



## 06. Camada de persistência com DAO

📅 Date	@13/12/2022
📁 Categoria	Java e persistência
📖 Curso	Java e JDBC: trabalhando com um banco de dados

### Tópicos

- Projeto da aula anterior
- Criando o modelo Produto
- DAO com inserção do produto
- Vantagens do DAO
- Listando no ProdutoDAO
- DAO's e construtor
- Faça que eu fiz
- O que aprendemos?

### Criando o modelo Produto

Nessa aula foi criado o modelo produto e uma classe com o objetivo de testa esse modelo

#### Produto

```
package br.com.alura.jdbc.modelo;

public class Produto{

    private Integer id;
    private String nome;
    private String descricao;

    public Produto(String nome, String descricao){
        this.nome = nome;
        this.descricao = descricao;
    }

    public String getNome(){
        return this.nome;
    }

    public String getDescricao(){
        return this.descricao;
    }

    public void setId(Integer id){
        this.id = id;
    }
}
```

```

    }

    @Override
    public String toString(){
        return String.format("O produto criado foi: %d, %s, %s", this.id, this.nome, this.descricao);
    }
}

```

## Classe para testa produto

```

import br.com.alura.jdbc.modelo.Produto;

public class TestaInsercaoComProduto{
    public static void main(String[] args){

        Produto comoda = new Produto("Cômoda", "Cômoda Vertical");

        try(Connection connection = new ConnectionFactory().recuperarConexao()){

            String sql = "INSERT INTO PRODUTO (NOME, DESCRICAO) VALUES (?,?)";

            try(PreparedStatement pstm = connection.prepareStatement(sql, Statement.RETURN_GENERATED_KEYS)){

                pstm.setString(1, comoda.getNome());
                pstm.setString(2, comoda.getDescricao());

                pstm.execute();

                try(ResultSet rst = pstm.getGenerateKeys()){
                    while(rst.next()){
                        comoda.setId(rst.getInt(1));
                    }
                }
            }
        }
        System.out.println(comoda);
    }
}

```

## DAO com inserção do produto

Nesta aula foi introduzido o conceito de um objeto DAO (*Data Access Object*), abaixo é possível ver um exemplo de um DAO do modelo produto:

```

package br.com.alura.jdbc;

import java.sql.Connection;
import java.sql.SQLException;
import java.sql.Statement;

import br.com.alura.jdbc.modelo.Produto;

public class ProdutoDAO{

    private Connection connection;

    public ProdutoDAO(Connection connection){
        this.connection = connection;
    }
}

```

```

public void salvar(Produto produto) throws SQLException{
    String sql = "INSERT INTO PRODUTO (NOME, DESCRICAO) VALUES (?,?)";

    try(PreparedStatement pstm = connection.prepareStatement(sql, Statement.RETURN_GENERATED_KEYS)){

        pstm.setString(1,produto.getNome());
        pstm.setString(2,produto.getDescricao());

        pstm.execute();

        try(ResultSet rst = pstm.getGenerateKeys()){
            while(rst.next()){
                produto.setId(rst.getInt(1));
            }
        }
    }
}

```

Trabalhando dessa forma todo o código que realiza o acesso de um produto está localizado em um único local, assim facilitando a manutenção do código.

Depois de utilizar a abordagem de criar um DAO de produtos a classe que foi criada anteriormente para testar o produto ficou bem mais “simples” de ser lida comparando com a sua versão anterior, ficando da seguinte forma:

```

import br.com.alura.jdbc.modelo.Produto;

public class TestaInsercaoComProduto{
    public static void main(String[] args){

        Produto comodo = new Produto("Cômuda", "Cômuda Vertical");

        try(Connection connection = new ConnectionFactory().recuperarConexao()){
            ProdutoDAO produtoDAO = new ProdutoDAO(connection);
            produtoDAO.salvar(comodo);
        }
    }
}

```

## Vantagens do DAO

Quais as vantagens de usar classes com o padrão DAO?

- Está ligada com a capacidade de isolar, em um único lugar, todo o código que acessa o seu repositório de dados. Dessa forma, evitaremos códigos repetidos e os desenvolvedores mexem em um único lugar.

## O que aprendemos?

Nesta aula, aprendemos que:

- Para cada tabela de domínio, temos uma classe de domínio
  - Por exemplo, a tabela `produtos` tem uma classe `Produto` associada

- Objetos dessa classe representa um registro na tabela
  - Para acessar a tabela, usaremos um padrão chamado **Data Access Object (DAO)**
    - Para cada classe de domínio, existe um DAO. Por exemplo, a classe `Produto` possui um `ProdutoDao`
    - Todos os métodos JDBC relacionados com produto devem estar encapsulados no `ProdutoDao`
-