Manual de Execução - Sistema Java com Conexão JDBC

Este documento descreve os requisitos mínimos e os passos necessários para executar um projeto Java com interface gráfica (Swing) integrado a um banco de dados MySQL utilizando JDBC (Java Database Connectivity).

# 1. Requisitos Mínimos

Para executar o sistema corretamente, é necessário ter os seguintes requisitos instalados e configurados:

- Java Development Kit (JDK) 8 ou superior

- Ambiente de desenvolvimento (recomendado: IntelliJ IDEA, Eclipse ou NetBeans)

- Sistema operacional Windows, Linux ou macOS

- MySQL Server instalado e em execução

- Driver JDBC para MySQL (mysql-connector-java-x.x.x.jar)

- Variável de ambiente JAVA\_HOME configurada corretamente

- Caminho do JDK adicionado à variável PATH

# 2. Configuração do Banco de Dados

1. Instale e inicie o MySQL Server.

2. Crie um banco de dados no MySQL com o nome apropriado, por exemplo: `sistema\_cursos`

3. Execute os scripts SQL para criar as tabelas necessárias, como `alunos`, `cursos`, `matriculas`, etc.

4. Certifique-se de que o usuário e a senha usados no código Java estejam corretos e tenham permissão de acesso.

# 3. Estrutura de Conexão JDBC

O código Java utiliza a biblioteca JDBC para conectar-se ao banco de dados. Um exemplo de conexão pode ser:  
  
```java  
String url = "jdbc:mysql://localhost:3306/sistema\_cursos";  
String usuario = "root";  
String senha = "sua\_senha";  
Connection conexao = DriverManager.getConnection(url, usuario, senha);  
```  
  
Certifique-se de importar o driver JDBC com:  
`Class.forName("com.mysql.cj.jdbc.Driver");`

# 4. Como Executar o Projeto

1. Importe o projeto em sua IDE preferida.

2. Adicione o driver JDBC (arquivo `.jar`) ao classpath do projeto.

3. Verifique se o banco de dados está rodando e acessível.

4. Execute a classe principal que inicia a interface gráfica do sistema.

5. Utilize as funcionalidades da interface para interagir com o banco de dados.

# 5. Dicas Adicionais

- Certifique-se de que a porta do MySQL (geralmente 3306) está liberada e não bloqueada por firewall.

- Utilize `try/catch` para tratar erros de conexão e exibir mensagens ao usuário.

- Faça backup do banco de dados antes de testes importantes.

- Mantenha o driver JDBC atualizado para evitar problemas de compatibilidade.

# Requisitos Mínimos para Executar um Projeto Java com Conexão JDBC

Para executar um projeto Java com conexão a banco de dados utilizando JDBC (Java Database Connectivity), como o sistema de biblioteca apresentado, são necessários os seguintes requisitos:  
  
1. Requisitos de Software  
  
- Java Development Kit (JDK)  
 - Versão recomendada: JDK 8 ou superior.  
 - O JDK deve estar corretamente instalado e configurado na variável de ambiente JAVA\_HOME.  
  
- IDE Java (opcional, mas recomendada)  
 - Exemplo: IntelliJ IDEA, Eclipse, NetBeans ou VS Code com extensões Java.  
  
- MySQL Server  
 - Versão recomendada: 5.7 ou superior.  
 - O serviço do banco de dados deve estar ativo na máquina local (localhost) ou em servidor remoto.  
  
- Driver JDBC do MySQL  
 - Nome do driver: mysql-connector-java (ex: mysql-connector-java-8.0.33.jar).  
 - Esse arquivo .jar precisa estar incluído no classpath do projeto.  
  
2. Configuração da Base de Dados  
  
- Um banco de dados chamado biblioteca deve ser criado.  
- As tabelas e relações devem ser criadas conforme o script SQL fornecido.  
- O usuário do MySQL (root no exemplo) deve ter permissões para acesso e manipulação do banco.  
  
3. Estrutura Mínima no Código Java  
  
- Classe de Conexão  
 Exemplo: ConnectionFactory.java, que encapsula a lógica de conexão com o banco utilizando DriverManager.  
  
 Class.forName("com.mysql.cj.jdbc.Driver");  
 Connection conn = DriverManager.getConnection("jdbc:mysql://localhost/biblioteca", "root", "senha");  
  
- Tratamento de Exceções  
 É essencial tratar SQLException e ClassNotFoundException de forma adequada para evitar falhas em tempo de execução.  
  
- Pacotes Organizacionais  
 É recomendado organizar as classes em pacotes, como factory, dao, model e view.  
  
4. Execução  
  
- Compilar o projeto com todas as dependências (especialmente o .jar do MySQL Connector).  
- Garantir que o servidor MySQL esteja ativo.  
- Executar a aplicação a partir de uma main() ou interface gráfica (como Swing, JavaFX etc.).

5. Considerações de Segurança  
  
- Evite deixar senhas explícitas no código. Utilize arquivos .properties ou variáveis de ambiente para armazenar dados sensíveis.  
- Feche sempre as conexões ao final de cada operação com o banco (uso de try-with-resources é recomendado).