
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: ProjetoSQLite.
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.

1. 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**.

1. 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;
        do {
            System.out.println("\n=== Menu ===");
            System.out.println("1. Inserir produto");
            System.out.println("2. Listar produtos");
            System.out.println("3. Sair");
            System.out.print("Escolha uma opção: ");
            opcao = scanner.nextInt();

            switch (opcao) {
                case 1:
                    inserirProduto(conn, scanner);
                    break;
                case 2:
                    listarProdutos(conn);
                    break;
                case 3:
                    System.out.println("Saindo...");
                    break;
                default:

```

```

        System.out.println("Opção inválida.");
    }
} while (opcao != 3);
}

public static void inserirProduto(Connection conn, Scanner scanner) {
    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";
    try (PreparedStatement pstmt = conn.prepareStatement(sql);
        ResultSet rs = pstmt.executeQuery()) {

        System.out.println("\n--- Lista de Produtos ---");
        while (rs.next()) {
            System.out.println("ID: " + rs.getInt("id") +
                               " | Nome: " + rs.getString("nome") +
                               " | Preço: " + rs.getDouble("preco"));
        }
    } catch (SQLException e) {
        System.out.println("Erro ao listar produtos: " + e.getMessage());
    }
}
}

```

✅ Esse código permite:

- Inserir produtos no SQLite.
- Listar produtos 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.
-

- Criamos o projeto ProjetoSQLite 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! 🎉

OBS.:

- 1) Arquivo .jar para download do SQLite:
<https://github.com/xerial/sqlite-jdbc/releases>
- 2) Você deve adicionar o **sqlite-jdbc-3.49.1.0.jar** no **Classpath**, e não no Modulepath.

📌 Motivo para escolher Classpath:

O **Classpath** é utilizado para carregar bibliotecas externas **não-modulares**, como o SQLite JDBC.

O **Modulepath** é usado para projetos Java que utilizam módulos (module-info.java), o que **não é necessário** para este projeto simples com SQLite.

📌 Como adicionar ao Classpath no NetBeans

Clique com o botão direito no nome do seu projeto no NetBeans.

Escolha **"Properties"**.

No menu esquerdo, clique em **"Libraries"**.

Na aba "Compile", clique em **"Add JAR/Folder..."**.

Selecione o arquivo sqlite-jdbc-3.49.1.0.jar.

Confirme que ele foi adicionado ao Classpath e clique em **"OK"**.