

VEM SER

Banco de Dados Oracle

JDBC

VEM SER

BÔNUS

Informática básica

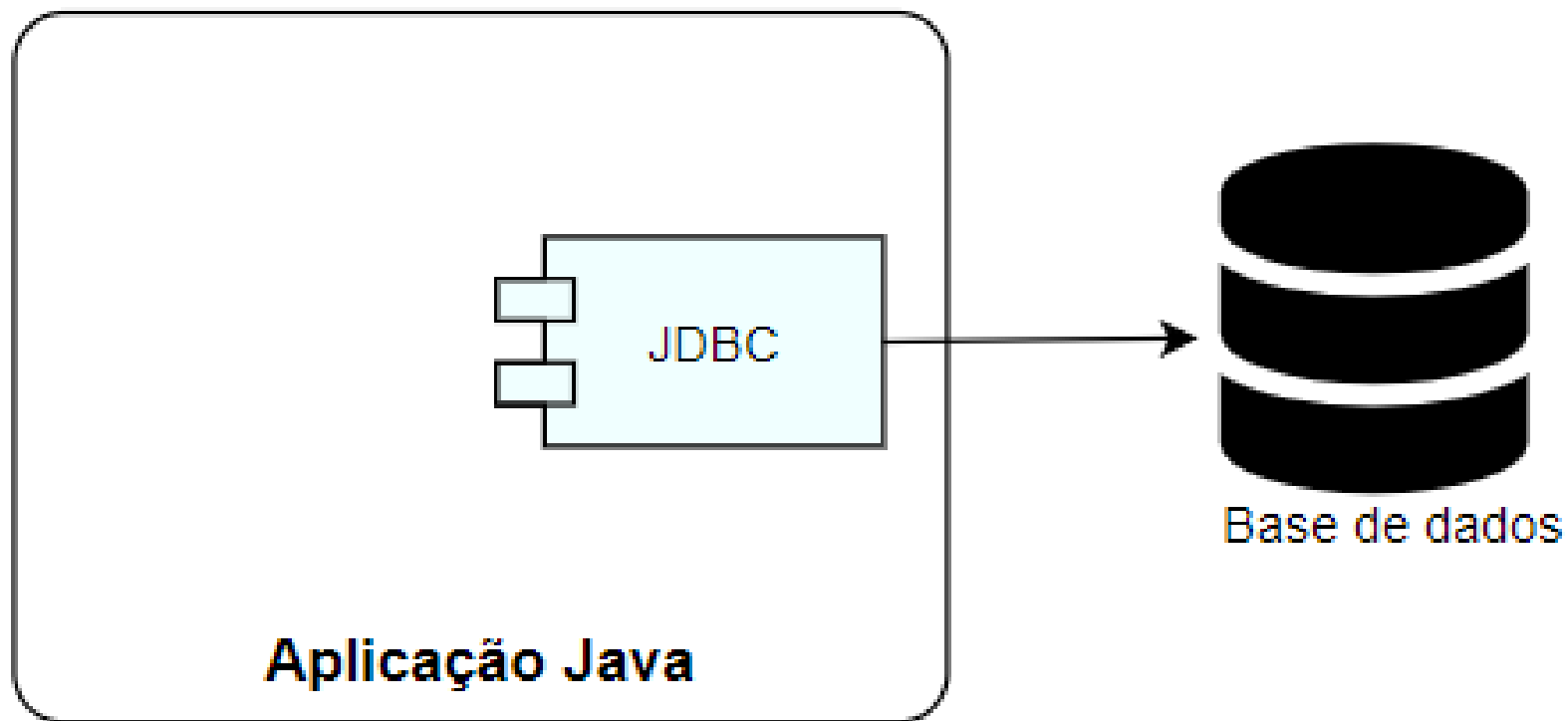
Sumário

- Java Database Connectivity (JDBC)
- Model View Controller (MVC)

JDBC

- A **JDBC** é uma **API do Java** que possibilita que uma aplicação construída na linguagem consiga acessar uma **fonte de dados** configurado local ou remotamente.
- A **API** é composta pelos pacotes **java.sql** e **javax.sql**, incluídos no **JavaSE**.
- Por meio das classes e interfaces fornecidas por esses dois pacotes, as pessoas podem desenvolver softwares que acessem qualquer fonte de dados, desde bancos relacionais até planilhas.

JDBC



- <https://www.alura.com.br/artigos/conhecendo-o-jdbc>

Drivers

- Um **Driver JDBC** é uma classe que implementa a interface `java.sql.Driver`.
 - Todo driver JDBC deve dar suporte mínimo às funcionalidades especificadas no padrão [ANSI2 SQL-92](#)
 - Através do **driver**, a aplicação Java **acessa** as implementações de classes e interfaces que vão permitir a execução dos comandos SQL em uma base de dados.
-
- <https://www.alura.com.br/artigos/conhecendo-o-jdbc>

Acesso à Base com JDBC

- 1) Definir o **Driver JDBC**.
- 2) Definir o **comando** de conexão do banco de dados, basicamente uma cadeia de caracteres com informações para conectar. É importante mencionar que a maneira de definir este comando varia entre bancos diferentes.
- 3) Fornecer o **endereço** do servidor, **usuário e senha** para nos conectar no banco de dados.

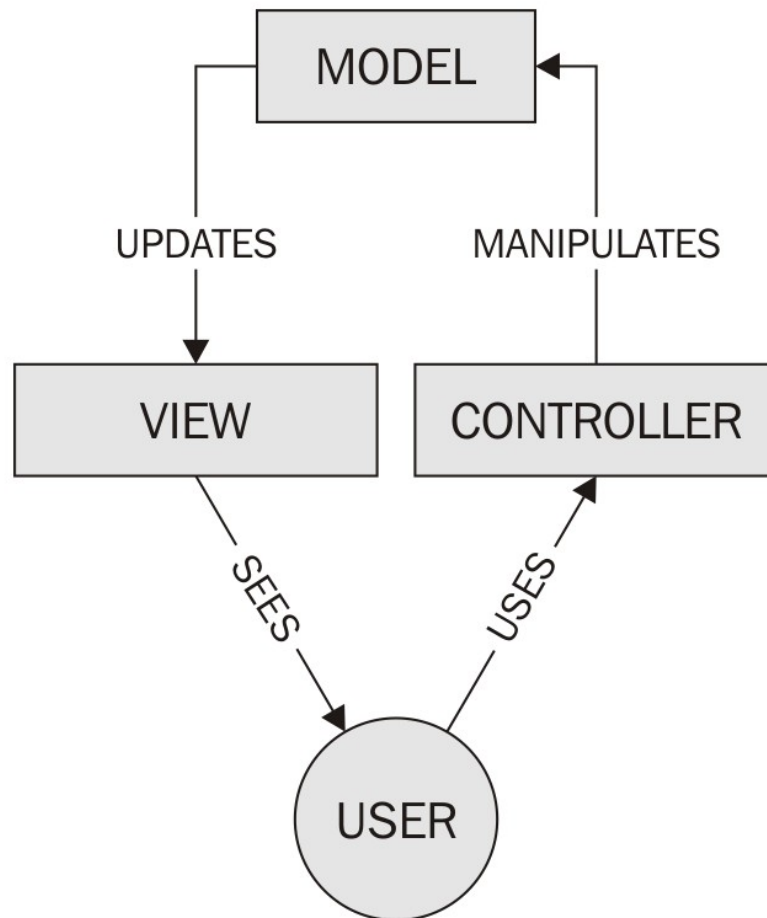
Principais Classes

- **Connection** - Representa uma sessão junto ao banco de dados desejado. Vimos a classe no trecho de código do exemplo anterior e vamos executar as instruções SQL dentro da conexão estabelecida.
 - **Statement** - Tem como objetivo a execução do comando SQL. Temos também a `PreparedStatement`, que pré-compila o comando e armazena o SQL em um objeto.
 - **ResultSet** - Esta interface tem por objetivo armazenar o retorno de uma consulta realizada no banco de dados. As informações das tabelas são recuperadas na sequência e podem ser iteradas em loops para a manipulação.
-
- <https://www.alura.com.br/artigos/conhecendo-o-jdbc>

Padrão MVC

- **A arquitetura MVC** é dividida em três componentes essenciais:
- **Model**
 - Sua responsabilidade é gerenciar e controlar a forma como os dados se comportam.
- **View**
 - Essa camada é responsável por apresentar as informações de forma visual ao usuário.
- **Controller**
 - A camada de controle é responsável por intermediar as requisições enviadas pelo View com as respostas fornecidas pelo Model.

Padrão MVC



- <https://www.lewagon.com/pt-BR/blog/o-que-e-padrao-mvc>



Vamos praticar!



Let's *Tech Up Together*