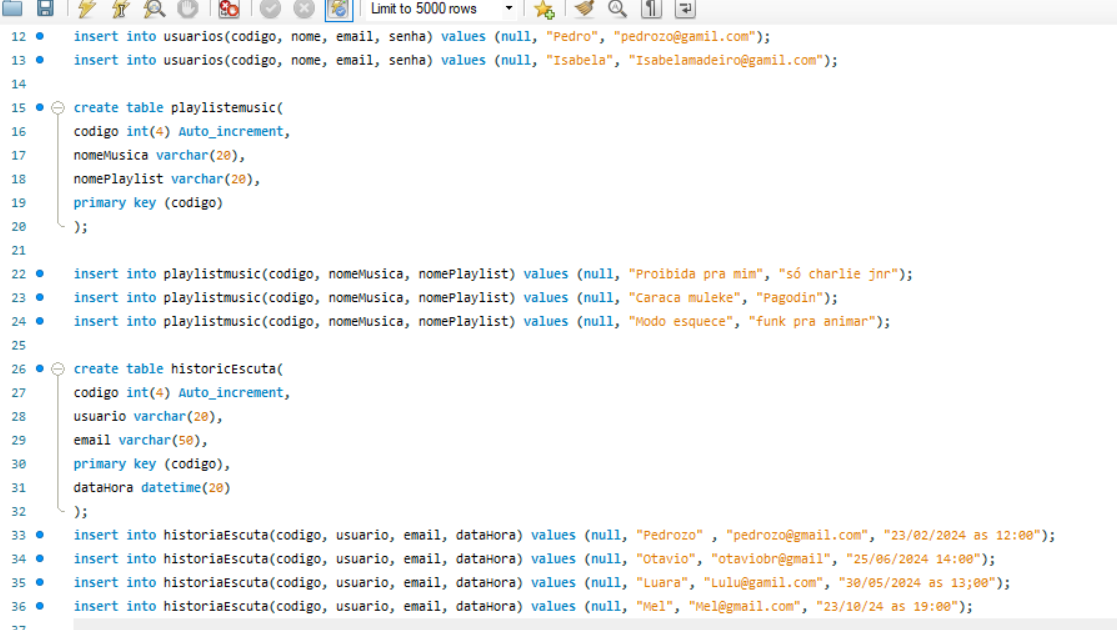
insert into historiaEscuta(codigo, usuario, email, dataHora) values (null, "Pedrozo" , "pedrozo@gmail.com", "23/02/2024 as 12:00");

insert into historiaEscuta(codigo, usuario, email, dataHora) values (null, "Otavio", "otaviobr@gmail", "25/06/2024 14:00");

insert into historiaEscuta(codigo, usuario, email, dataHora) values (null, "Luara", "Lulu@gamil.com", "30/05/2024 as 13;00");

insert into historiaEscuta(codigo, usuario, email, dataHora) values (null, "Mel", "Mel@gmail.com", "23/10/24 as 19:00");





# Dicionário de Dados:

Tabelas

Usuários (dados relacionais)

Histórico de reprodução (dados não-relacionais)

Playlists e músicas (dados relacionais)

Inserção de Dados como

Nome

Código

E-mail

Usuário

Playlist

Música

Data e hora

Relacionamento ao Plano Musical, Usuário, Playlist, Música e o Histórico.