**SISTEMA DE ORGANIZAÇÃO DE DADOS**  
**Desenvolvimento de um aplicativo para organização de *Playlist***

ERIK EIJI MATSUZAKI - 824224220

GUILHERME DA COSTA LIMA - 824215481

JOÃO VICTOR GOUVEIA SANTINELLO - 824212987

KAUE VIEIRA CERQUEIRA - 824217712

LUIZ HENRYQUE DA CONCEIÇÃO - 824210312

PEDRO SARMENTO - 82429017

PIETRO CIARCIA - 824214866

ORIENTADOR: PROF. ROBSON CALVETTI

**UNIVERSIDADE SÃO JUDAS TADEU**  
**Curso de Programação de Soluções Computacionais**

São Paulo – 2024

**SISTEMA DE ORGANIZAÇÃO DE DADOS  
Desenvolvimento de um aplicativo para organização de *Playlist***

ERIK EIJI MATSUZAKI – 824224220

GUILHERME DA COSTA LIMA – 824215481

JOÃO VICTOR GOUVEIA SANTINELLO – 824212987

KAUE VIEIRA CERQUEIRA - 824217712

LUIZ HENRYQUE DA CONCEIÇÃO - 824210312

PEDRO

PIETRO CIARCIA - 824214866

Orientador: PROF. ROBSON CALVETTI

Projeto de Desenvolvimento de Software

**UNIVERSIDADE SÃO JUDAS TADEU**  
**Curso de Programação de Soluções Computacionais**

São Paulo – 2024

Sumário

[1.Introdução 4](#_Toc184242822)

**2**. **Fundamentação Teórica...............................................................................................................5**

[**3. Metodologia** 6](#_Toc184242823)

[**4. Desenvolvimento do Sistema** 8](#_Toc184242824)

[**5. Análise de Resultados e Conclusão** 10](#_Toc184242825)

# **1.Introdução**

O crescente consumo de música digital tem gerado uma demanda significativa por soluções eficientes de organização e gerenciamento de conteúdo musicais. Muitas plataformas de streaming oferecem funcionalidades como playlists personalizadas, mas o gerenciamento dessas playlists de forma individual, personalizada e off-line ainda representa um desafio para muitos usuários.

O trabalho propõe o desenvolvimento de um aplicativo para organização destas playlists, permitindo ao usuário a criação, edição, e remoção de playlists de forma simples e eficiente. Este aplicativo será construído utilizando a linguagem de programação Java, com a integração ao banco de dados MySQL para persistência dos dados, e uma interface gráfica desenvolvida com a biblioteca Swing.

O objetivo principal deste projeto é proporcionar aos usuários uma ferramenta de fácil utilização para gerenciar suas playlists de maneira personalizada, tanto em termos de conteúdo quanto na forma de interação com o sistema. Além disso, o projeto visa aplicar conceitos de programação orientada a objetos e manipulação de banco de dados relacional, com foco na criação de uma interface eficiente.

Para o desenvolvimento do aplicativo, serão utilizadas as tecnologias Java para a construção do sistema, MySQL para o armazenamento dos dados, e a biblioteca Swing para a criação da interface gráfica. O processo de desenvolvimento será dividido em fases: definição de requisitos, desenvolvimento da interface gráfica, implementação da lógica de negócio e integração com o banco de dados.

Ao final do projeto, espera-se que o sistema atenda às necessidades de usuários que buscam uma ferramenta eficiente para organização de playlists, garantindo a facilidade de uso e a confiabilidade no armazenamento das informações.

**2. Fundamentação Teórica**

**2.1 Java e Orientação a Objetos**  
A linguagem Java é amplamente utilizada no desenvolvimento de aplicativos devido à sua robustez, portabilidade e ao suporte completo para programação orientada a objetos (POO). A POO é um paradigma que organiza o código em torno de "objetos", que representam entidades do mundo real. O uso de classes e objetos proporciona modularidade, reutilização de código e facilita a manutenção do sistema. No nosso projeto, utilizaremos essas técnicas para criar uma aplicação de fácil manutenção e extensão.  
  
  
**2.2 JDBC e Banco de Dados MySQL**  
O **Java Database Connectivity** (JDBC) é uma API que permite que aplicativos Java interajam com bancos de dados relacionais. No projeto, utilizaremos o JDBC para estabelecer a conexão entre o aplicativo e o banco de dados **MySQL**, permitindo que o sistema armazene e recupere dados das playlists de maneira eficiente e segura. O **MySQL** é um sistema de gerenciamento de banco de dados relacional amplamente utilizado, que fornece funcionalidades como armazenamento e consulta de dados em tabelas, essenciais para o armazenamento das playlists.  
Para a criação e gerenciamento do banco de dados MySQL, utilizaremos o **MySQL Workbench**, uma ferramenta gráfica que facilita a modelagem, design e administração de bancos de dados. O MySQL Workbench será fundamental na criação das tabelas necessárias para armazenar as informações das playlists, como nome, músicas e dados dos usuários. Além disso, ele proporcionará uma visualização intuitiva dos dados armazenados e permitirá a execução de queries SQL diretamente na interface, o que agiliza o processo de manutenção e testes do banco de dados.  
  
  
**2.3 Swing e Interface Gráfica**A biblioteca Swing é uma das ferramentas mais populares para a criação de interfaces gráficas no Java. Através do Swing, é possível criar interfaces amigáveis e interativas, utilizando componentes como botões, caixas de diálogo, painéis de texto, tabelas e outros. No nosso aplicativo, a interface gráfica será construída de forma a permitir ao usuário interagir facilmente com as playlists, podendo criar, editar e excluir playlists de forma intuitiva.

### **3. Metodologia**

O desenvolvimento do sistema será realizado por meio de um processo estruturado, dividido em várias etapas sequenciais, que visam garantir a qualidade e a eficácia do aplicativo de organização de playlists. O projeto será conduzido utilizando práticas de **programação orientada a objetos (POO)**, **integração com banco de dados relacional** (MySQL), e **interface gráfica** desenvolvida com a biblioteca **Swing**. A metodologia adotada para o desenvolvimento será o modelo **cascata**, dividido em fases bem definidas.

#### **3.1 Planejamento e Definição de Requisitos**

A primeira etapa do desenvolvimento será a definição detalhada dos requisitos do sistema. Isso inclui a identificação das funcionalidades principais do aplicativo, como:

* Criação de playlists
* Edição e remoção de playlists
* Visualização das playlists existentes
* Armazenamento de dados no banco de dados MySQL
* Criação de uma interface gráfica intuitiva para interação com o usuário.

Durante esta fase, será elaborado um documento de requisitos que definirá as funcionalidades específicas e as expectativas para o sistema. A comunicação entre os membros do grupo será feita por meio de reuniões, utilizando ferramentas de gestão de tarefas e versionamento de código, como o **GitHub**, para garantir que todos os integrantes estejam alinhados.

#### **3.2 Desenvolvimento da Estrutura do Banco de Dados**

Nesta fase, será utilizado o **MySQL Workbench** para o design do banco de dados relacional que armazenará as playlists. O banco de dados será estruturado com as tabelas necessárias para armazenar as informações das playlists.

Além disso, será realizada a configuração da conexão entre o banco de dados e o aplicativo por meio do **JDBC**, que permitirá ao sistema executar operações de CRUD (Create, Read, Update e Delete) no banco de dados de maneira segura e eficiente.

#### **3.3 Desenvolvimento da Interface Gráfica**

A interface gráfica será construída utilizando a biblioteca **Swing**, que faz parte da API padrão do **Java**. A interface será projetada para ser simples e intuitiva, permitindo ao usuário interagir facilmente com o sistema. As funcionalidades da interface incluem:

* Tela para criar e editar playlists
* Tela de exibição das playlists criadas
* Botões e campos de entrada para interação com o sistema

O design da interface será orientado à experiência do usuário, buscando um layout funcional e de fácil navegação.

#### **3.4 Implementação da Lógica de Negócio**

Nesta etapa, será implementada a lógica de negócio que controla as interações entre a interface gráfica e o banco de dados. Isso inclui a implementação de funcionalidades como:

* **Criação de playlists**: o usuário será capaz de criar uma nova playlist e adicionar músicas a ela.
* **Edição de playlists**: o usuário poderá modificar a playlist existentes, incluindo adicionar ou remover músicas.
* **Persistência de dados**: todas as informações de playlists e músicas serão armazenadas no banco de dados MySQL através da integração com o JDBC.

#### **3.5 Testes e Validação**

Após o desenvolvimento das funcionalidades, serão realizados testes para garantir que o sistema esteja funcionando corretamente e atendendo aos requisitos estabelecidos. Os testes envolverão:

* **Testes unitários**: para verificar se cada módulo de código (como métodos e funções) está funcionando conforme esperado.
* **Testes de integração**: para garantir que a integração entre o sistema e o banco de dados, bem como a comunicação entre os componentes do aplicativo, esteja ocorrendo corretamente.
* **Testes de usabilidade**: para avaliar a interface gráfica e assegurar que ela seja fácil de usar e intuitiva para os usuários.

#### **3.6 Entrega e Documentação**

Por fim, o sistema será documentado conforme as normas da **ABNT**, incluindo a descrição detalhada do funcionamento do sistema, as tecnologias utilizadas, o processo de desenvolvimento e os testes realizados. A documentação técnica será entregue em formato digital, juntamente com o código-fonte do projeto no **GitHub**.

### **4. Desenvolvimento do Sistema**

O desenvolvimento do sistema de organização de playlists foi realizado em várias etapas, incluindo a implementação das funcionalidades principais, a criação da interface gráfica e a integração com o banco de dados MySQL. A seguir, descrevemos as etapas de desenvolvimento e como cada componente do sistema foi implementado.

#### **4.1 Arquitetura do Sistema**

O sistema foi desenvolvido utilizando a linguagem de programação **Java**, com uma arquitetura orientada a objetos (POO), permitindo uma maior organização e manutenção do código. O sistema é dividido em três camadas principais:

* **Camada de Apresentação**: Responsável pela interface gráfica com o usuário, construída com a biblioteca **Swing**.
* **Camada de Lógica de Negócio**: Contém as regras do sistema, implementadas através de classes Java.
* **Camada de Persistência**: Responsável pela comunicação com o banco de dados MySQL através da **JDBC** (Java Database Connectivity).

#### **4.2 Banco de Dados MySQL**

O banco de dados foi criado utilizando o **MySQL**, um sistema de gerenciamento de banco de dados relacional. O **MySQL Workbench** foi utilizado para projetar e administrar o banco de dados, criando tabelas, consultas e relacionamentos entre os dados. O banco de dados foi estruturado para armazenar informações essenciais para o sistema, como dados de usuários, playlists e músicas.

A seguir, estão as principais tabelas do banco de dados:

* **Tabela de Músicas**: Armazena as músicas que pertencem ao grupo.
  + Campos: id\_musica, nome\_musica, artista.

A persistência de dados foi realizada através da utilização de **JDBC**, que permite a execução de comandos SQL diretamente do código Java para manipulação do banco de dados.

#### **4.3 Interface Gráfica**

A interface gráfica foi desenvolvida utilizando a biblioteca **Swing** do Java, que fornece componentes como botões, campos de texto e painéis para construir interfaces de usuário. A interface foi projetada para ser simples e intuitiva, com as seguintes telas principais:

* **Tela Principal**: Exibe a lista de músicas e opções para adicionar, editar e excluir.

A interface foi organizada de forma a proporcionar uma navegação fluida e fácil interação com o sistema. Todos os componentes da interface foram criados manualmente, sem o uso de ferramentas automáticas, para garantir que o código estivesse totalmente alinhado com as exigências do projeto.

#### **4.4 Conexão com o Banco de Dados**

A comunicação entre o aplicativo Java e o banco de dados MySQL foi estabelecida utilizando a **API JDBC** (Java Database Connectivity), que permite executar comandos SQL diretamente do código Java, facilitando a manipulação dos dados persistidos. Para garantir a integração bem-sucedida, foi necessário configurar corretamente o driver JDBC para o MySQL e criar métodos de conexão.

A estrutura do banco de dados foi projetada para armazenar informações essenciais relacionadas às músicas. A **tabela de músicas** armazena as informações das músicas que pertencem ao grupo. A tabela de músicas foi definida da seguinte forma:

* **Tabela de Músicas**: Responsável por armazenar informações sobre as músicas que pertencem à playlist. Cada música possui um identificador único (id\_musica), o nome da música (nome\_musica), e o nome do artista (artista).

**Campos da Tabela de Músicas**:

* + id\_musica: Identificador da música.
  + nome\_musica: Nome da música.
  + artista: Nome do artista da música.

A persistência de dados foi realizada por meio de comandos SQL para executar operações de **CRUD** (criação, leitura, atualização e exclusão) nas tabelas do banco de dados.

#### **4.5 Funcionalidades da Interface Gráfica**

A interface gráfica foi projetada utilizando a biblioteca **Swing** do Java. A interface foi elaborada para ser simples e intuitiva, permitindo que os usuários interajam de maneira eficiente com o sistema. A seguir, estão as telas principais do aplicativo, que foram desenvolvidas para facilitar a interação do usuário:

* **Tela Principal**: A tela principal é responsável por exibir a lista de músicas disponíveis no sistema, com informações como nome da música e artista. Além disso, o usuário pode **adicionar**, **editar** ou **excluir músicas** diretamente da tela. Para tornar a navegação mais fluida, foram inseridos botões e campos de entrada que facilitam o gerenciamento das músicas na playlist.

### **5. Análise de Resultados e Conclusão**

O desenvolvimento do aplicativo de organização de playlists foi realizado conforme os requisitos definidos na fase de planejamento, utilizando a linguagem de programação **Java**, a API **JDBC** para integração com o banco de dados **MySQL**, e a biblioteca **Swing** para a criação da interface gráfica. O objetivo do projeto era criar uma ferramenta eficiente para o gerenciamento de playlists de música, com funcionalidades como criação, edição, e remoção de playlists.

#### **5.1 Resultados Alcançados**

Ao final do desenvolvimento, o sistema atendeu aos principais requisitos definidos inicialmente. Os resultados alcançados incluem:

* **Funcionalidade de CRUD**: O aplicativo permite a criação, visualização, edição e exclusão de musicas, armazenando os dados de forma persistente no banco de dados MySQL. A integração entre o aplicativo e o banco de dados foi bem-sucedida, permitindo uma comunicação eficiente e segura.
* **Interface Gráfica**: A interface gráfica, desenvolvida com a biblioteca Swing, é intuitiva e fácil de usar, proporcionando uma experiência de usuário agradável. As telas de navegação são claras, permitindo que o usuário realize as operações de maneira simples e direta.
* **Persistência de Dados**: A utilização do MySQL como sistema de gerenciamento de banco de dados nos permitiu o armazenamento confiável das musicas. A comunicação com o banco de dados através do JDBC funcionou corretamente, garantindo a integridade e segurança dos dados.
* **Testes**: Foram realizados testes unitários e de integração, que confirmaram o bom funcionamento das funcionalidades do sistema. Não foram identificados erros significativos durante os testes, e a usabilidade da interface foi validada com a realização de testes com usuários.

#### **5.2 Desafios Enfrentados**

Embora o projeto tenha atingido seus objetivos principais, alguns desafios surgiram ao longo do desenvolvimento:

* **Integração entre o Java e o MySQL**: A integração do JDBC com o banco de dados MySQL exigiu a configuração correta das conexões e a gestão de exceções. Isso demandou um tempo maior de desenvolvimento, principalmente durante os testes iniciais.
* **Desenvolvimento da Interface Gráfica**: A criação de uma interface gráfica sem o uso de ferramentas automáticas, como o NetBeans, foi um desafio. No entanto, a implementação de componentes Swing personalizados foi uma boa oportunidade para aprender mais sobre a biblioteca e suas capacidades.
* **Gerenciamento do Tempo**: A complexidade do projeto e a necessidade de conciliar o desenvolvimento com outras atividades do dia a dia exigiu um planejamento rigoroso, e muito empenho por parte de todos do grupo especialmente durante as fases de iniciais e testes.

#### **5.3 Aprendizados**

O projeto proporcionou uma oportunidade significativa de aprendizado, tanto em termos técnicos quanto de trabalho em equipe. Entre os principais aprendizados, destacam-se:

* **Programação Orientada a Objetos (POO)**: O uso de POO em Java foi essencial para a organização e modularização do código, facilitando a manutenção e a escalabilidade do sistema.
* **Banco de Dados Relacional**: A integração com o banco de dados MySQL e o uso do JDBC aprimoraram a compreensão sobre a persistência de dados em sistemas relacionais.
* **Interface Gráfica**: O trabalho com a biblioteca Swing contribuiu para o aprimoramento das habilidades de desenvolvimento de interfaces gráficas no Java, permitindo a criação de uma interface funcional e agradável para o usuário.
* **Trabalho em Equipe**: A divisão de tarefas entre os membros do grupo foi fundamental para o sucesso do projeto, sendo essencial a boa comunicação e o compartilhamento de responsabilidades para o desenvolvimento de um sistema completo e funcional.

#### **5.4 Conclusão**

O projeto atingiu com sucesso os objetivos propostos de desenvolver uma aplicação funcional para a organização de uma playlist, utilizando tecnologias como **Java**, **MySQL**, **JDBC**, e **Swing**. O processo de desenvolvimento proporcionou um aprendizado significativo para todos os integrantes do grupo, além de permitir a aplicação prática de conceitos de programação orientada a objetos, banco de dados relacional e desenvolvimento de interfaces gráficas. Apesar dos desafios enfrentados ao longo do processo, o sistema final atende aos requisitos iniciais e pode ser expandido e melhorado em versões futuras.