



3° Sprint: Domain Driven Design

Nomes e RM:

Thiago Ulrych (97951)

Pamella Schimalesky Engholm (551600)

David Bryan Viana de Sales (551236)

Pedro Gomes Fernandes (551480)

Eduardo Violante (550364)

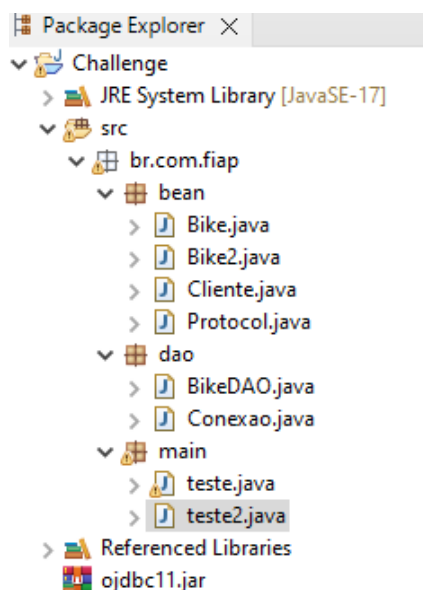
12/09/2023

Objetivo e Escopo do projeto:

Nós do projeto Portinho, buscamos atender todas as solicitações da Porto seguro, fazendo com que a solução seja simples e intuitiva para todos os clientes utilizarem de forma acessível para todos e que não seja um ponto de grande abandono da jornada, criando uma solução digital para o cliente que tenha uma bicicleta e consiga realizar a vistoria online, no momento da contratação, sem a necessidade de intervenção humana no processo. Fazendo com que as fraudes sejam minimizadas e manter a sinistralidade do produto controlada no canal digital! Para isso nessa segunda sprint mudamos totalmente o nosso método, fizemos uma classe Cliente, seguindo o que o foi pedido na sprint, utilizamos os Setters e os Getters, para informações do cliente, classe Protocolo, para gerar um número de protocolo para o cliente não assegurado, fizemos a Classe Bike, para informações da Bike que vai ser segurada, e a main que foi pedido na Sprint, nela vai ser recebido todas as informações do cliente e da Bike.

Breve descrição e Protótipo:

Neste código preparamos três pacotes, o pacote Bean, o pacote DAO e o pacote Main



Cada pacote acompanha suas devidas classes, no pacote bean temos as preparações para executar nas classe teste dentro do pacote main, No pacote DAO foi desenvolvido uma classe para linkar o Banco de dados(SQL) com o código java.

```
package br.com.fiap.dao;

import java.sql.Connection;

public class BikeDAO {
    private Connection con;

    public BikeDAO(Connection con) {
        setCon(con);
    }
    public Connection getCon() {
        return con;
    }
    public void setCon(Connection con) {
        this.con = con;
    }

    public String inserir(Bike2 bike2) {
        String sql = "insert into bike(marca,cor,descricao)values(?,?,?)";
        try {
            PreparedStatement ps = getCon().prepareStatement(sql);
            ps.setString(1, bike2.getMarca());
            ps.setString(2, bike2.getCor());
            ps.setString(3, bike2.getDescricao());
            if (ps.executeUpdate() > 0) {
                return "inserido com sucesso";
            }else {
                return "Erro ao inserir";
            }
        } catch (Exception e) {
            return e.getMessage();
        }
    }
}
```

Para executar o código devemos preparar a classe BikeDAO e introduzir a classe Conexão para mandar as informações para o banco de dados

```
package br.com.fiap.dao;

import java.sql.*;

public class Conexao {
    public static Connection abrirConexao() {
        Connection con = null;
        try {
            Class.forName("oracle.jdbc.driver.OracleDriver");
            String url = "jdbc:oracle:thin:@oracle.fiap.com.br:1521:ORCL";
            final String USER = "RM97951";
            final String PASS = "100703";
            con = DriverManager.getConnection(url, USER, PASS);
            System.out.println("Conexão Aberta");
        } catch (ClassNotFoundException e) {
            System.out.println(e.getMessage());
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        } catch (Exception e) {
            System.out.println(e.getMessage());
        }

        return con;
    }

    public static void fecharConexao(Connection con) {
        try {
            con.close();
            System.out.println("Conexão Fechada");
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        } catch (Exception e) {
            System.out.println(e.getMessage());
        }
    }
}
```

Feito isso agora é só executar a classe teste para certificar que a conexão foi aberta e as informações enviadas para o banco de dados, após isso a conexão deve ser fechada como na foto a seguir.

```
package br.com.fiap.main;

import java.sql.Connection;

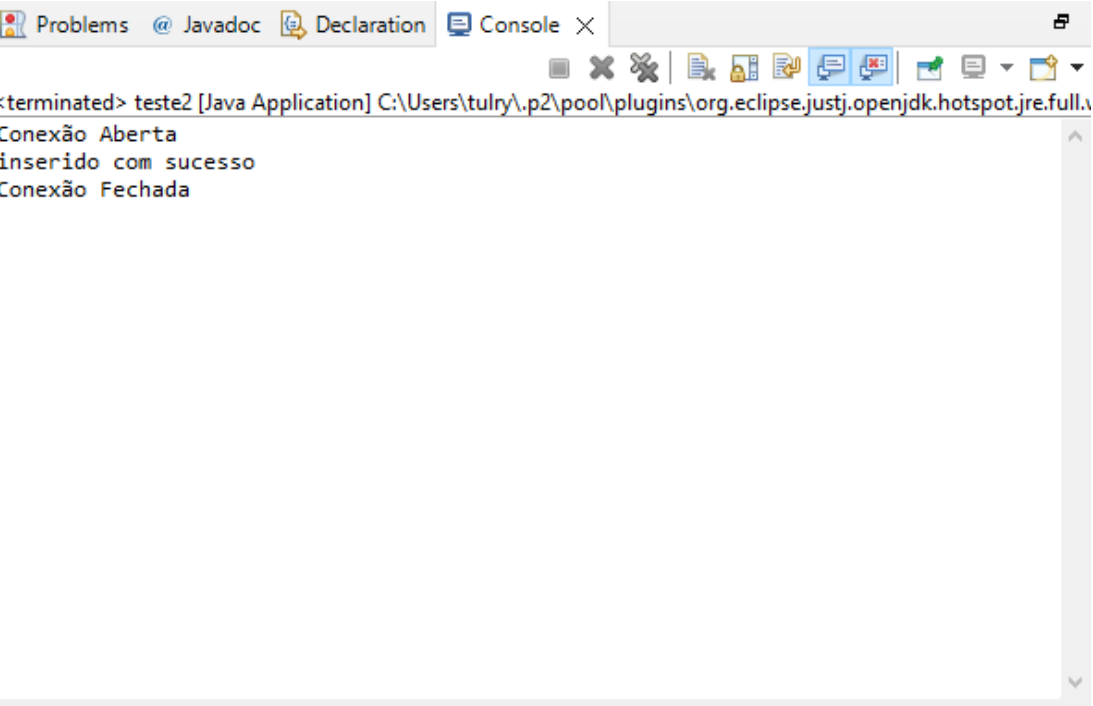
public class teste2 {

    public static void main(String[] args) {
        Connection con = Conexao.abrirConexao();
        Bike2 bk = new Bike2();
        BikeDAO bd = new BikeDAO(con);

        //testando metodo inserir

        bk.setMarca("oggi");
        bk.setCor("preta");
        bk.setDescricao("Bike Cliente 1");
        System.out.println(bd.inserir(bk));

        Conexao.fecharConexao(con);
    }
}
```



<terminated> teste2 [Java Application] C:\Users\tulry\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.\n
Conexão Aberta
inserido com sucesso
Conexão Fechada