Exercício - Desenvolvimento de Web Service Controle de Estoque

Evandro César Freiberger

Instituto Federal de Educação, Ciência e Tecnologia

evandro.freiberger@cba.ifmt.edu.br

17 de abril de 2023

Descrição

Produzir uma solução orientada a serviço para o controle de estoque de produtos. Deverá ser produzido um projeto para o serviço Web e um projeto para o cliente consumidor do serviço.
O serviço deverá fornecer as seguintes capacidades:

- Adicionar produto.
- Remover produto.
- Adicionar estoque a um produto.
- Remover estoque de um produto.
- Localizar produto por código.
- Listar produtos.
- Totalizar produtos cadastrados.
- Totalizar estoque físico.

Preparando Projeto Java/Maven do Lado Servidor

Criar um projeto Java/Maven com as seguintes definições:

groupId: ifmt.cba

artifactId: controle-estoque

version: 1.0-SNAPSHOT

Alterando o arquivo POM I

```xml
<?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>ifmt.cba</groupId>
  <artifactId>controle-estoque</artifactId>
  <version>1.0-SNAPSHOT</version>

  <name>controle-estoque</name>

  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <maven.compiler.source>17</maven.compiler.source>
    <maven.compiler.target>17</maven.compiler.target>
  </properties>

  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.11</version>
      <scope>test</scope>
    </dependency>

    <!-- https://mvnrepository.com/artifact/jakarta.xml.ws/jakarta.xml.ws-api -->
```

Alterando o arquivo POM II

```
29      <dependency>
30        <groupId>jakarta.xml.ws</groupId>
31        <artifactId>jakarta.xml.ws-api</artifactId>
32        <version>4.0.0</version>
33      </dependency>
34
35      <!-- https://mvnrepository.com/artifact/com.sun.xml.ws/jaxws-rt -->
36      <dependency>
37        <groupId>com.sun.xml.ws</groupId>
38        <artifactId>jaxws-rt</artifactId>
39        <version>4.0.1</version>
40      </dependency>
41
42      <!-- https://mvnrepository.com/artifact/com.sun.xml.ws/jaxws-ri -->
43      <dependency>
44        <groupId>com.sun.xml.ws</groupId>
45        <artifactId>jaxws-ri</artifactId>
46        <version>4.0.1</version>
47        <type>pom</type>
48      </dependency>
49
50      <!-- https://mvnrepository.com/artifact/jakarta.xml.bind/jakarta.xml.bind-api
            -->
51      <dependency>
52        <groupId>jakarta.xml.bind</groupId>
53        <artifactId>jakarta.xml.bind-api</artifactId>
54        <version>4.0.0</version>
55      </dependency>
56
```

Alterando o arquivo POM III

```
57      <!-- https://mvnrepository.com/artifact/com.sun.xml.bind/jaxb-impl -->
58      <dependency>
59        <groupId>com.sun.xml.bind</groupId>
60        <artifactId>jaxb-impl</artifactId>
61        <version>4.0.2</version>
62      </dependency>
63    </dependencies>
64    <build>
65      <plugins>
66        <plugin>
67          <groupId>org.apache.maven.plugins</groupId>
68          <artifactId>maven-compiler-plugin</artifactId>
69          <version>3.11.0</version>
70          <configuration>
71            <release>17</release>
72          </configuration>
73          <dependencies>
74            <!-- https://mvnrepository.com/artifact/org.ow2.asm/asm -->
75            <dependency>
76              <groupId>org.ow2.asm</groupId>
77              <artifactId>asm</artifactId>
78              <version>9.5</version>
79            </dependency>
80          </dependencies>
81        </plugin>
82        <plugin>
83          <groupId>com.sun.xml.ws</groupId>
84          <artifactId>jaxws-maven-plugin</artifactId>
85          <version>4.0.1</version>
```

Alterando o arquivo POM IV

```
86          <executions>
87            <execution>
88              <?m2e execute onConfiguration,onIncremental?>
89              <goals>
90                <goal>wsgen</goal>
91              </goals>
92            </execution>
93          </executions>
94          <configuration>
95            <sei>ifmt.cba.servico.ServicoControleEstoqueImpl</sei>
96            <genWsdl>true</genWsdl>
97            <resourceDestDir>${project.build.outputDirectory}/</resourceDestDir>
98          </configuration>
99        </plugin>
100     </plugins>
101   </build>
102 </project>
```

VO - ProdutoVO I

Classe ProdutoVO representa cada produto mantido no estoque.
Implementar no subpacote **vo**.

```java
package ifmt.cba.vo;

public class ProdutoVO {
    private int codigo;
    private String nome;
    private int estoque;

    public ProdutoVO(){
        this.estoque = 0;
    }

    public int getCodigo() {
        return codigo;
    }

    public void setCodigo(int codigo) {
        this.codigo = codigo;
    }

    public String getNome() {
        return nome;
    }

    public void setNome(String nome) {
```

VO - ProdutoVO II

```
25          this.nome = nome;
26      }
27
28      public int getEstoque() {
29          return estoque;
30      }
31
32      public void setEstoque(int estoque) {
33          this.estoque = estoque;
34      }
35
36      public void adicionarEstoque(int quantidade) throws Exception {
37          if (quantidade > 0) {
38              this.estoque += quantidade;
39          } else {
40              throw new Exception("Quantidade deve ser maior que zero");
41          }
42      }
43
44      public void baixarEstoque(int quantidade) throws Exception {
45          if (quantidade > 0) {
46              if (quantidade <= this.estoque) {
47                  this.estoque -= quantidade;
48              } else {
49                  throw new Exception("Estoque insuficiente...");
50              }
51          } else {
52              throw new Exception("Quantidade deve ser maior que zero");
53          }
```

VO - ProdutoVO III

```
54      }
55
56      @Override
57      public boolean equals(Object obj) {
58          if (this == obj) {
59              return true;
60          }
61          if (obj == null) {
62              return false;
63          }
64          if (getClass() != obj.getClass()) {
65              return false;
66          }
67          final ProdutoVO other = (ProdutoVO) obj;
68          if (this.codigo != other.codigo) {
69              return false;
70          }
71          return true;
72      }
73
74
75  }
```

Código 1: ProdutoVO.java

Classe de Negócio - GerenciadorEstoque I

Mantém uma lista de ProdutoVO e implementa as operações de gerenciamento dessa lista e dos dados que estão nessa lista. Implementar no subpacote **negocio**.

```
1  package ifmt.cba.negocio;
2
3  import java.util.ArrayList;
4
5  import ifmt.cba.vo.ProdutoVO;
6
7  public class GerenciadorEstoque {
8
9
10     private ArrayList<ProdutoVO> listaProduto;
11
12     public GerenciadorEstoque() {
13         this.listaProduto = new ArrayList<ProdutoVO>();
14     }
15
16     public void adicionarProduto(ProdutoVO produtoVO) throws Exception {
17         if (produtoVO != null) {
18             if (this.buscarProdutoPorCodigo(produtoVO.getCodigo()) == null) {
19                 this.listaProduto.add(produtoVO);
20             } else {
21                 throw new Exception("Produto ja existe");
22             }
```

Classe de Negócio - GerenciadorEstoque II

```
23          } else {
24              throw new Exception("Produto nao pode ser nulo");
25          }
26      }
27
28      public void removerProduto(ProdutoVO produtoVO) throws Exception {
29          if (produtoVO != null) {
30              if (this.listaProduto.indexOf(produtoVO) >= 0) {
31                  this.listaProduto.remove(produtoVO);
32              } else {
33                  throw new Exception("Produto nao localizado");
34              }
35          } else {
36              throw new Exception("Produto nao pode ser nulo");
37          }
38      }
39
40      public void adicionarEstoqueProduto(ProdutoVO produtoVO, int quantidade)
            throws Exception {
41          if (produtoVO != null || quantidade > 0) {
42              if (this.listaProduto.indexOf(produtoVO) >= 0) {
43                  ProdutoVO produtoVOTemp = this.listaProduto.get(this.listaProduto
                        .indexOf(produtoVO));
44                  produtoVOTemp.adicionarEstoque(quantidade);
45              } else {
46                  throw new Exception("Produto nao localizado");
47              }
48          } else {
49              throw new Exception("Produto ou quantidade inconsistente");
```

Classe de Negócio - GerenciadorEstoque III

```
50          }
51      }
52
53      public void baixarEstoqueProduto(ProdutoVO produtoVO, int quantidade) throws
            Exception {
54          if (produtoVO != null || quantidade > 0) {
55              if (this.listaProduto.indexOf(produtoVO) >= 0) {
56                  ProdutoVO produtoVOTemp = this.listaProduto.get(this.listaProduto
                        .indexOf(produtoVO));
57                  produtoVOTemp.baixarEstoque(quantidade);
58              } else {
59                  throw new Exception("Produto nao localizado");
60              }
61          } else {
62              throw new Exception("Produto ou quantidade inconsistente");
63          }
64      }
65
66      public ProdutoVO buscarProdutoPorCodigo(int codigo) {
67
68          ProdutoVO produtoVOTemp = null;
69
70          for (ProdutoVO produtoVO : this.listaProduto) {
71              if (produtoVO.getCodigo() == codigo) {
72                  produtoVOTemp = produtoVO;
73                  break;
74              }
75          }
76
```

Classe de Negócio - GerenciadorEstoque IV

```
77          return produtoVOTemp;
78      }
79
80      public int contadorProduto() {
81          return this.listaProduto.size();
82      }
83
84      public ArrayList<ProdutoVO> listaProduto() {
85          return this.listaProduto;
86      }
87
88      public int totalEstoqueFisico() {
89          int total = 0;
90          for (ProdutoVO produtoVO : this.listaProduto) {
91              total += produtoVO.getEstoque();
92          }
93          return total;
94      }
95
96  }
```

Código 2: GerenciadorEstoque.java

SEI - ServicoControleEstoque I

Implementa a interface do Web Service que publica as operações da classe GerenciadorEstoque. Implementar no subpacote **servico**.

```java
package ifmt.cba.servico;

import java.util.ArrayList;

import ifmt.cba.vo.ProdutoVO;
import jakarta.jws.WebMethod;
import jakarta.jws.WebService;
import jakarta.jws.soap.SOAPBinding;
import jakarta.jws.soap.SOAPBinding.Style;

//Service Endpoint Interface(SEI)

@WebService
@SOAPBinding(style = Style.DOCUMENT)
public interface ServicoControleEstoque {

    @WebMethod()
    public void adicionarProduto(ProdutoVO produtoVO) throws Exception;

    @WebMethod()
    public void removerProduto(ProdutoVO produtoVO) throws Exception;

    @WebMethod()
```

SEI - ServicoControleEstoque II

```
24      public void adicionarEstoqueProduto(ProdutoVO produtoVO, int quantidade)
            throws Exception;
25
26      @WebMethod()
27      public void baixarEstoqueProduto(ProdutoVO produtoVO, int quantidade) throws
            Exception;
28
29      @WebMethod()
30      public int contadorProduto();
31
32      @WebMethod()
33      public ProdutoVO buscarProdutoPorCodigo(int codigo);
34
35      @WebMethod()
36      public ArrayList<ProdutoVO> listaProduto();
37
38      @WebMethod()
39      public int totalEstoqueFisico();
40
41  }
```

Código 3: ServicoControleEstoque.java

SIB - ServicoControleEstoqueImpl I

Implementa o Web Service que implementa as operações da classe GerenciadorEstoque. Implementar no subpacote **servico**.

```java
package ifmt.cba.servico;

import java.util.ArrayList;

import ifmt.cba.negocio.GerenciadorEstoque;
import ifmt.cba.vo.ProdutoVO;
import jakarta.jws.WebService;

//Service Implementation Bean (SIB)

@WebService(endpointInterface = "ifmt.cba.servico.ServicoControleEstoque")
public class ServicoControleEstoqueImpl implements ServicoControleEstoque{

    private GerenciadorEstoque gerenciadorEstoque;

    public ServicoControleEstoqueImpl() {
        this.gerenciadorEstoque = new GerenciadorEstoque();
    }

    @Override
    public void adicionarProduto(ProdutoVO produtoVO) throws Exception {
        this.gerenciadorEstoque.adicionarProduto(produtoVO);
    }

```

SIB - ServicoControleEstoqueImpl II

```
25      @Override
26      public void removerProduto(ProdutoVO produtoVO) throws Exception {
27          this.gerenciadorEstoque.removerProduto(produtoVO);
28      }
29
30      @Override
31      public void adicionarEstoqueProduto(ProdutoVO produtoVO, int quantidade)
            throws Exception {
32          this.gerenciadorEstoque.adicionarEstoqueProduto(produtoVO, quantidade);
33      }
34
35      @Override
36      public void baixarEstoqueProduto(ProdutoVO produtoVO, int quantidade) throws
            Exception {
37          this.gerenciadorEstoque.baixarEstoqueProduto(produtoVO, quantidade);
38      }
39
40      @Override
41      public int contadorProduto() {
42          return this.gerenciadorEstoque.contadorProduto();
43      }
44
45      @Override
46      public ProdutoVO buscarProdutoPorCodigo(int codigo) {
47          return this.gerenciadorEstoque.buscarProdutoPorCodigo(codigo);
48      }
49
50      @Override
51      public ArrayList<ProdutoVO> listaProduto() {
```

SIB - ServicoControleEstoqueImpl III

```
52          return this.gerenciadorEstoque.listaProduto();
53      }
54
55      @Override
56      public int totalEstoqueFisico() {
57          return this.gerenciadorEstoque.totalEstoqueFisico();
58      }
59
60  }
```

Código 4: ServicoControleEstoqueImpl.java

Preparando Projeto Java/Maven do Lado Cliente

Criar um projeto Java/Maven com as seguintes definições:

groupId: ifmt.cba

artifactId: cliente-estoque

version: 1.0-SNAPSHOT

Alterando o arquivo POM I

```xml
<?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>ifmt.cba</groupId>
  <artifactId>cliente-estoque</artifactId>
  <version>1.0-SNAPSHOT</version>

  <name>cliente-estoque</name>

   <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <maven.compiler.source>17</maven.compiler.source>
    <maven.compiler.target>17</maven.compiler.target>
  </properties>

<dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.11</version>
      <scope>test</scope>
    </dependency>

    <!-- https://mvnrepository.com/artifact/jakarta.xml.ws/jakarta.xml.ws-api -->
```

Alterando o arquivo POM II

```
28      <dependency>
29        <groupId>jakarta.xml.ws</groupId>
30        <artifactId>jakarta.xml.ws-api</artifactId>
31        <version>4.0.0</version>
32      </dependency>
33
34      <!-- https://mvnrepository.com/artifact/com.sun.xml.ws/jaxws-rt -->
35      <dependency>
36        <groupId>com.sun.xml.ws</groupId>
37        <artifactId>jaxws-rt</artifactId>
38        <version>4.0.1</version>
39      </dependency>
40
41      <!-- https://mvnrepository.com/artifact/com.sun.xml.ws/jaxws-ri -->
42      <dependency>
43        <groupId>com.sun.xml.ws</groupId>
44        <artifactId>jaxws-ri</artifactId>
45        <version>4.0.1</version>
46        <type>pom</type>
47      </dependency>
48
49      <!-- https://mvnrepository.com/artifact/jakarta.xml.bind/jakarta.xml.bind-api
            -->
50      <dependency>
51        <groupId>jakarta.xml.bind</groupId>
52        <artifactId>jakarta.xml.bind-api</artifactId>
53        <version>4.0.0</version>
54      </dependency>
55
```

Alterando o arquivo POM III

```
56      <!-- https://mvnrepository.com/artifact/com.sun.xml.bind/jaxb-impl -->
57      <dependency>
58        <groupId>com.sun.xml.bind</groupId>
59        <artifactId>jaxb-impl</artifactId>
60        <version>4.0.2</version>
61      </dependency>
62    </dependencies>
63
64    <build>
65      <plugins>
66        <plugin>
67          <groupId>org.apache.maven.plugins</groupId>
68          <artifactId>maven-compiler-plugin</artifactId>
69          <version>3.11.0</version>
70        </plugin>
71        <plugin>
72          <groupId>com.sun.xml.ws</groupId>
73          <artifactId>jaxws-maven-plugin</artifactId>
74          <version>4.0.1</version>
75          <executions>
76            <execution>
77              <goals>
78                <goal>wsimport</goal>
79              </goals>
80            </execution>
81          </executions>
82          <configuration>
83            <wsdlUrls>
84              <wsdlUrl>http://localhost:8083/servico/estoque?wsdl</wsdlUrl>
```

Alterando o arquivo POM IV

```
85              </wsdlUrls>
86              <keep>true</keep>
87              <packageName>ifmt.cba.servico</packageName>
88              <sourceDestDir>src/main/java</sourceDestDir>
89            </configuration>
90          </plugin>
91        </plugins>
92    </build>
93 </project>
```

Cliente do Serviço ServicoGerenciadorEstoque I

Cliente que consome os serviços.

```
1  package ifmt.cba;
2
3  import java.net.MalformedURLException;
4  import java.net.URL;
5  import java.util.List;
6
7  import javax.swing.JOptionPane;
8  import javax.xml.namespace.QName;
9
10 import ifmt.cba.servico.ProdutoVO;
11 import ifmt.cba.servico.ServicoControleEstoque;
12 import jakarta.xml.ws.Service;
13
14 public class App {
15
16     private static ServicoControleEstoque controleEstoque;
17
18     public static void main(String[] args) {
19
20         URL url;
21         try {
22             url = new URL("http://localhost:8083/servico/estoque?wsdl");
23             QName qname = new QName("http://servico.cba.ifmt/", "
                   ServicoControleEstoqueImplService");
24
25             Service service = Service.create(url, qname);
```

Cliente do Serviço ServicoGerenciadorEstoque II

```
26
27              controleEstoque = service.getPort(ServicoControleEstoque.class);
28          } catch (MalformedURLException e) {
29              e.printStackTrace();
30          }
31
32          if (controleEstoque != null) {
33              String opcoesMenu = "[1] Adicionar Produto\n[2] Remover Produto\n[3]
                    Adicionar Estoque\n[4] Baixar Estoque\n"
34                      + "[5] Contar Produtos\n[6] Contar Estoque FiÂ sico\n[7]
                            Listar Produtos\n[8] Sair";
35              int opcao;
36              do {
37                  opcao = Integer.parseInt(JOptionPane.showInputDialog(null,
                        opcoesMenu));
38
39                  switch (opcao) {
40                      case 1:
41                          novoProduto();
42                          break;
43                      case 2:
44                          removerProduto();
45                          break;
46                      case 3:
47                          adicionarEstoque();
48                          break;
49                      case 4:
50                          baixarEstoque();
51                          break;
```

Cliente do Serviço ServicoGerenciadorEstoque III

```
52                      case 5:
53                          contarProdutos();
54                          break;
55                      case 6:
56                          contarEstoqueProdutos();
57                          break;
58                      case 7:
59                          listarProdutos();
60                          break;
61                  }
62              } while (opcao != 8);
63          }
64
65      }
66
67      private static void novoProduto() {
68          ProdutoVO produtoVOTemp = null;
69          int codigo;
70          String nome;
71          boolean sair = false;
72          do {
73              try {
74                  codigo = Integer.parseInt(JOptionPane.showInputDialog(null, "
                        Forneca o codigo do produto"));
75                  nome = JOptionPane.showInputDialog(null, "Forneca o nome do
                        produto");
76                  produtoVOTemp = new ProdutoVO();
77                  produtoVOTemp.setCodigo(codigo);
78                  produtoVOTemp.setNome(nome);
```

Cliente do Serviço ServicoGerenciadorEstoque IV

```
79                  controleEstoque.adicionarProduto(produtoVOTemp);
80                  sair = true;
81              } catch (Exception ex) {
82                  JOptionPane.showMessageDialog(null, "Erro ao executar a operacao
                        " + ex.getMessage());
83              }
84          } while (!sair);
85      }
86
87      private static void removerProduto() {
88          ProdutoVO produtoVOTemp = null;
89          int codigo;
90          try {
91              codigo = Integer.parseInt(JOptionPane.showInputDialog(null, "Forneca
                    o codigo do produto"));
92              produtoVOTemp = controleEstoque.buscarProdutoPorCodigo(codigo);
93              if (produtoVOTemp != null) {
94                  controleEstoque.removerProduto(produtoVOTemp);
95              } else {
96                  JOptionPane.showMessageDialog(null, "Produto nao localizado");
97              }
98          } catch (Exception ex) {
99              JOptionPane.showMessageDialog(null, "Dados incosistentes");
00          }
01      }
02
03      private static void adicionarEstoque() {
04          ProdutoVO produtoVOTemp = null;
05          int codigo;
```

Cliente do Serviço ServicoGerenciadorEstoque V

```
106         int quantidade;
107         boolean sair = false;
108         do {
109             try {
110                 codigo = Integer.parseInt(JOptionPane.showInputDialog(null, "
                        Forneca o codigo do produto"));
111                 produtoVOTemp = controleEstoque.buscarProdutoPorCodigo(codigo);
112                 if (produtoVOTemp != null) {
113                     quantidade = Integer.parseInt(
114                             JOptionPane.showInputDialog(null, "Forneca a
                                quantidade a ser adicionada ao estoque"));
115                     controleEstoque.adicionarEstoqueProduto(produtoVOTemp,
                            quantidade);
116                     sair = true;
117                 } else {
118                     JOptionPane.showMessageDialog(null, "Produto nao localizado")
                            ;
119                 }
120             } catch (Exception ex) {
121                 JOptionPane.showMessageDialog(null, "Erro ao executar a operacao
                        " + ex.getMessage());
122             }
123         } while (!sair);
124     }
125
126     private static void baixarEstoque() {
127         ProdutoVO produtoVOTemp = null;
128         int codigo;
129         int quantidade;
```

Cliente do Serviço ServicoGerenciadorEstoque VI

```
130         boolean sair = false;
131         do {
132             try {
133                 codigo = Integer.parseInt(JOptionPane.showInputDialog(null, "
                        Forneca o codigo do produto"));
134                 produtoVOTemp = controleEstoque.buscarProdutoPorCodigo(codigo);
135                 if (produtoVOTemp != null) {
136                     quantidade = Integer.parseInt(
137                             JOptionPane.showInputDialog(null, "Forneca a
                                quantidade a ser baixada do estoque"));
138                     controleEstoque.baixarEstoqueProduto(produtoVOTemp,
                            quantidade);
139                     sair = true;
140                 } else {
141                     JOptionPane.showMessageDialog(null, "Produto nao localizado")
                            ;
142                 }
143             } catch (Exception ex) {
144                 JOptionPane.showMessageDialog(null, "Erro ao executar a operacao
                        " + ex.getMessage());
145             }
146         } while (!sair);
147     }
148
149     private static void contarProdutos() {
150
151         System.out.println("-----------------------------------");
152         System.out.println("Quantidade de Produtos: " + controleEstoque.
                contadorProduto());
```

Cliente do Serviço ServicoGerenciadorEstoque VII

```
153
154         }
155
156         private static void contarEstoqueProdutos() {
157
158             System.out.println("-------------------------------------");
159             System.out.println("Total Estoque Fisico dos Produtos: " +
                    controleEstoque.totalEstoqueFisico());
160
161         }
162
163         private static void listarProdutos() {
164
165             List<ProdutoVO> listaProduto = controleEstoque.listaProduto();
166             for (ProdutoVO produtoTemp : listaProduto) {
167                 System.out.println("-------------------------------------");
168                 System.out.println("Codigo: " + produtoTemp.getCodigo());
169                 System.out.println("Nome: " + produtoTemp.getNome());
170                 System.out.println("Estque: " + produtoTemp.getEstoque());
171             }
172         }
173
174 }
```

Código 5: Cliente.java

Ampliação de Funcionalidades I

Acrescente as seguintes características e funcionalidades ao estudo de caso:

- Na classe ProdutoVO, acrescentar o campo valorUnitario (float)
- Na classe GerenciadorEstoque acrescente dois novos métodos:
  - **float totalizarValorEstoqueProduto(ProdutoVO produtoVO)** - devolve o valor do estoque de um produto (valorUnitario * quantidade em estoque)
  - **float totalizarValorEstoqueGeral()** - calcula o valor do estoque de todos os produtos
  - Replicar os novos métodos da classe GerenciadorEstoque para a classe do serviço, publicando as novas capacidades
  - Adicionar as novas funcionalidades no lado cliente (opções de menu) para que possam ser invocadas no lado cliente