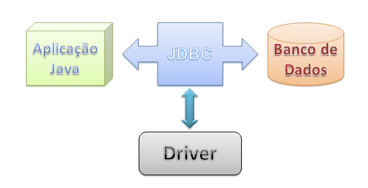
**JAVA com banco de Dados – JDBC**

**JDBC - Java Database Connectivity**

JDBC é uma API que reúne conjuntos de classes e interfaces escritas na linguagem Java na qual possibilita se conectar através de um driver específico do banco de dados desejado. Com esse driver pode-se executar instruções SQL de qualquer tipo de banco de dados relacional.

Para fazer a comunicação entre a Aplicação e o SGBDs é necessário possuir um driver para a conexão desejada. Geralmente, as empresas de SGBDs oferecem o driver de conexão que seguem a especificação JDBC para caso de algum desenvolvedor querer utilizar.

**Pacote java.sql**

Esse pacote oferece à biblioteca Java o acesso e processamento de dados em uma fonte de dados. As classes e interfaces mais importantes são:

A classe DriverManager oferece as seguintes Interfaces:

* Driver
* Connection
* Statement
* ResultSet
* PreparedStatement

**Driver**

Cada driver que se deseja usar precisa ser registrado nessa classe, pois oferece métodos estáticos para gerenciar um driver JDBC.

Para a aplicação reconhecer a comunicação com o banco de dados escolhido, é preciso obter um arquivo com a extensão .jar que geralmente se consegue através dos sites das empresas que distribuem o SGBD.



**A interface Connection**

Representa uma conexão ao banco de dados. Nessa interface são apresentados os métodos mais utilizados.

Após o driver estar inicializado, pode-se abrir uma conexão ao banco de dados através do método **getConection**. Esse método possui três sobrecargas apresentado na Listagem

Texto

Descrição gerada automaticamente

**Método PrepareStatement**

É usado para criar um objeto que representa a instrução SQL que será executada, sendo que é invocado através do objeto Connection.

Texto

Descrição gerada automaticamente

**A interface Statement**

Nesta interface são listados os métodos executeQuery e executeUpdate que são considerados os mais importantes referente a execução de uma query.

* **executeQuery**- Executa uma instrução **SQL**que retorna um único objeto ResultSet que indica o numero de registros da tabela.
* **executeUpdate**- Executa uma instrução SQL referente a um **INSERT, UPDATE e DELETE**. Esse método retorna a quantidade de registros que são afetados pela execução do comando SQL.

Texto

Descrição gerada automaticamente

**Interface ResultSet**

Representa o conjunto de resultados de uma tabela no banco de dados. Esse objeto mantém o cursor apontando para a sua linha atual de dados, sendo que seu início fica posicionado na primeira linha.

Além disso, esse objeto fornece métodos getters referentes aos tipos de dados como: getInt, getString, getDouble, getFloat, getLong entre outros. Com esses métodos são possíveis recuperar valores usando, por exemplo, o nome da coluna ou número do índice.

Interface gráfica do usuário, Aplicativo, Word

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente

Fonte: <https://www.devmedia.com.br/aprendendo-java-com-jdbc/29116>