

## Conectividade com Banco de Dados (BD) em Java

### 14.1. Java DataBase Connectivity (JDBC API)

- **JDBC** é a API padrão do Java para se comunicar com **bancos de dados relacionais**.
- Permite escrever código independente do banco de dados (portável).
- Fornecedores como Oracle, MySQL, PostgreSQL oferecem **drivers JDBC específicos**.

### Arquitetura JDBC

Inclui:

- **Driver Manager**: carrega o driver correto.
- **Driver JDBC**: responsável pela comunicação com o banco.
- **Connection, Statement, ResultSet**: principais interfaces/classes usadas para interação.

---

#### 14.1.2. Tipos de Drivers JDBC

Tipo	Descrição	Portabilidade	Status Atual
		e	
<b>Tipo 1</b>	JDBC-ODBC Bridge (usa ODBC)	✗ Baixa	<b>Obsoleto</b> desde Java 8
<b>Tipo 2</b>	Parcialmente Java + código nativo	✗ Baixa	Pouco usado
<b>Tipo 3</b>	Totalmente Java, usa middleware	✓ Alta	Raro hoje
<b>Tipo 4</b>	Totalmente Java, comunicação direta com o BD	✓ Alta	✓ <b>Mais comum</b>

---

### Exemplo Básico de Uso do JDBC (Tipo 4)

```
import java.sql.*;
```

```
public class ConexaoExemplo {
    public static void main(String[] args) {
        String url = "jdbc:mysql://localhost:3306/meubanco";
        String usuario = "root";
        String senha = "senha";

        try (Connection conn = DriverManager.getConnection(url, usuario, senha)) {
            System.out.println("Conexão bem-sucedida!");

            Statement stmt = conn.createStatement();
            ResultSet rs = stmt.executeQuery("SELECT * FROM clientes");

            while (rs.next()) {
                System.out.println(rs.getString("nome"));
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```