📌 Passo a Passo: Criando um Projeto Java no NetBeans com SQLite

Este guia cobre criação do projeto, configuração do SQLite JDBC, criação do banco e testes.

1. Criar o Projeto Java no NetBeans

- 1. Abra o NetBeans.
- 2. No menu superior, clique em File > New Project....
- 3. Em Categories, escolha Java with Ant.
- 4. Em Projects, escolha Java Application e clique em Next.
- 5. Em **Project Name**, digite: ProjetoSQLiteCrud.
- 6. Marque Create Main Class e digite: com.meuprojeto.Main.
- 7. Clique em Finish.
- 🔷 O NetBeans criará a estrutura do projeto dentro da pasta escolhida.

2. Adicionar o Driver SQLite ao Projeto

- 1. No painel "Projects", clique com o botão direito sobre ProjetoSQLite.
- 2. Escolha Properties.
- 3. No menu à esquerda, clique em Libraries.
- 4. No lado direito, clique em "Add JAR/Folder...".
- 5. **Selecione o arquivo** sqlite-jdbc-3.49.1.0.jar e clique em **Open**.
- 6. Clique em **OK** para salvar.

Agora o driver do SQLite está configurado! 🜠

3. Criar a Classe de Conexão com o Banco

Agora vamos criar uma classe que conecta ao SQLite.

- No NetBeans, clique com o botão direito em "Source Packages" > "com.meuprojeto".
- 2. Escolha New > Java Class.
- 3. Nomeie como ConexaoSQLite e clique em Finish.
- 4. Substitua o conteúdo do arquivo ConexaoSQLite.java pelo seguinte código:

```
package com.meuprojeto;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class ConexaoSQLite {
    private static final String URL = "jdbc:sqlite:produtos.db";

    public static Connection conectar() {
        try {
            return DriverManager.getConnection(URL);
        } catch (SQLException e) {
            System.out.println("Erro ao conectar: " + e.getMessage());
            return null;
        }
    }
}
```

O que esse código faz?

- Conecta ao banco produtos.db.
- Retorna uma conexão ativa ou null se houver erro.

4. Criar o Banco de Dados SQLite

Agora precisamos criar o banco produtos.db e a tabela produtos.

- No NetBeans, clique com o botão direito em "Source Packages" > "com.meuprojeto".
- 2. Escolha New > Java Class.
- 3. Nomeie como CriarBanco e clique em Finish.
- 4. Substitua o conteúdo do arquivo CriarBanco.java pelo seguinte código:

```
package com.meuprojeto;
import java.sql.Connection;
import java.sql.Statement;
import java.sql.SQLException;
public class CriarBanco {
 public static void main(String[] args) {
   Connection conn = ConexaoSQLite.conectar();
   if (conn == null) {
     System.out.println("Erro ao conectar com o banco de dados.");
     return;
   String sql = "CREATE TABLE IF NOT EXISTS produtos ("
        + "id INTEGER PRIMARY KEY AUTOINCREMENT, "
        + "nome TEXT NOT NULL, "
        + "preco REAL NOT NULL)";
   try (Statement stmt = conn.createStatement()) {
     stmt.execute(sql);
     System.out.println("Banco de dados criado com sucesso!");
```

```
} catch (SQLException e) {
      System.out.println("Erro ao criar o banco: " + e.getMessage());
    }
  }
}
```

Execute esse código UMA VEZ para criar o banco.

Como executar:

- No NetBeans, clique com o botão direito em CriarBanco.java e escolha "Run File" (Shift + F6).
- Se aparecer "Banco de dados criado com sucesso!", deu certo! 🦫



5. Criar o Programa Principal

Agora criamos o programa principal para interagir com o usuário.

- 1. No NetBeans, abra Main.java dentro do pacote com.meuprojeto.
- 2. Substitua o código por este:

```
package com.meuprojeto;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.Scanner;
public class Main {
  public static void main(String[] args) {
   Scanner scanner = new Scanner(System.in);
   Connection conn = ConexaoSQLite.conectar();
   if (conn == null) {
     System.out.println("Erro ao conectar com o banco de dados.");
     return;
   int opcao;
     System.out.println("\n=== Menu Principal ===");
     System.out.println("1. Inserir produto");
     System.out.println("2. Listar produtos");
     System.out.println("3. Editar produto");
     System.out.println("4. Excluir produto");
     System.out.println("5. Buscar produto por ID");
     System.out.println("6. Sair");
     System.out.print("Escolha uma opção: ");
     opcao = scanner.nextInt();
     switch (opcao) {
         inserirProduto(conn, scanner);
         break;
       case 2:
         listarProdutos(conn);
         break:
       case 3:
         editarProduto(conn, scanner);
         break;
       case 4:
```

```
excluirProduto(conn, scanner);
       break;
     case 5:
       buscarProdutoPorld(conn, scanner);
     case 6:
       System.out.println("Saindo do sistema...");
       break:
     default:
       System.out.println("Opção inválida. Tente novamente.");
 } while (opcao != 6);
 try {
   if (conn != null) {
     conn.close();
 } catch (SQLException e) {
   System.out.println("Erro ao fechar conexão: " + e.getMessage());
 scanner.close();
public static void inserirProduto(Connection conn, Scanner scanner) {
 System.out.println("\n--- Inserir Novo Produto ---");
 System.out.print("Nome do produto: ");
 scanner.nextLine(); // Limpar buffer
 String nome = scanner.nextLine();
 System.out.print("Preço: ");
 double preco = scanner.nextDouble();
 String sql = "INSERT INTO produtos(nome, preco) VALUES(?, ?)";
 try (PreparedStatement pstmt = conn.prepareStatement(sql)) {
   pstmt.setString(1, nome);
   pstmt.setDouble(2, preco);
   pstmt.executeUpdate();
   System.out.println("Produto inserido com sucesso!");
 } catch (SQLException e) {
   System.out.println("Erro ao inserir produto: " + e.getMessage());
public static void listarProdutos(Connection conn) {
 String sql = "SELECT * FROM produtos ORDER BY id";
 try (PreparedStatement pstmt = conn.prepareStatement(sql); ResultSet rs = pstmt.executeQuery()) {
   System.out.println("\n--- Lista de Produtos ---");
   if (!rs.isBeforeFirst()) {
     System.out.println("Nenhum produto cadastrado.");
     return;
   }
   while (rs.next()) {
     System.out.println("ID: " + rs.getInt("id")
         + " | Nome: " + rs.getString("nome")
         + " | Preço: R$" + String.format("%.2f", rs.getDouble("preco")));
 } catch (SQLException e) {
   System.out.println("Erro ao listar produtos: " + e.getMessage());
public static void editarProduto(Connection conn, Scanner scanner) {
 System.out.println("\n--- Editar Produto ---");
 listarProdutos(conn);
 System.out.print("\nDigite o ID do produto que deseja editar: ");
 int id = scanner.nextInt();
 // Verificar se o produto existe
 if (!produtoExiste(conn, id)) {
```

```
System.out.println("Produto com ID " + id + " não encontrado.");
   return;
 System.out.print("Novo nome (deixe em branco para não alterar): ");
 scanner.nextLine(); // Limpar buffer
 String novoNome = scanner.nextLine();
 System.out.print("Novo preço (digite 0 para não alterar): ");
 double novoPreco = scanner.nextDouble();
 StringBuilder sql = new StringBuilder("UPDATE produtos SET");
 boolean temAlteracao = false;
 if (!novoNome.isEmpty()) {
   sql.append("nome = ?");
   temAlteracao = true;
 if (novoPreco > 0) {
   if (temAlteracao) {
     sql.append(", ");
   sql.append("preco = ?");
   temAlteracao = true;
 if (!temAlteracao) {
   System.out.println("Nenhuma alteração foi informada.");
   return;
 sql.append("WHERE id = ?");
 try (PreparedStatement pstmt = conn.prepareStatement(sql.toString())) {
   int paramIndex = 1;
   if (!novoNome.isEmpty()) {
     pstmt.setString(paramIndex++, novoNome);
   if (novoPreco > 0) {
     pstmt.setDouble(paramIndex++, novoPreco);
   pstmt.setInt(paramIndex, id);
   int linhasAfetadas = pstmt.executeUpdate();
   if (linhasAfetadas > 0) {
     System.out.println("Produto atualizado com sucesso!");
   } else {
     System.out.println("Nenhum produto foi atualizado.");
 } catch (SQLException e) {
   System.out.println("Erro ao editar produto: " + e.getMessage());
public static void excluirProduto(Connection conn, Scanner scanner) {
 System.out.println("\n--- Excluir Produto ---");
 listarProdutos(conn);
 System.out.print("\nDigite o ID do produto que deseja excluir: ");
 int id = scanner.nextInt();
 // Verificar se o produto existe
 if (!produtoExiste(conn, id)) {
   System.out.println("Produto com ID " + id + " não encontrado.");
   return;
 }
```

}

```
System.out.print("Tem certeza que deseja excluir este produto? (S/N): ");
 scanner.nextLine(); // Limpar buffer
 String confirmacao = scanner.nextLine().toUpperCase();
 if (!confirmacao.equals("S")) {
   System.out.println("Operação cancelada.");
   return:
 String sql = "DELETE FROM produtos WHERE id = ?";
 try (PreparedStatement pstmt = conn.prepareStatement(sql)) {
   pstmt.setInt(1, id);
   int linhasAfetadas = pstmt.executeUpdate();
   if (linhasAfetadas > 0) {
     System.out.println("Produto excluído com sucesso!");
   } else {
     System.out.println("Nenhum produto foi excluído.");
 } catch (SQLException e) {
   System.out.println("Erro ao excluir produto: " + e.getMessage());
public static void buscarProdutoPorld(Connection conn, Scanner scanner) {
 System.out.println("\n--- Buscar Produto por ID ---");
 System.out.print("Digite o ID do produto: ");
 int id = scanner.nextInt();
 String sal = "SELECT * FROM produtos WHERE id = ?":
 try (PreparedStatement pstmt = conn.prepareStatement(sql)) {
   pstmt.setInt(1, id);
   ResultSet rs = pstmt.executeQuery();
   if (rs.next()) {
     System.out.println("\n--- Produto Encontrado ---");
     System.out.println("ID: " + rs.getInt("id"));
     System.out.println("Nome: " + rs.getString("nome"));
     System.out.println("Preço: R$" + String.format("%.2f", rs.getDouble("preco")));
   } else {
     System.out.println("Produto com ID " + id + " não encontrado.");
 } catch (SQLException e) {
   System.out.println("Erro ao buscar produto: " + e.getMessage());
private static boolean produtoExiste(Connection conn, int id) {
 String sql = "SELECT 1 FROM produtos WHERE id = ?";
 try (PreparedStatement pstmt = conn.prepareStatement(sql)) {
   pstmt.setInt(1, id);
   ResultSet rs = pstmt.executeQuery();
   return rs.next();
 } catch (SQLException e) {
   System.out.println("Erro ao verificar produto: " + e.getMessage());
   return false;
```

Esse código permite:

- CRUD de produtos no SQLite.
- Buscar produto cadastrados.
- Sair do programa.

6. Rodar o Programa

- 1. No NetBeans, clique com o botão direito em Main.java e escolha "Run File" (Shift + F6).
- 2. Escolha as opções do menu para testar a inserção e listagem de produtos.

@ Resumo

- Criamos o projeto Projeto SQLite no NetBeans
- Adicionamos o driver sqlite-jdbc-3.49.1.0.jar
- Criamos a conexão com o SQLite (ConexaoSQLite.java)
- Criamos o banco e a tabela (CriarBanco.java)
- Criamos o menu para inserir e listar produtos (Main.java)
- Executamos o programa e testamos!

Agora seu projeto está pronto! 🏂

