
Newton
Quem se prepara, não para.

Newton
Quem se prepara, não para.

Linguagens de Programação

3º período

Prof. Dr. João Paulo Aramuni

Sumário

- Java
    - Conexão com Banco de Dados

Sumário

- Java
    - **Conexão com Banco de Dados**

- Conexão com Banco de Dados
  - Construiremos um exemplo simples de conexão com o banco de dados **SQLite**.
  - Para esse exemplo, usaremos um modelo de banco de dados de uma **biblioteca acadêmica**.
  - Faremos uma consulta do tipo **SELECT** para recuperar o título de um livro.

- BD Biblioteca

  DDL - Data Definition Language

```sql
-- Criando a tabela Livro
CREATE TABLE [Livro]
(
    [LivroId] INTEGER  NOT NULL,
    [Titulo] NVARCHAR(160)  NOT NULL,
    [Autor] NVARCHAR(160)  NOT NULL,
    [ISBN] NVARCHAR(160)  NOT NULL,
    [Quantidade] INTEGER  NOT NULL,
    CONSTRAINT [PK_Livro] PRIMARY KEY ([LivroId])
);

CREATE UNIQUE INDEX [IPK_Livro] ON [Livro]([LivroId]);

-- Criando a tabela Aluno
CREATE TABLE [Aluno]
(
    [Matricula] INTEGER  NOT NULL,
    [Nome] NVARCHAR(160)  NOT NULL,
    CONSTRAINT [PK_Aluno] PRIMARY KEY ([Matricula])
);

CREATE UNIQUE INDEX [IPK_Aluno] ON [Aluno]([Matricula]);

-- Criando a tabela Emprestimo
CREATE TABLE [Emprestimo]
(
    [EmprestimoId] INTEGER  NOT NULL,
    [LivroId] INTEGER  NOT NULL,
    [Matricula] INTEGER  NOT NULL,
    [DataHora_Emprestimo] DATETIME NOT NULL,
    CONSTRAINT [PK_Emprestimo] PRIMARY KEY ([EmprestimoId]),
    CONSTRAINT [FK_Emprestimo_Livro] FOREIGN KEY ([LivroId]) REFERENCES Livro(LivroId),
    CONSTRAINT [FK_Emprestimo_Aluno] FOREIGN KEY ([Matricula]) REFERENCES Aluno(Matricula)
);

CREATE UNIQUE INDEX [IPK_Emprestimo] ON [Emprestimo]([EmprestimoId]);

-- Para deletar as tabelas:
DROP TABLE Livro;
DROP TABLE Aluno;
DROP TABLE Emprestimo;
```

- BD Biblioteca

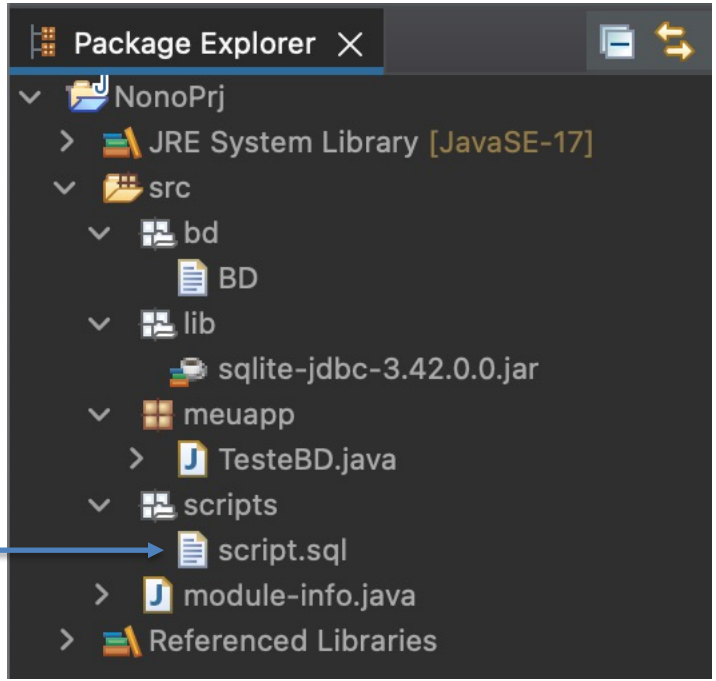
  DML - Data Manipulation Language e

  DQL - Data Query Language

```sql
-- CRUD (Create, Read, Update and Delete)

-- INSERT

-- Inserindo um registro na tabela Livro
INSERT INTO Livro (LivroId, Titulo, Autor, ISBN, Quantidade)
VALUES (1, 'Introdução aos Fundamentos da Computação', 'Newton José Vieira', 'ISBN 123456', 10);

-- Inserindo um registro na tabela Aluno
INSERT INTO Aluno (Matricula, Nome)
VALUES (1, 'Aramuni');

-- Inserindo um registro na tabela Emprestimo
INSERT INTO Emprestimo (EmprestimoId, LivroId, Matricula, DataHora_Emprestimo)
VALUES (1, 1, 1, '2023-07-08 19:00:00');

-- SELECT
SELECT * FROM Livro WHERE Titulo LIKE 'Introdução%';
SELECT * FROM Aluno WHERE Nome LIKE 'Ara%';
SELECT * FROM Emprestimo WHERE LivroId LIKE '%1'

-- UPDATE
UPDATE Livro SET Titulo = 'Introdução aos Compiladores' WHERE LivroId  = 1;
UPDATE Aluno SET Nome = 'João Paulo' WHERE Matricula = 1;
UPDATE Emprestimo SET DataHora_Emprestimo = '2023-07-08 22:00:00' WHERE EmprestimoId = 1;

-- DELETE
DELETE FROM Emprestimo WHERE EmprestimoId = 1;
DELETE FROM Aluno WHERE Matricula = 1;
DELETE FROM Livro WHERE LivroId = 1;

-- INNER JOIN
SELECT e.EmprestimoId, l.Titulo, a.Nome
FROM Emprestimo e
INNER JOIN Livro l ON e.LivroId = l.LivroId
INNER JOIN Aluno a ON e.Matricula = a.Matricula;
```

Java

- Conexão com Banco de Dados
  - Estrutura do Projeto



DDL, DML e DQL estão aqui
apenas para backup

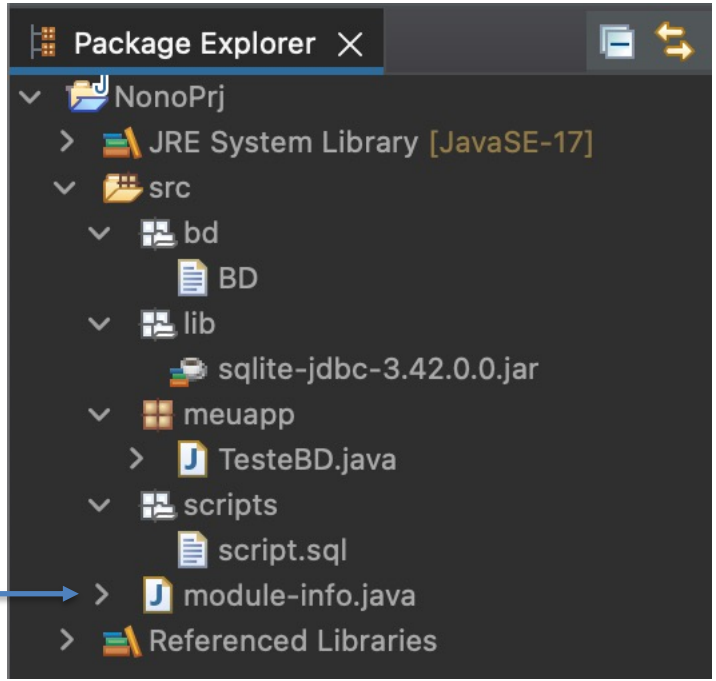
Java

- Conexão com Banco de Dados
  - Estrutura do Projeto

TesteBD.java | module-info.java

```
1 module NonoPrj {
2     requires java.sql;
3 }
```

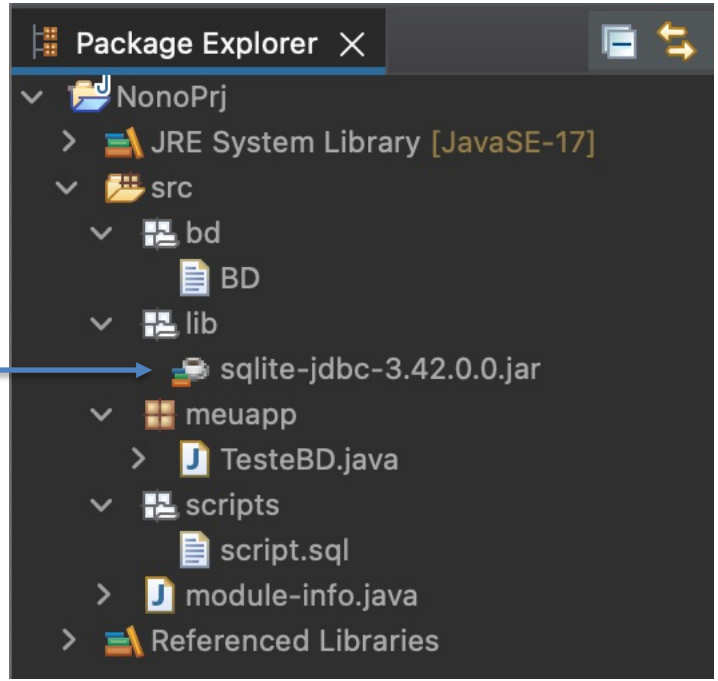
Adicionar a dependência do pacote java.sql no módulo



Java

- Conexão com Banco de Dados
  - Estrutura do Projeto

Fazer o download do Drive
JDBC para SQLite

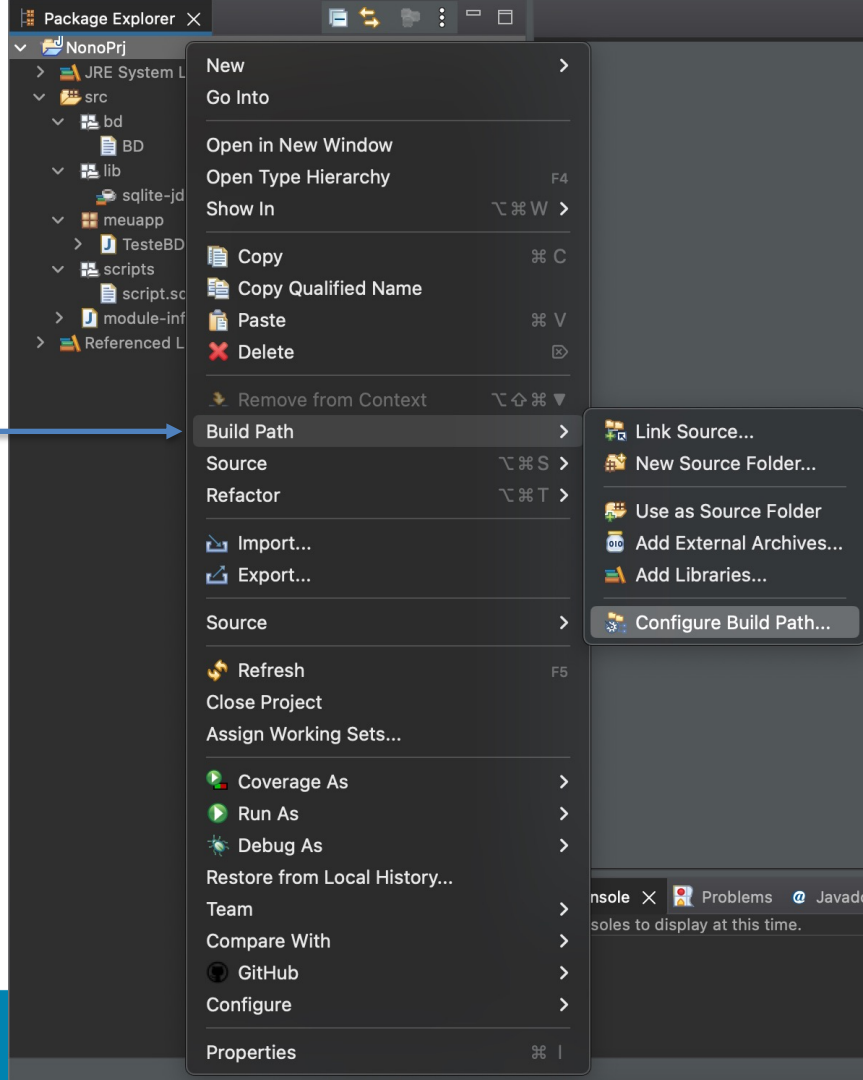


Link: https://mvnrepository.com/artifact/org.xerial/sqlite-jdbc/3.42.0.0

Java

- Conexão com Banco de Dados
  - Estrutura do Projeto

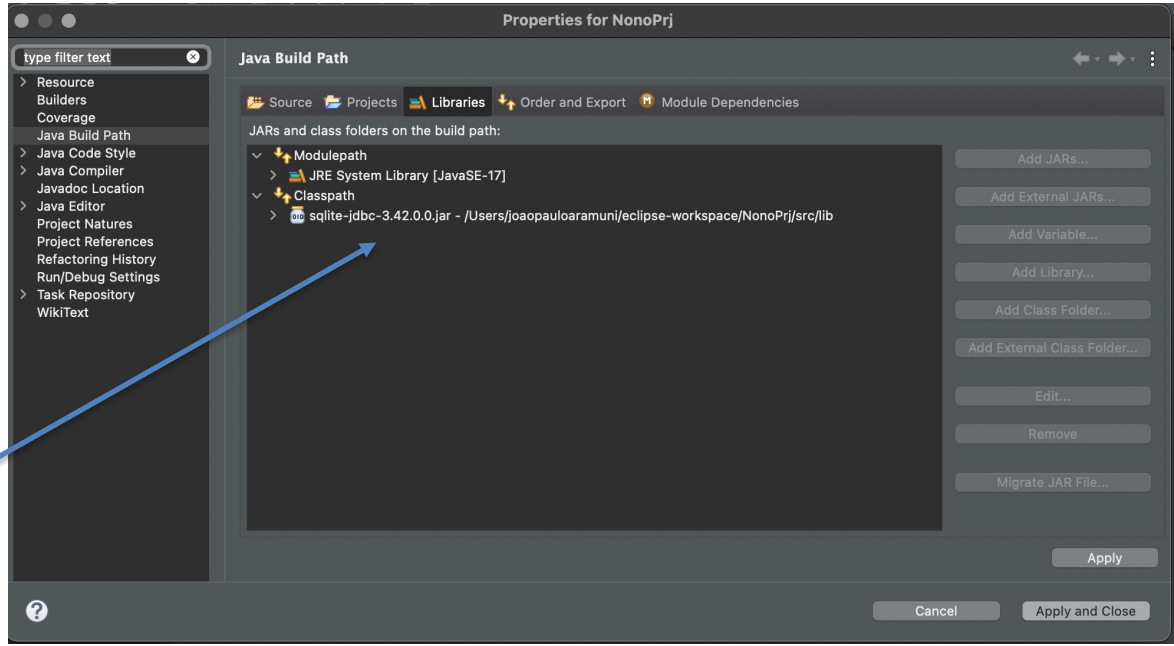
Adicionar o sqlite-jdbc-3.42.0.0.jar no **classpath** do projeto



Java

- Conexão com Banco de Dados
  - Estrutura do Projeto

Add External JARs...



Java

Newton
Quem se prepara, não para.

- Conexão com Banco de Dados
  - Um driver **JDBC** (Java Database Connectivity) é um componente de software que permite que aplicativos Java se comuniquem e interajam com diferentes bancos de dados por meio da linguagem Java.

  - O JDBC é uma API padrão do Java que fornece métodos e classes para estabelecer conexões com bancos de dados, enviar consultas (queries), receber resultados e executar operações relacionadas a bancos de dados.

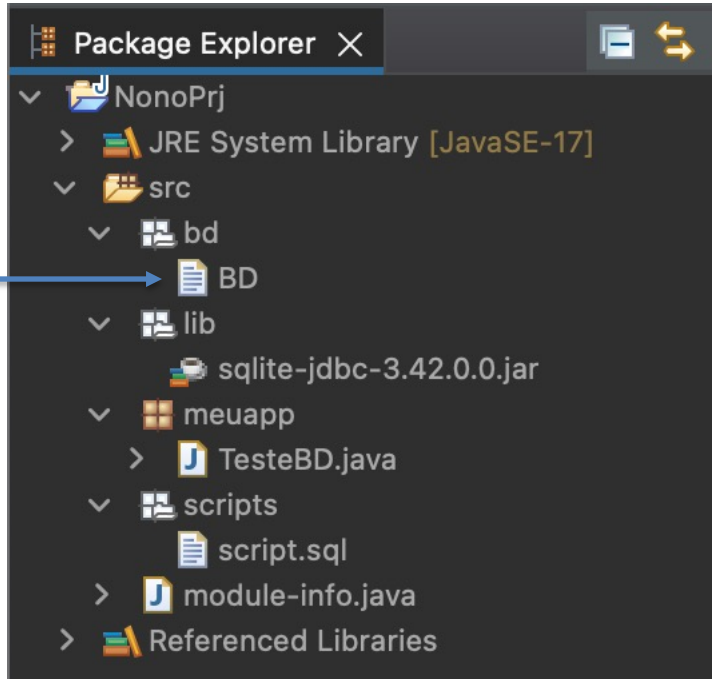
  - Os drivers JDBC atuam como uma ponte entre a aplicação Java e o sistema de gerenciamento de banco de dados (SGBD). Cada banco de dados possui sua própria implementação de driver JDBC, já que as especificações de como se comunicar com diferentes bancos de dados podem variar.

Java

Newton
Quem se prepara, não para.

- Conexão com Banco de Dados
  - Estrutura do Projeto

Arquivo do banco de dados
É comum ter a extensão
.db ou .sqlite

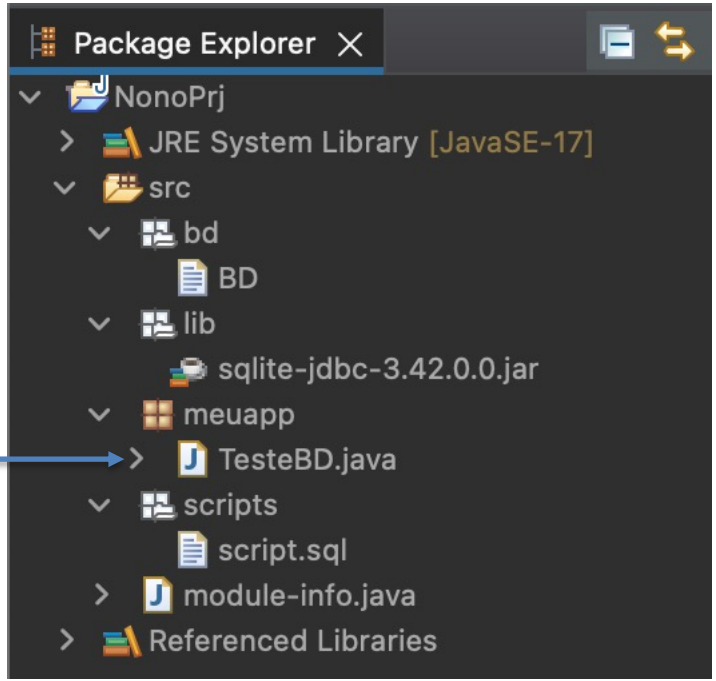


- É neste arquivo que estão as tabelas e os dados do banco de dados.
- Ele também serve para fazer backup e restore do BD.

Java

- Conexão com Banco de Dados
  - Estrutura do Projeto

Classe .java que realiza a conexão com o BD e executa as queries



Java

- Classe TesteBD.java

TesteBD.java

```java
package meuapp;

import java.sql.*;

public class TesteBD {
    public static void main(String[] args) {
        String url = "jdbc:sqlite:/Users/joaopauloaramuni/eclipse-workspace/NonoPrj/src/bd/BD";
        String usuario = "root";
        String senha = "";

        try {
            // Faz a conexão com o banco de dados
            Connection conexao = DriverManager.getConnection(url, usuario, senha);

            // Cria uma declaração (statement) para executar a query
            Statement statement = conexao.createStatement();

            // Define a query que deseja executar
            String query = "SELECT * FROM Livro WHERE Titulo LIKE 'Introdução%';";

            // Executa a query e obtém o resultado
            ResultSet resultado = statement.executeQuery(query);

            // Percorre o resultado e exibe os valores
            while (resultado.next()) {
                String titulo = resultado.getString("Titulo");
                // ... outros campos

                System.out.println("Titulo: " + titulo);
            }

            // Fecha os recursos
            resultado.close();
            statement.close();
            conexao.close();
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }
    }
}
```

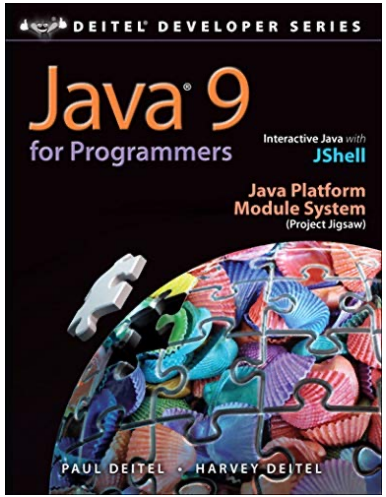
Java

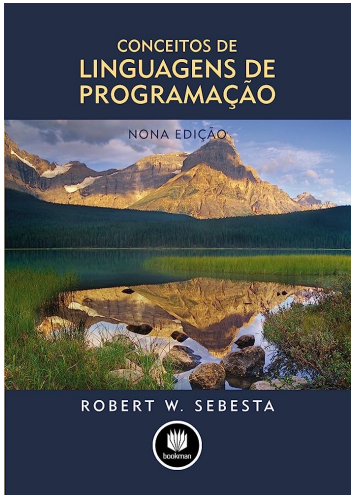
Newton
Quem se prepara, não para.

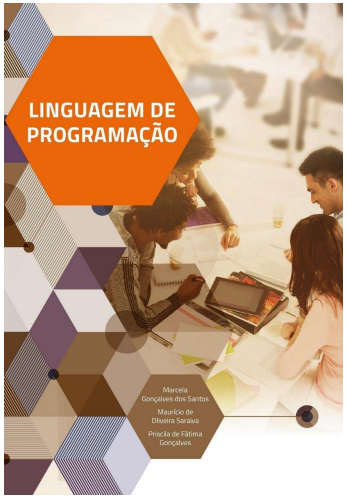
- Experimente também:
  - O Hibernate é um framework de mapeamento objeto-relacional (ORM) em Java.
  - Ele fornece uma maneira poderosa e eficiente de mapear **objetos Java** para tabelas em um banco de dados relacional, permitindo que os desenvolvedores trabalhem com dados de forma mais **orientada a objetos**, sem a necessidade de lidar **diretamente com SQL**.
  - https://hibernate.org/

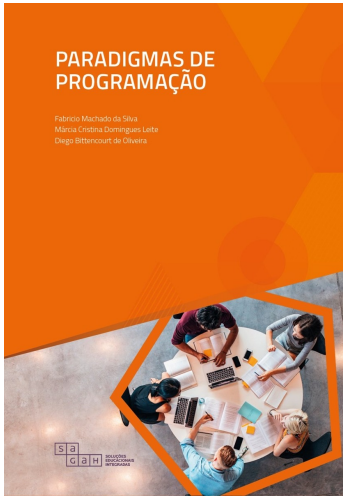
Referências

Newton
Quem se prepara, não para.









Obrigado!

joao.aramuni@newtonpaiva.br