

Configuração de acesso ao banco de dados da sua aplicação.

**JDBC versus JNDI**

[[](https://www.linkedin.com/in/chmulato/)](https://www.linkedin.com/in/chmulato/)

**[Christian Mulato](https://www.linkedin.com/in/chmulato/)**

Desenvolvedor Java Sênior na Develcode

31 de março de 2024

A diferença entre uma conexão de banco de dados JDBC (**Java Database Connectivity**) e uma conexão de banco de dados JNDI (**Java Naming and Directory Interface**) está na forma como elas são configuradas e utilizadas:

**1. JDBC:**

- Definição: O JDBC é uma API do Java que possibilita que uma aplicação construída na linguagem consiga acessar um banco de dados configurado local ou remotamente.

- Componentes:

- Pacotes: A API JDBC é composta pelos pacotes java.sql e javax.sql, incluídos no JavaSE.

- Drivers: Os drivers JDBC são responsáveis pela conexão e interação com um banco de dados específico. Eles podem ser totalmente desenvolvidos em Java ou escritos de forma nativa, acessando outras bibliotecas ou drivers de sistema.

- Tipos de Drivers:

- Tipo 1 (JDBC-ODBC): Permite o acesso a drivers ODBC, um padrão consolidado para o acesso a bases de dados.

- Tipo 2: Implementa o protocolo do proprietário do banco de dados, transformando chamadas JDBC em chamadas do banco com o uso da API proprietária.

- Tipo 3: Faz a conversão das chamadas JDBC em outras chamadas do banco de dados, direcionando-as para uma camada intermediária de software (middleware).

- Tipo 4: É um driver puro Java que se comunica diretamente com o banco de dados, sem a necessidade de camadas intermediárias.

**2. JNDI:**

- Definição: O JNDI é uma API que permite que aplicações Java localizem e acessem objetos em um ambiente distribuído. Ele é frequentemente usado para configurar conexões com recursos, como bancos de dados, servidores de correio e serviços de diretório.

- Utilização:

- O JNDI é usado para obter referências a objetos, como conexões de banco de dados, a partir de um servidor de aplicação ou outro contexto.

- Ele permite que as configurações de conexão (como URL, usuário e senha) sejam centralizadas e gerenciadas externamente, facilitando a manutenção e a escalabilidade.

Em resumo, o **JDBC** é uma API para acessar bancos de dados diretamente, enquanto o **JNDI** é usado para obter referências a recursos, como conexões de banco de dados, de forma mais flexível e centralizada.