



# 01. Introdução ao JDBC e o padrão Factory

📅 Date	@12/12/2022
📁 Categoria	Java e persistência
📖 Curso	Java e JDBC: trabalhando com um banco de dados

## Tópicos

- Introdução
- Ambiente e versões
- Usando um banco de dados
- API e interfaces
- Download do JAR
- A primeira conexão
- Exceção ao recuperar a conexão
- Faça que eu fiz
- O que aprendemos?

## API e interfaces

Qual a vantagem de utilizar uma API como o JDBC, baseada em interfaces, para fazer a comunicação entre o seu código e um banco de dados relacional?

- Facilidade ao trocar de um banco de dados para outro, sem muitas alterações. Só é necessário trocar o *driver* JDBC (JAR) para o banco específico e o JDBC se encarregará do resto.

## Exceção ao recuperar a conexão

Paulo desenvolveu o seguinte código, que abre e fecha uma conexão com o banco de dados:

```
public class TestaConexaoParaExercicio {
```

```
public static void main(String[] args) throws SQLException {
    // TODO Auto-generated method stub

    Connection con = DriverManager
        .getConnection("jdbc:mysql://localhost/loja_virtual?useTimezone=true&serverTimezone=UTC");

    con.close();
}
```

Mas na hora de executar, o mesmo recebe a seguinte exceção: *Access denied for user*.

O que Paulo precisa fazer para conseguir realizar com sucesso a sua operação?

- Precisa inserir o usuário e senha na string de conexão.
  - Se, ao configurar o banco de dados, foi cadastrado um usuário e senha, essas informações devem estar na string de conexão, por exemplo:

```
"jdbc:mysql://localhost/loja_virtual?useTimezone=true&serverTimezone=UTC", "root", "root"
```

---

## O que aprendemos?

Nesta aula, aprendemos que:

- Para acessar o banco de dados, precisamos de um **driver**
    - Um *driver* nada mais é do que uma biblioteca (JAR)
  - **JDBC** significa *Java DataBase Connectivity*
    - JDBC define uma camada de abstração entre a sua aplicação e o *driver* do banco de dados
    - Essa camada possui, na sua grande maioria, interfaces que o *driver* implementa
  - Para abrir uma conexão, devemos usar o método `getConnection`, da classe `DriverManager`.
    - O método `getConnection` recebe uma string de conexão JDBC, que define a URL, usuário, senha, etc.
-