

INF335 – Ambientes para Concepção de Software

Formação de Especialista em Engenharia de Software

INSTITUTO DE COMPUTAÇÃO – UNICAMP

1º SEMESTRE DE 2024

Prof. Rodrigo Bonacin rbonacin@unicamp.br

13/Julho/2024 TRABALHO 6

1 Objetivos

Este trabalho tem como objetivo desenvolver habilidades de integração entre o gerenciador de banco de dados MongoDB (e MySQL) com o Java. Este trabalho compõe a avaliação da disciplina INF335, e poderá ser realizado individualmente ou em dupla.

2 Atividades

Os seguintes passos devem ser realizados:

1. Considerar os métodos desenvolvidos para o MySQL nos slides da aula 6 números 43 até 52, onde foram desenvolvidos os seguintes métodos:
 - a. *conectar* - Retorna uma conexão com o banco de dados.
 - b. *listaProdutos* – Imprime uma lista de todos os produtos
 - c. *insereproduto* – Insere um novo produto
 - d. *alteraValorProduto* – Altera valor de um produto
 - e. *apagaProduto* – Apaga um produto
 - f. *main* – No qual: (i) imprime a lista original de produto, (ii) insere um novo produto, (iii) lista com o novo produto, (iv) altera o valor do produto inserido, (v) lista com valor do produto alterado, (vi) apaga o produto que foi inserido, (vii) lista novamente de modo que seja igual a lista original e (viii) fecha conexão.
2. Crie coleção em base do MongoDB com os produtos listados nos slides.
3. Desenvolver código equivalente (com todas as funcionalidades) ao código do item (1), entretanto integrada com a base de dados do MongoDB (item 2).

3 Entrega

Os seguintes itens devem ser entregues:

1. Código fonte *.java* com as funcionalidades descritas acima.
2. Relatório com *prints* dos comandos usados para criar e popular coleção no MongoDB, bem como *prints* da execução do programa em java.

Deverá ser entregue um arquivo por dupla, constando os respectivos nomes e números de matrículas. As entregas deverão ser realizadas pelo Moodle.

DATA FINAL DE ENTREGA: 19/07/2024

A coleção de dados apresentada nos slides já está inserida no banco de dados Mongo DB.

```
osboxes@osboxes: ~  
osboxes@osboxes:~$ systemctl status mongod  
● mongod.service - MongoDB Database Server  
   Loaded: loaded (/lib/systemd/system/mongod.service; enabled; vendor preset: enabled)  
   Active: active (running) since Mon 2024-07-15 09:12:03 EDT; 2h 31min ago  
     Docs: https://docs.mongodb.org/manual  
  Main PID: 1071 (mongod)  
    Memory: 32.8M  
       CPU: 52.236s  
    CGroup: /system.slice/mongod.service  
            └─1071 /usr/bin/mongod --config /etc/mongod.conf  
  
Jul 15 09:12:03 osboxes systemd[1]: Started MongoDB Database Server.  
lines 1-11/11 (END)
```

```
osboxes@osboxes: ~  
metrics about your deployment (disk utilization, CPU, operation statistics, etc).  
  
The monitoring data will be available on a MongoDB website with a unique URL accessible to you  
and anyone you share the URL with. MongoDB may use this information to make product  
improvements and to suggest MongoDB products and deployment options to you.  
  
To enable free monitoring, run the following command: db.enableFreeMonitoring()  
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()  
---  
> db.produtos.find()  
{ "_id" : ObjectId("62587e50b80bedf38dbf596e"), "nome" : "Calular Galaxy S10", "descricao" : "128 Gb, Preto, com carregador", "valor"  
: "1250.0", "estado" : "Poucos riscos, estado de novo" }  
{ "_id" : ObjectId("62587e50b80bedf38dbf596f"), "nome" : "Prod2", "descricao" : "Bla bla", "valor" : "1200", "estado" : "Bla bla" }  
{ "_id" : ObjectId("62587e50b80bedf38dbf5970"), "nome" : "Prod3", "descricao" : "Bla bla", "valor" : "120.0", "estado" : "Bla bla" }  
{ "_id" : ObjectId("62587e50b80bedf38dbf5971"), "nome" : "Prod4", "descricao" : "Bla bla", "valor" : "1300.0", "estado" : "Bla bla" }  
{ "_id" : ObjectId("62587e50b80bedf38dbf5972"), "nome" : "Prod5", "descricao" : "Bla bla", "valor" : "9400.0", "estado" : "Bla bla" }  
{ "_id" : ObjectId("62587e50b80bedf38dbf5973"), "nome" : "Prod6", "descricao" : "Bla bla", "valor" : "1500.0", "estado" : "Bla bla" }
```

1.a concetar - Retorna uma conexão com o banco de dados

Aula 6 - mongoProject/pom.xml - Eclipse IDE

File Edit Navigate Search Project Run Window Help

Package Explorer x

- mongoProject
 - src/main/java
 - mongoProject
 - MongoAppMain.java
 - MongoDBConnection.java
 - src/main/resources
 - src/test/java
 - src/test/resources
 - JRE System Library [J2SE-1.5]
 - Maven Dependencies
 - bson-4.5.1.jar - /home/osboxes/.m2/
 - mongodb-driver-core-4.5.1.jar - /home/osboxes/.m2/
 - mongodb-driver-sync-4.5.1.jar - /home/osboxes/.m2/
 - src
 - target
 - pom.xml

mongoProject/pom.xml x | MongoAppMain.java | MongoDBConnection.java

```
<?xml version="1.0" encoding="UTF-8" ?>  
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">  
  <modelVersion>4.0.0</modelVersion>  
  <groupId>br.unicamp.ic.inf335.app</groupId>  
  <artifactId>mongoProject</artifactId>  
  <version>0.0.1-SNAPSHOT</version>  
  <name>br.unicamp.ic.inf335.app.mongoProject</name>  
  <dependencies>  
    <dependency>  
      <groupId>org.mongodb</groupId>  
      <artifactId>mongodb-driver-sync</artifactId>  
      <version>4.5.1</version>  
    </dependency>  
  </dependencies>  
</project>
```

Overview Dependencies Dependency Hierarchy Effective POM pom.xml

Problems Javadoc Declaration Console x

<terminated> MongoAppMain [Java Application] /home/osboxes/.p2/pool/plugins/org.eclipse.justi.openjdk.hotspot.jre.full.linux.x86_64_17.0.2

Aula 6 - mongoProject/src/main/java/mongoProject/MongoDBConnection.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer

- mongoProject
 - src/main/java
 - mongoProject
 - MongoAppMain.java
 - MongoDBConnection.java
 - src/main/resources
 - src/test/java
 - src/test/resources
 - JRE System Library [J2SE-1.5]
 - Maven Dependencies
 - bson-4.5.1.jar - /home/osboxes/.m2/
 - mongodb-driver-core-4.5.1.jar - /home/osboxes/.m2/
 - mongodb-driver-sync-4.5.1.jar - /home/osboxes/.m2/
 - src
 - target
 - pom.xml

mongoProject/pom.xml

MongoAppMain.java

MongoDBConnection.java

```

1 package mongoProject;
2
3 import com.mongodb.client.MongoClient;
4 import com.mongodb.client.MongoClients;
5 import com.mongodb.client.MongoCollection;
6 import com.mongodb.client.MongoDatabase;
7 import com.mongodb.client.result.DeleteResult;
8 import com.mongodb.client.result.UpdateResult;
9
10 public class MongoDBConnection {
11
12     private static final String CONNECTION_STRING = "mongodb://localhost:27017"; // URL de conexão do MongoDB
13     private static final String DATABASE_NAME = "test"; // Substitua pelo nome do seu banco de dados
14
15     private MongoClient mongoClient;
16     private MongoDatabase database;
17
18     // Método para conectar ao MongoDB
19     public void conectar() {
20         try {
21             mongoClient = MongoClients.create(CONNECTION_STRING);
22             database = mongoClient.getDatabase(DATABASE_NAME);
23             System.out.println("Conectado ao banco de dados " + DATABASE_NAME);
24         } catch (Exception e) {
25             System.err.println("Erro ao conectar ao MongoDB: " + e.getMessage());
26             e.printStackTrace();
27         }
28     }
29 }

```

Problems Javadoc Declaration Console

<terminated> MongoAppMain [Java Application] /home/osboxes/.p2/pool/plugins/org.eclipse.justi.openjdk.hotspot.jre.full.linux.x86_64_17.0.2
 Jul 15, 2024 11:20:58 AM com.mongodb.diagnostics.logging.Loggers shouldUseSLF4J
 WARNING: SLF4J not found on the classpath. Logging is disabled for the 'org.mongodb.driver' component

Conectado ao banco de dados test

1.b listaProdutos – Imprime uma lista de todos os produtos

Aula 6 - mongoProject/src/main/java/mongoProject/MongoDBConnection.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer

- mongoProject
 - src/main/java
 - mongoProject
 - MongoAppMain.java
 - MongoDBConnection.java
 - src/main/resources
 - src/test/java
 - src/test/resources
 - JRE System Library [J2SE-1.5]
 - Maven Dependencies
 - bson-4.5.1.jar - /home/osboxes/.m2/
 - mongodb-driver-core-4.5.1.jar - /home/osboxes/.m2/
 - mongodb-driver-sync-4.5.1.jar - /home/osboxes/.m2/
 - src
 - target
 - pom.xml

mongoProject/pom.xml

MongoAppMain.java

MongoDBConnection.java

```

32 // Método para listar
33 public void listar() {
34     if (database == null) {
35         System.err.println("Banco de dados não está conectado. Conecte-se primeiro.");
36         return;
37     }
38     try {
39         MongoCollection<org.bson.Document> collection = database.getCollection("produtos");
40
41         System.out.println("Imprimindo Produtos");
42         Iterable<org.bson.Document> produtos = collection.find();
43
44         for (org.bson.Document produto : produtos) {
45             String nome = ((org.bson.Document) produto).getString("nome");
46             String descricao = produto.getString("descricao");
47             String valor = produto.getString("valor");
48             String estado = produto.getString("estado");
49             System.out.println(nome + " -- " + descricao + " -- " + valor + " -- " + estado);
50         }
51     } catch (Exception e) {
52         System.err.println("Erro ao listar coleções: " + e.getMessage());
53         e.printStackTrace();
54     }
55 }

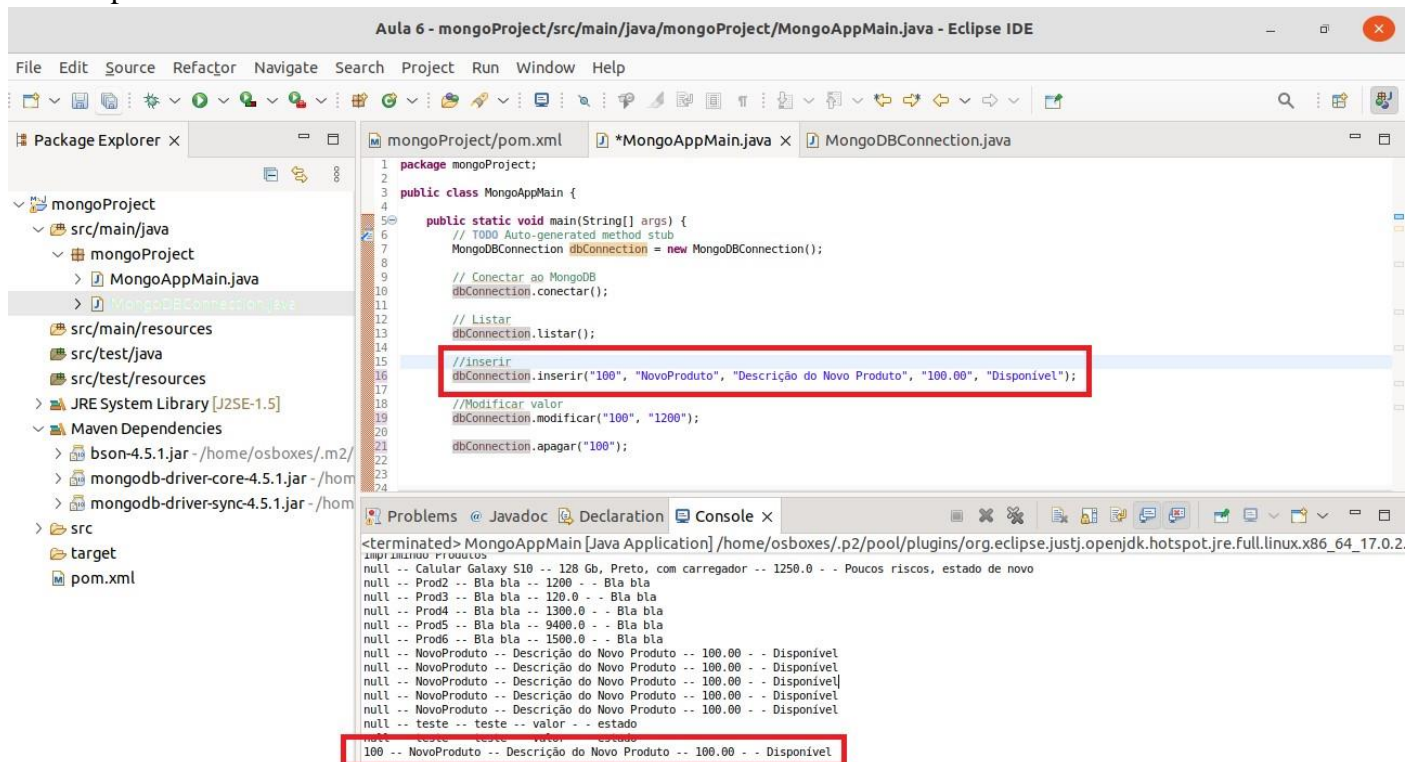
```

Problems Javadoc Declaration Console

<terminated> MongoAppMain [Java Application] /home/osboxes/.p2/pool/plugins/org.eclipse.justi.openjdk.hotspot.jre.full.linux.x86_64_17.0.2
 Conectado ao banco de dados test
 Imprimindo Produtos
 Calcular Galaxy S10 -- 128 Gb, Preto, com carregador -- 1250.0 -- Poucos riscos, estado de novo
 Prod2 -- Bla bla -- 1200.0 -- Bla bla
 Prod3 -- Bla bla -- 120.0 -- Bla bla
 Prod4 -- Bla bla -- 1300.0 -- Bla bla
 Prod5 -- Bla bla -- 9400.0 -- Bla bla
 Prod6 -- Bla bla -- 1500.0 -- Bla bla
 NovoProduto -- Descrição do Novo Produto -- 100.00 -- Disponível
 NovoProduto -- Descrição do Novo Produto -- 100.00 -- Disponível
 NovoProduto -- Descrição do Novo Produto -- 100.00 -- Disponível
 NovoProduto -- Descrição do Novo Produto -- 100.00 -- Disponível
 NovoProduto -- Descrição do Novo Produto -- 100.00 -- Disponível

1.c insereproduto – Insere um novo produto

Inserido produto com id 100.



```
package mongoProject;

public class MongoAppMain {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        MongoDBConnection dbConnection = new MongoDBConnection();

        // Conectar ao MongoDB
        dbConnection.conectar();

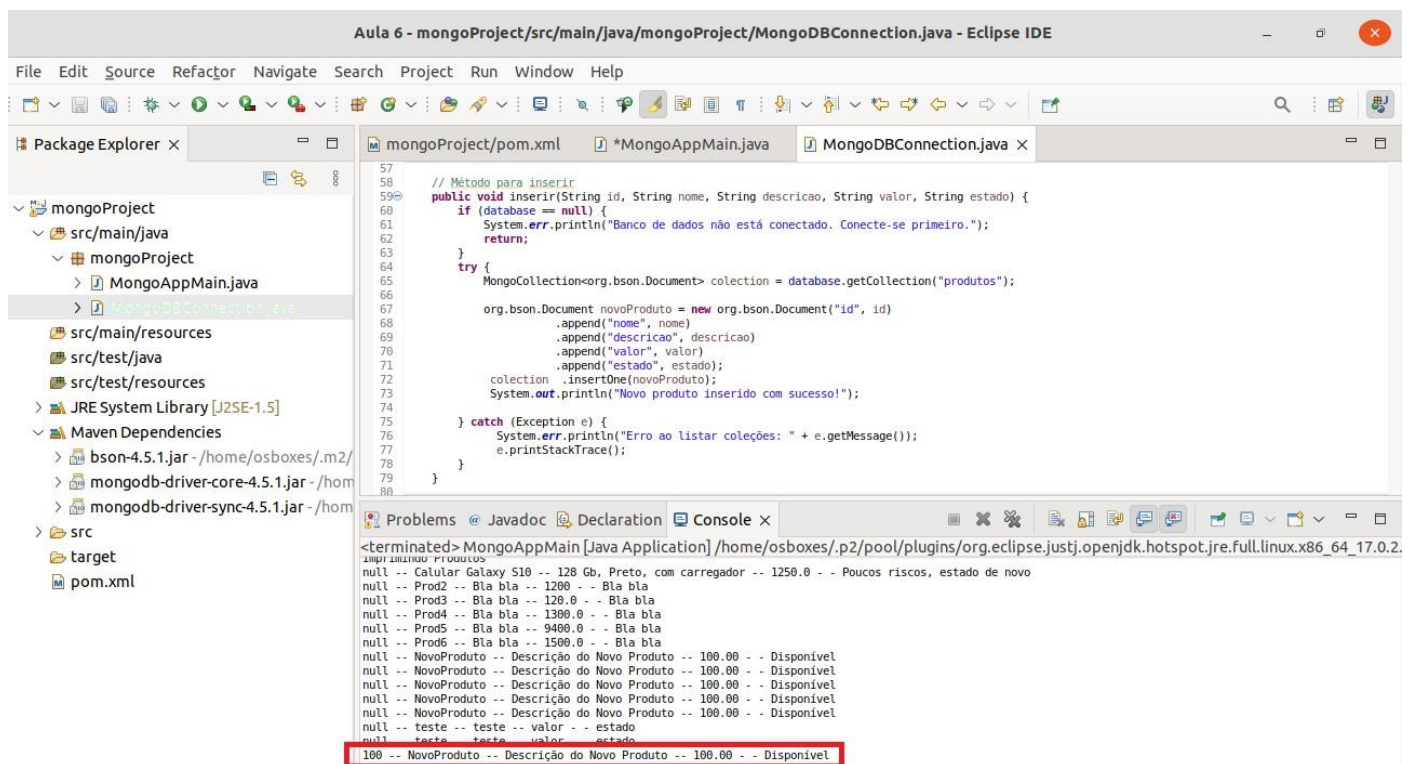
        // Listar
        dbConnection.listar();

        //Inserir
        dbConnection.inserir("100", "NovoProduto", "Descrição do Novo Produto", "100.00", "Disponível");

        //Modificar valor
        dbConnection.modificar("100", "1200");

        dbConnection.apagar("100");
    }
}
```

```
<terminated> MongoAppMain [Java Application] /home/osboxes/.p2/pool/plugins/org.eclipse.justi.openjdk.hotspot.jre.full.linux.x86_64_17.0.2
null -- Calular Galaxy S10 -- 128 Gb, Preto, com carregador -- 1250.0 -- Poucos riscos, estado de novo
null -- Prod2 -- Bla bla -- 1200 -- Bla bla
null -- Prod3 -- Bla bla -- 120.0 -- Bla bla
null -- Prod4 -- Bla bla -- 1300.0 -- Bla bla
null -- Prod5 -- Bla bla -- 9400.0 -- Bla bla
null -- Prod6 -- Bla bla -- 1500.0 -- Bla bla
null -- NovoProduto -- Descrição do Novo Produto -- 100.00 -- Disponível
null -- NovoProduto -- Descrição do Novo Produto -- 100.00 -- Disponível
null -- NovoProduto -- Descrição do Novo Produto -- 100.00 -- Disponível
null -- NovoProduto -- Descrição do Novo Produto -- 100.00 -- Disponível
null -- teste -- teste -- valor -- estado
100 -- NovoProduto -- Descrição do Novo Produto -- 100.00 -- Disponível
```



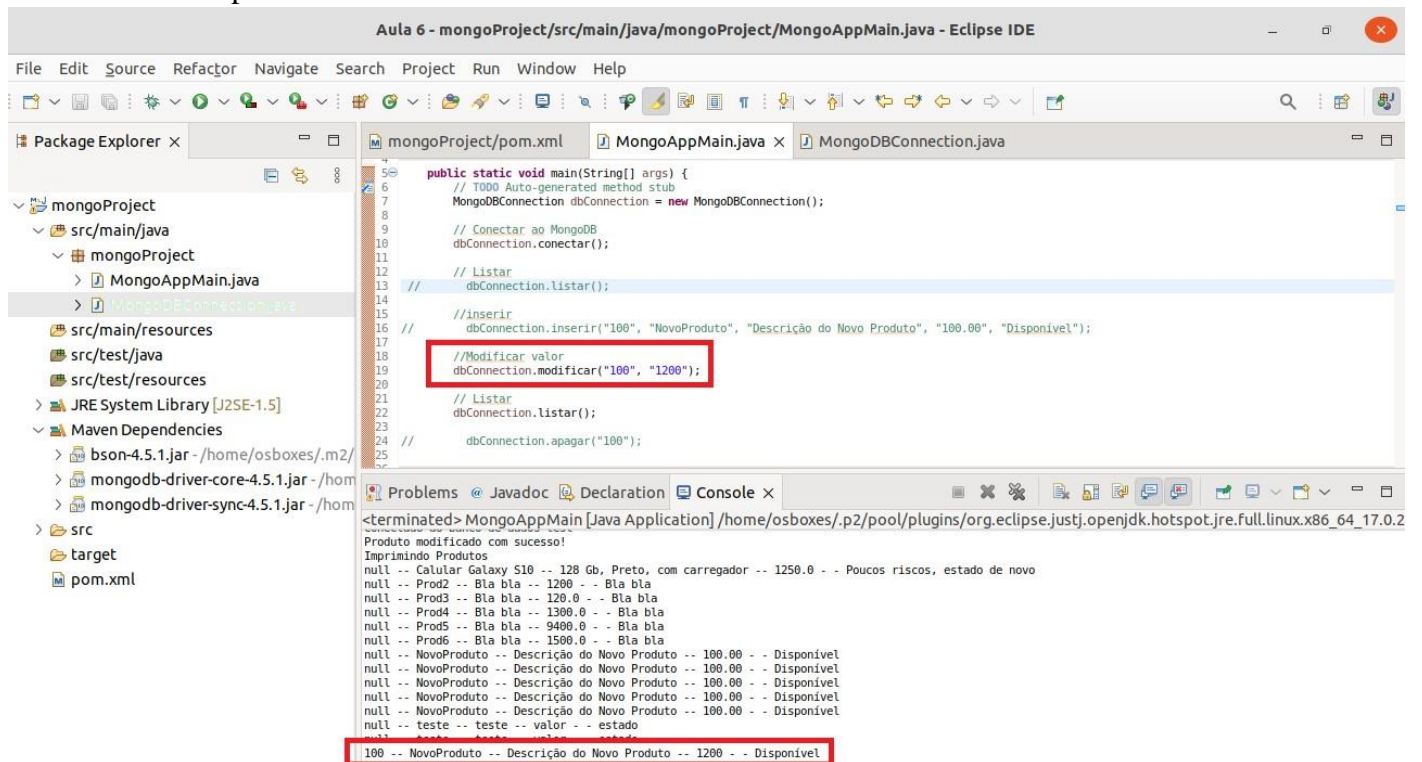
```
// Método para inserir
public void inserir(String id, String nome, String descricao, String valor, String estado) {
    if (database == null) {
        System.err.println("Banco de dados não está conectado. Conecte-se primeiro.");
        return;
    }
    try {
        MongoCollection<org.bson.Document> collection = database.getCollection("produtos");

        org.bson.Document novoProduto = new org.bson.Document("id", id)
            .append("nome", nome)
            .append("descricao", descricao)
            .append("valor", valor)
            .append("estado", estado);
        collection.insertOne(novoProduto);
        System.out.println("Novo produto inserido com sucesso!");
    } catch (Exception e) {
        System.err.println("Erro ao listar coleções: " + e.getMessage());
        e.printStackTrace();
    }
}
```

```
<terminated> MongoAppMain [Java Application] /home/osboxes/.p2/pool/plugins/org.eclipse.justi.openjdk.hotspot.jre.full.linux.x86_64_17.0.2
null -- Calular Galaxy S10 -- 128 Gb, Preto, com carregador -- 1250.0 -- Poucos riscos, estado de novo
null -- Prod2 -- Bla bla -- 1200 -- Bla bla
null -- Prod3 -- Bla bla -- 120.0 -- Bla bla
null -- Prod4 -- Bla bla -- 1300.0 -- Bla bla
null -- Prod5 -- Bla bla -- 9400.0 -- Bla bla
null -- Prod6 -- Bla bla -- 1500.0 -- Bla bla
null -- NovoProduto -- Descrição do Novo Produto -- 100.00 -- Disponível
null -- NovoProduto -- Descrição do Novo Produto -- 100.00 -- Disponível
null -- NovoProduto -- Descrição do Novo Produto -- 100.00 -- Disponível
null -- NovoProduto -- Descrição do Novo Produto -- 100.00 -- Disponível
null -- teste -- teste -- valor -- estado
100 -- NovoProduto -- Descrição do Novo Produto -- 100.00 -- Disponível
```

1.d alteraValorProduto – Altera valor de um produto

Alterado valor do produto de id 100.



```
public static void main(String[] args) {
    // TODO Auto-generated method stub
    MongoDBConnection dbConnection = new MongoDBConnection();

    // Conectar ao MongoDB
    dbConnection.conectar();

    // Listar
    dbConnection.listar();

    // Inserir
    dbConnection.inserir("100", "NovoProduto", "Descrição do Novo Produto", "100.00", "Disponível");

    // Modificar valor
    dbConnection.modificar("100", "1200");

    // Listar
    dbConnection.listar();

    // Apagar
    dbConnection.apagar("100");
}
```

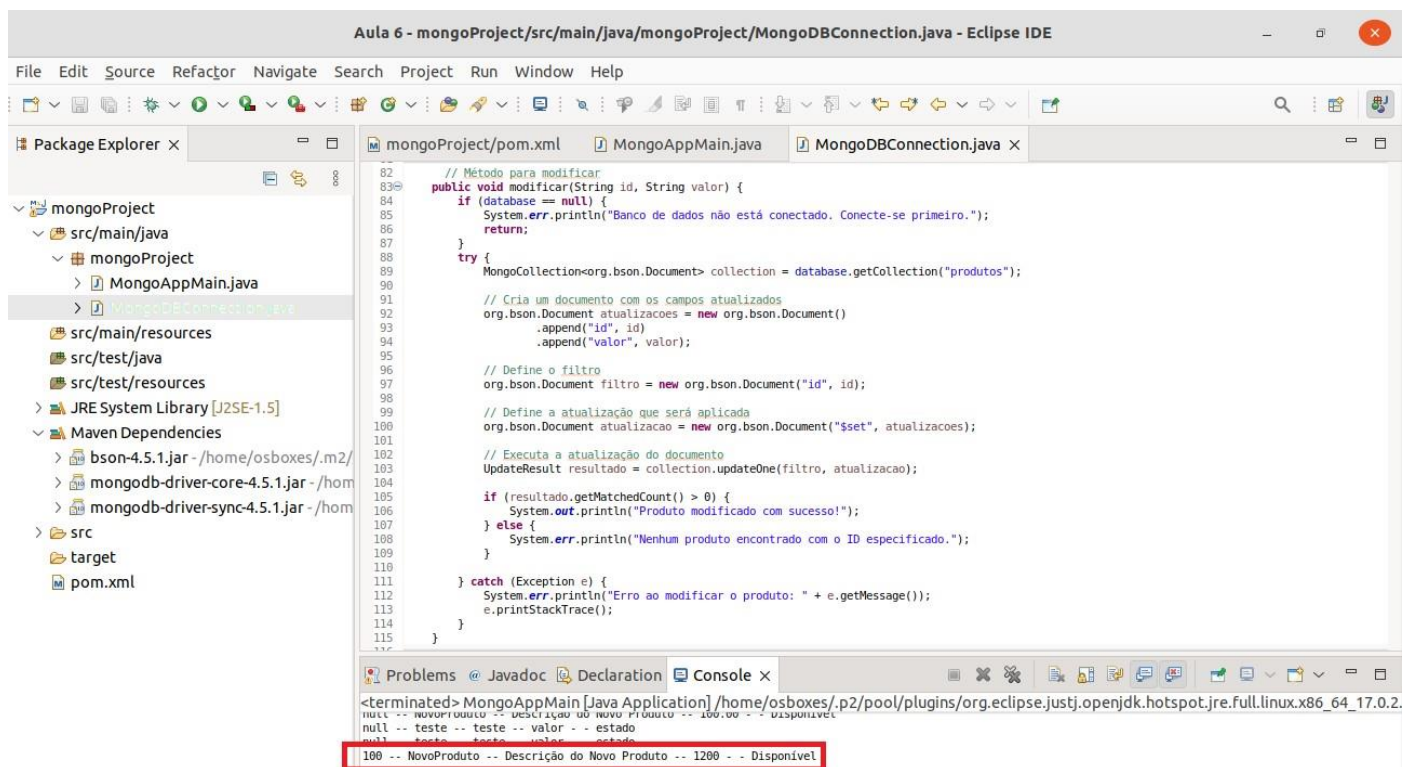
Problems @ Javadoc Declaration Console X

<terminated> MongoAppMain [Java Application] /home/osboxes/.p2/pool/plugins/org.eclipse.justi.openjdk.hotspot.jre.full.linux.x86_64_17.0.2

Produto modificado com sucesso!

Imprimindo Produtos

```
null -- Calular Galaxy S10 -- 128 Gb, Preto, com carregador -- 1250.0 -- Poucos riscos, estado de novo
null -- Prod2 -- Bla bla -- 1200 -- Bla bla
null -- Prod3 -- Bla bla -- 120.0 -- Bla bla
null -- Prod4 -- Bla bla -- 1300.0 -- Bla bla
null -- Prod5 -- Bla bla -- 9400.0 -- Bla bla
null -- Prod6 -- Bla bla -- 1500.0 -- Bla bla
null -- NovoProduto -- Descrição do Novo Produto -- 100.00 -- Disponível
null -- NovoProduto -- Descrição do Novo Produto -- 100.00 -- Disponível
null -- NovoProduto -- Descrição do Novo Produto -- 100.00 -- Disponível
null -- NovoProduto -- Descrição do Novo Produto -- 100.00 -- Disponível
null -- NovoProduto -- Descrição do Novo Produto -- 100.00 -- Disponível
null -- teste -- teste -- valor -- estado
100 -- NovoProduto -- Descrição do Novo Produto -- 1200 -- Disponível
```



```
// Método para modificar
public void modificar(String id, String valor) {
    if (database == null) {
        System.err.println("Banco de dados não está conectado. Conecte-se primeiro.");
        return;
    }
    try {
        MongoClient mongoClient = org.mongodb.driver.MongoClient.builder().addCredential(database).build();
        MongoCollection<org.bson.Document> collection = database.getCollection("produtos");

        // Cria um documento com os campos atualizados
        org.bson.Document atualizacoes = new org.bson.Document()
            .append("id", id)
            .append("valor", valor);

        // Define o filtro
        org.bson.Document filtro = new org.bson.Document("id", id);

        // Define a atualização que será aplicada
        org.bson.Document atualizacao = new org.bson.Document("$set", atualizacoes);

        // Executa a atualização do documento
        UpdateResult resultado = collection.updateOne(filtro, atualizacao);

        if (resultado.getMatchedCount() > 0) {
            System.out.println("Produto modificado com sucesso!");
        } else {
            System.err.println("Nenhum produto encontrado com o ID especificado.");
        }
    } catch (Exception e) {
        System.err.println("Erro ao modificar o produto: " + e.getMessage());
        e.printStackTrace();
    }
}
```

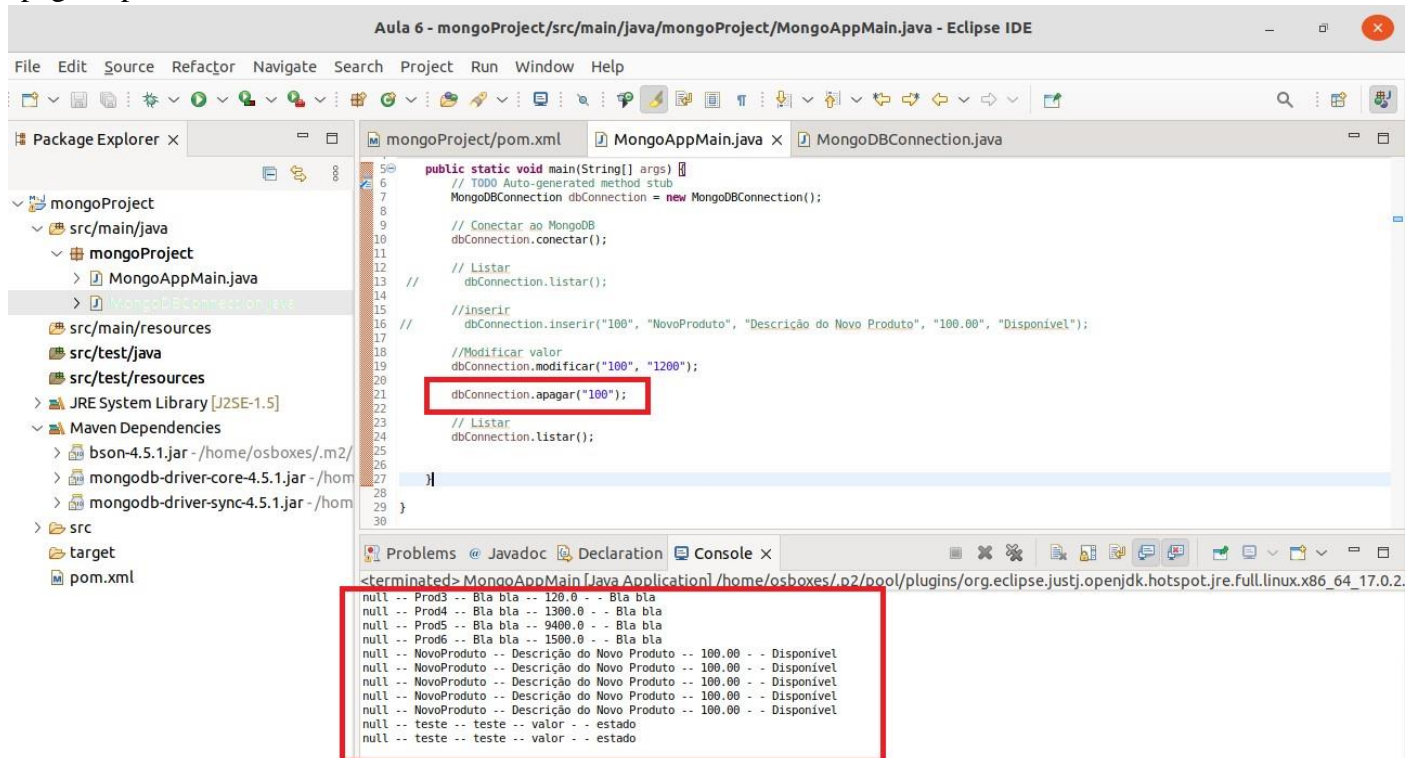
Problems @ Javadoc Declaration Console X

<terminated> MongoAppMain [Java Application] /home/osboxes/.p2/pool/plugins/org.eclipse.justi.openjdk.hotspot.jre.full.linux.x86_64_17.0.2

```
null -- NovoProduto -- Descrição do Novo Produto -- 100.00 -- Disponível
null -- teste -- teste -- valor -- estado
100 -- NovoProduto -- Descrição do Novo Produto -- 1200 -- Disponível
```

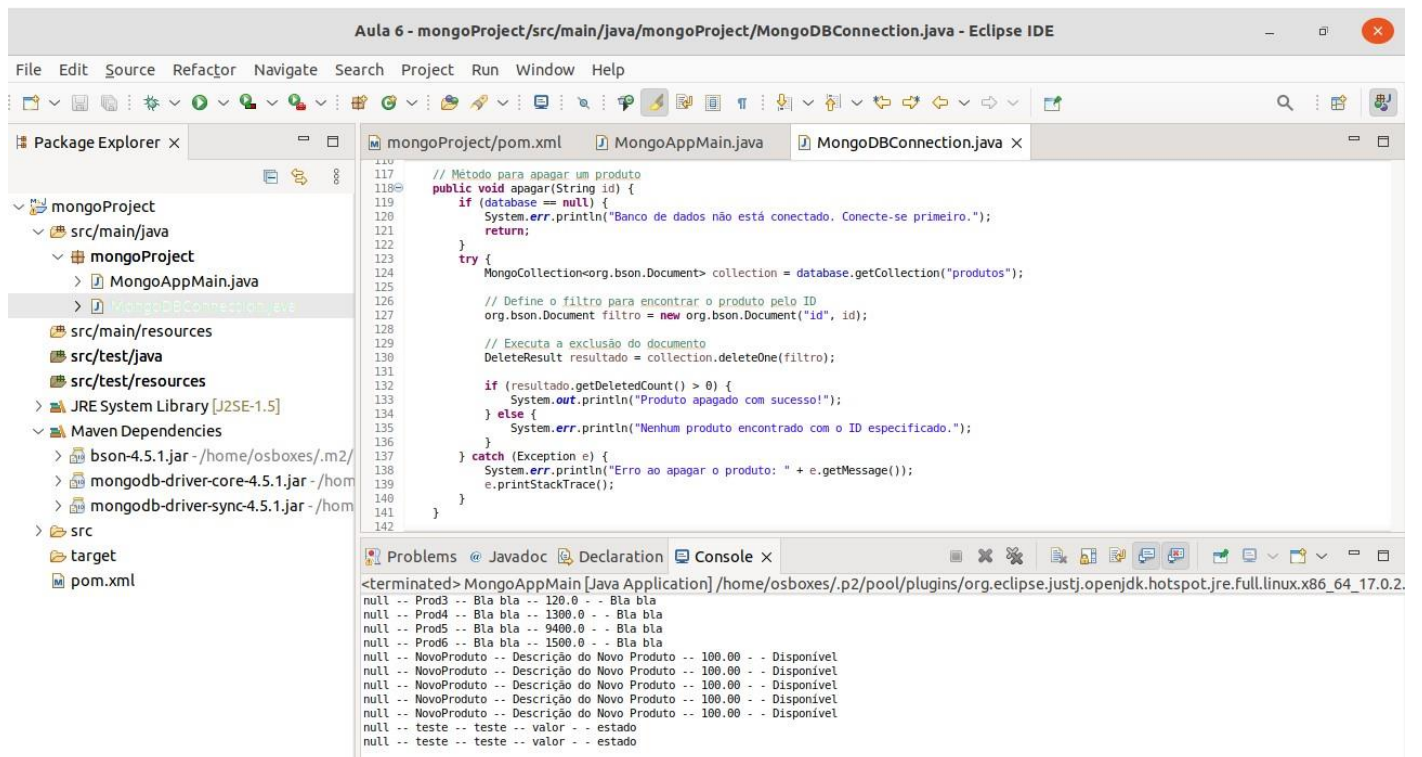

1.e apagaProduto – Apagar um produto

Apagado produto de id 100.



```
public static void main(String[] args) {  
    // TODO Auto-generated method stub  
    MongoDBConnection dbConnection = new MongoDBConnection();  
  
    // Conectar ao MongoDB  
    dbConnection.conectar();  
  
    // Listar  
    dbConnection.listar();  
  
    // Inserir  
    dbConnection.inserir("100", "NovoProduto", "Descrição do Novo Produto", "100.00", "Disponível");  
  
    // Modificar valor  
    dbConnection.modificar("100", "1200");  
  
    dbConnection.apagar("100");  
  
    // Listar  
    dbConnection.listar();  
}
```

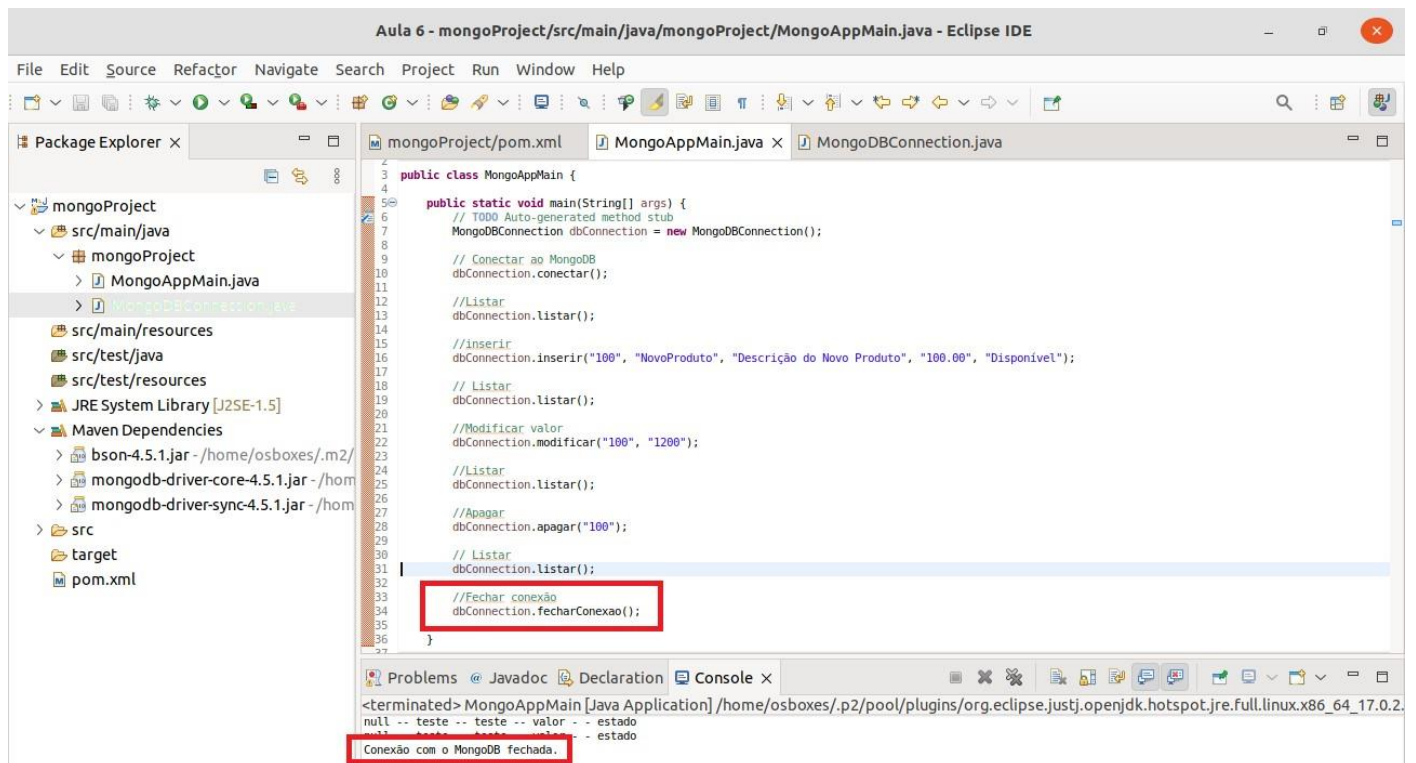
```
<terminated> MongoAppMain [Java Application] /home/osboxes/.p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.linux.x86_64_17.0.2.  
null -- Prod3 -- Bla bla -- 120.0 -- Bla bla  
null -- Prod4 -- Bla bla -- 1300.0 -- Bla bla  
null -- Prod5 -- Bla bla -- 9400.0 -- Bla bla  
null -- Prod6 -- Bla bla -- 1500.0 -- Bla bla  
null -- NovoProduto -- Descrição do Novo Produto -- 100.00 -- Disponível  
null -- NovoProduto -- Descrição do Novo Produto -- 100.00 -- Disponível  
null -- NovoProduto -- Descrição do Novo Produto -- 100.00 -- Disponível  
null -- NovoProduto -- Descrição do Novo Produto -- 100.00 -- Disponível  
null -- NovoProduto -- Descrição do Novo Produto -- 100.00 -- Disponível  
null -- teste -- teste -- valor -- estado  
null -- teste -- teste -- valor -- estado
```



```
// Método para apagar um produto  
public void apagar(String id) {  
    if (database == null) {  
        System.err.println("Banco de dados não está conectado. Conecte-se primeiro.");  
        return;  
    }  
    try {  
        MongoCollection<org.bson.Document> collection = database.getCollection("produtos");  
  
        // Define o filtro para encontrar o produto pelo ID  
        org.bson.Document filtro = new org.bson.Document("id", id);  
  
        // Executa a exclusão do documento  
        DeleteResult resultado = collection.deleteOne(filtro);  
  
        if (resultado.getDeletedCount() > 0) {  
            System.out.println("Produto apagado com sucesso!");  
        } else {  
            System.err.println("Nenhum produto encontrado com o ID especificado.");  
        }  
    } catch (Exception e) {  
        System.err.println("Erro ao apagar o produto: " + e.getMessage());  
        e.printStackTrace();  
    }  
}
```

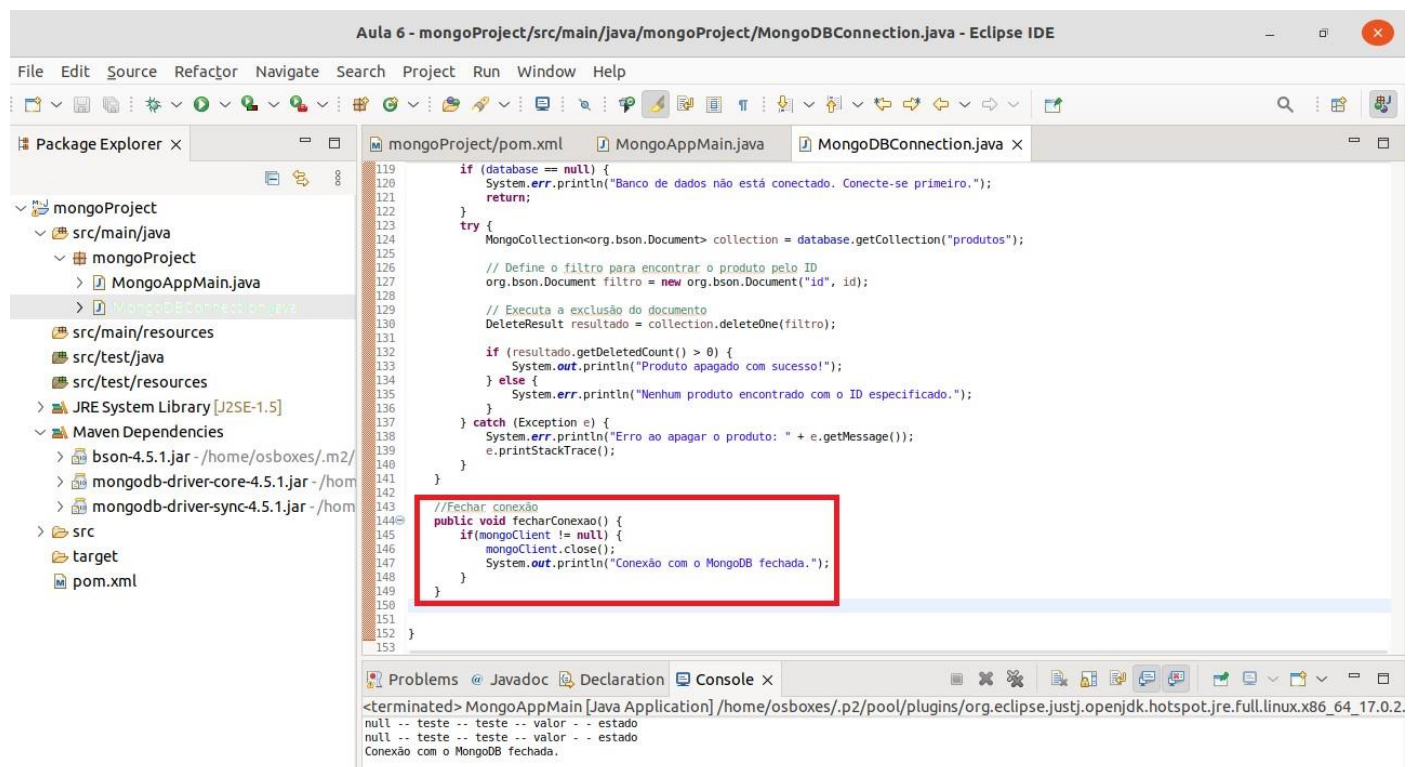
```
<terminated> MongoAppMain [Java Application] /home/osboxes/.p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.linux.x86_64_17.0.2.  
null -- Prod3 -- Bla bla -- 120.0 -- Bla bla  
null -- Prod4 -- Bla bla -- 1300.0 -- Bla bla  
null -- Prod5 -- Bla bla -- 9400.0 -- Bla bla  
null -- Prod6 -- Bla bla -- 1500.0 -- Bla bla  
null -- NovoProduto -- Descrição do Novo Produto -- 100.00 -- Disponível  
null -- NovoProduto -- Descrição do Novo Produto -- 100.00 -- Disponível  
null -- NovoProduto -- Descrição do Novo Produto -- 100.00 -- Disponível  
null -- NovoProduto -- Descrição do Novo Produto -- 100.00 -- Disponível  
null -- NovoProduto -- Descrição do Novo Produto -- 100.00 -- Disponível  
null -- teste -- teste -- valor -- estado  
null -- teste -- teste -- valor -- estado
```

1.f main – No qual: (i) imprime a lista original de produto, (ii) insere um novo produto, (iii) lista com o novo produto, (iv) altera o valor do produto inserido, (v) lista com valor do produto alterado, (vi) apaga o produto que foi inserido, (vii) lista novamente de modo que seja igual a lista original e (viii) fecha conexão



```
public class MongoAppMain {  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        MongoDBConnection dbConnection = new MongoDBConnection();  
        dbConnection.conectar();  
        //Listar  
        dbConnection.listar();  
        //Inserir  
        dbConnection.inserir("100", "NovoProduto", "Descrição do Novo Produto", "100.00", "Disponível");  
        // Listar  
        dbConnection.listar();  
        //Modificar valor  
        dbConnection.modificar("100", "1200");  
        //Listar  
        dbConnection.listar();  
        //Apagar  
        dbConnection.apagar("100");  
        // Listar  
        dbConnection.listar();  
        //Fechar conexão  
        dbConnection.fecharConexao();  
    }  
}
```

Conexão com o MongoDB fechada.



```
if (database == null) {  
    System.err.println("Banco de dados não está conectado. Conecte-se primeiro.");  
    return;  
}  
try {  
    MongoClient mongoClient = new MongoClient(database.getServerAddress());  
    MongoCollection<org.bson.Document> collection = database.getCollection("produtos");  
    // Define o filtro para encontrar o produto pelo ID  
    org.bson.Document filtro = new org.bson.Document("id", id);  
    // Executa a exclusão do documento  
    DeleteResult resultado = collection.deleteOne(filtro);  
    if (resultado.getDeletedCount() > 0) {  
        System.out.println("Produto apagado com sucesso!");  
    } else {  
        System.err.println("Nenhum produto encontrado com o ID especificado.");  
    }  
} catch (Exception e) {  
    System.err.println("Erro ao apagar o produto: " + e.getMessage());  
    e.printStackTrace();  
}  
}  
//Fechar conexão  
public void fecharConexao() {  
    if(mongoClient != null) {  
        mongoClient.close();  
        System.out.println("Conexão com o MongoDB fechada.");  
    }  
}
```

Conexão com o MongoDB fechada.