

**DOCUMENTAÇÃO DO SISTEMA DE
GERENCIAMENTO DE TAREFAS EM JAVA
COM CONEXÃO MYSQL.**

PROJETO A3 - Prof. CALVETTI

DOCUMENTAÇÃO DO SISTEMA DE GERENCIAMENTO DE TAREFAS EM JAVA COM CONEXÃO MYSQL

1. Introdução

Este trabalho apresenta a implementação de um aplicativo de gerenciamento de tarefas (Todo List) desenvolvido em Java, com interface gráfica baseada na biblioteca Swing, e conexão com banco de dados relacional MySQL. O sistema permite ao usuário adicionar, editar, excluir e visualizar tarefas, além de exibir tarefas pendentes ou concluídas. A comunicação com o banco de dados é feita utilizando JDBC (Java Database Connectivity), onde as operações de CRUD (Create, Read, Update, Delete) são realizadas.

2. Objetivo

O objetivo deste projeto é demonstrar a integração entre uma interface gráfica de usuário (GUI) e um banco de dados MySQL, permitindo a manipulação e exibição de dados dinâmicos de tarefas.

3. Tecnologias Utilizadas

- **Java 8+:** Linguagem de programação principal utilizada para o desenvolvimento do aplicativo.
- **Swing:** Biblioteca para criação da interface gráfica.
- **JDBC (Java Database Connectivity):** Tecnologia para comunicação entre a aplicação Java e o banco de dados MySQL.
- **MySQL 8.0+:** Sistema gerenciador de banco de dados relacional utilizado.
- **mysql-connector-j-8.4.0.jar:** Driver JDBC para comunicação com o banco de dados MySQL.

4. Estrutura do Código

O código está estruturado da seguinte forma:

- **Classe TodoListApp:** Classe principal que contém a interface gráfica e a lógica de manipulação das tarefas. Herda de `JFrame` e configura os componentes da GUI, como campos de texto, botões e lista de tarefas.
- **Classe Task:** Classe interna usada para representar as tarefas com seus atributos (título, descrição, data de vencimento e status).

A seguir, detalharemos as principais partes do código.

5. Descrição da Interface Gráfica

A interface gráfica é composta por três áreas principais:

- **Painel de Entrada de Dados (inputPanel):** Contém campos de texto para o título, descrição e data de vencimento das tarefas, além de um checkbox para marcar o status (concluída ou não).
- **Painel de Controle (controlPanel):** Contém botões para adicionar, editar, excluir e filtrar tarefas (mostrar todas, mostrar pendentes e mostrar concluídas).

- **Lista de Tarefas:** Exibe as tarefas em uma lista. A lista é atualizada conforme as operações realizadas (adicionar, editar, excluir).

6. Conexão com o Banco de Dados

A conexão com o banco de dados MySQL é feita no método `connectToDatabase()`. O código realiza a conexão utilizando o JDBC com a URL do banco de dados, usuário e senha. Caso haja falha na conexão, uma mensagem de erro é exibida ao usuário.

```

99 private Connection connectToDatabase() {
100     try {
101         String url = "jdbc:mysql://localhost:3306/todo_list_db";
102         String user = "root";
103         String password = "";
104         return DriverManager.getConnection(url, user, password);
105     } catch (SQLException e) {
106         JOptionPane.showMessageDialog(this, "Erro ao conectar ao banco de dados: " + e.getMessage());
107         return null;
108     }
109 }

```

7. Funcionalidade dos Botões

Os botões da interface gráfica possuem as seguintes funcionalidades:

- **Adicionar:** Adiciona uma nova tarefa no banco de dados e na lista de tarefas exibida.
- **Editar:** Permite editar os dados de uma tarefa selecionada.
- **Excluir:** Exclui a tarefa selecionada.
- **Mostrar Todas:** Exibe todas as tarefas cadastradas no banco de dados.
- **Mostrar Pendentes:** Exibe apenas as tarefas não concluídas.
- **Mostrar Concluídas:** Exibe apenas as tarefas concluídas.

Cada botão está vinculado a um `ActionListener` que chama os métodos apropriados para cada ação. Por exemplo, o método `addTask()` é responsável por adicionar uma tarefa no banco de dados e atualizar a lista exibida.

8. Interação com o Banco de Dados

A manipulação das tarefas no banco de dados ocorre nos seguintes métodos:

- **addTask():** Adiciona uma nova tarefa no banco de dados.
- **loadTasks():** Carrega todas as tarefas armazenadas no banco de dados para a lista de tarefas exibida.
- **editTask():** Edita a tarefa selecionada.
- **deleteTask():** Exclui a tarefa selecionada.

Exemplo de inserção de dados no banco:

```
try (Connection conn = connectToDatabase()) {
    if (conn != null) {
        String sql = "INSERT INTO tasks (title, description, due_date, completed) VALUES (?, ?, ?, ?)";
        PreparedStatement stmt = conn.prepareStatement(sql);
        stmt.setString(parameterIndex:1, title);
        stmt.setString(parameterIndex:2, description);
        stmt.setString(parameterIndex:3, dueDate);
        stmt.setBoolean(parameterIndex:4, status);
        stmt.executeUpdate();
    }
}
```

9. Arquivos Adicionais

No projeto, além do código fonte da aplicação, há outros arquivos importantes:

- **mysql-connector-j-8.4.0.jar**: Este arquivo JAR é o driver JDBC utilizado para permitir a comunicação entre a aplicação Java e o banco de dados MySQL. Sem ele, o código não seria capaz de realizar operações no banco de dados.
- **TodoListApp.class**: Este é o arquivo compilado da classe principal do aplicativo. Ele contém o bytecode que pode ser executado pelo Java Runtime Environment (JRE).
- **TodoListApp\$Task.class**: Este é o arquivo compilado da classe interna **Task**, responsável por representar uma tarefa no sistema.

10. Considerações Finais

Este sistema de gerenciamento de tarefas é uma implementação simples, mas eficaz, para demonstrar a integração de uma interface gráfica Java com um banco de dados MySQL. O código pode ser expandido para incluir funcionalidades adicionais, como ordenação de tarefas por data ou por título, e melhorias na interface gráfica, como a inclusão de ícones e temas personalizados.

A integração com o MySQL garante que as tarefas sejam persistidas de forma eficiente, mesmo após o fechamento do aplicativo.

11. Referências

- SUN Microsystems. Java™ Platform, Standard Edition. [Online] Disponível em: <https://www.oracle.com/java/>
- MySQL. MySQL 8.0 Documentation. [Online] Disponível em: <https://dev.mysql.com/doc/>

Trabalho realizado pelos alunos:

- Leonardo Mendes Gomes da Silva
- Vinicius Alves Simonetti
- Elias Henrique Rodrigues Vieira