

CONGRATULAÇÕES!

**Em instantes começaremos
nossa aula de hoje !!!**



Antonio Andrade .:

Membro ANPPD® | Professor
Mestre TI | Trasmontano Saúde TI...



LinkedIn



Prof. Antonio Andrade
a.a.s@uni9.pro.br

Inovação Tecnológica, Sistemas e Segurança da Informação

Olá! Congratulações!

Teste sua conexão (vídeo e áudio)

Relate no chat. Tudo bem com áudio e vídeo?

Teste de mídia



Em instantes começaremos nossa aula de hoje !!!



ANTONIO ANDRADE

LATTES : <http://lattes.cnpq.br/9400427944771244>

LINKEDIN : <https://www.linkedin.com/in/antonio-andrade-msc/>

FACEBOOK : <https://www.facebook.com/face.aas>

E-MAIL : a.a.s@uni9.pro.br



{ antonio
andrade }



MEIO ACADÊMICO & CORPORATIVO

- » Graduado em Tecnologia da Informação;
- » **Mestre** em Engenharia Biomédica;
- » Atua como **professor** em Instituição de Ensino Superior (IES) desde 2009.
- » Certificações na área de **Privacidade**
- » DPO - Membro do **Comitê de Segurança da APDados (Associação dos Profissionais de Privacidade de Dados)**
- » Sócio das empresas **Viaclient** e **SMIT** com foco em ERP e Inovações Tecnológicas;
- » Coordenador de **desenvolvimento de sistemas** em empresa da área da Saúde;
- » Membro do **Comitê de LGPD** em empresa da área da Saúde e organização do terceiro setor;
- » **Palestrante** em eventos e capacitações relacionados à GDPR e LGPD.



DW – Desenvolvimento Web

Delimitadores JSP

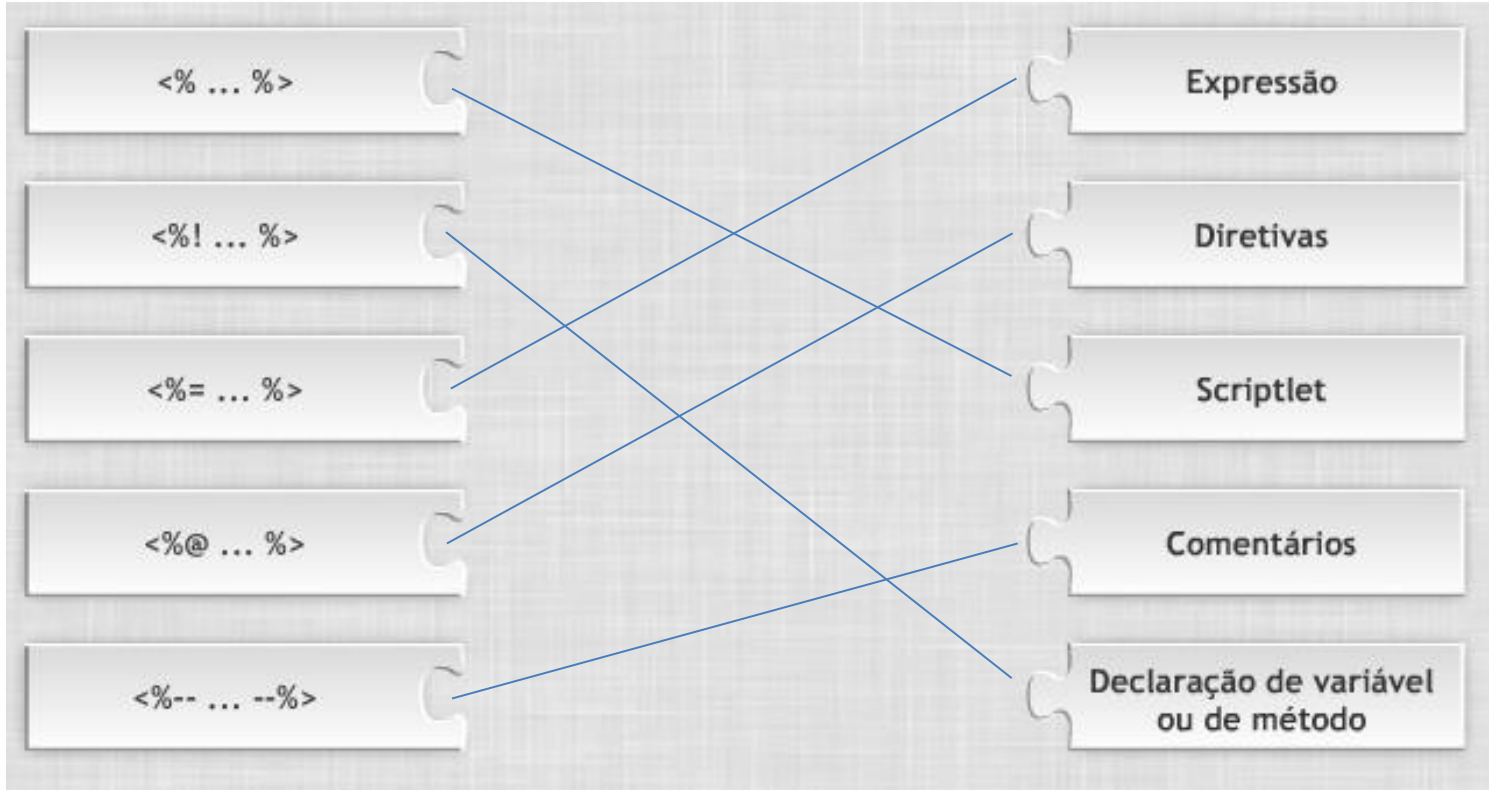
Cada delimitador tem uma funcionalidade específica.

A junção desses delimitadores dão toda flexibilidade necessário para programa em Java no JSP

<code><% ... %></code>	Scriptlet
<code><%! ... %></code>	Declaração de variável ou de método
<code><%= ... %></code>	Expressão
<code><%@ ... %></code>	Diretivas
<code><%-- ... --%></code>	Comentários

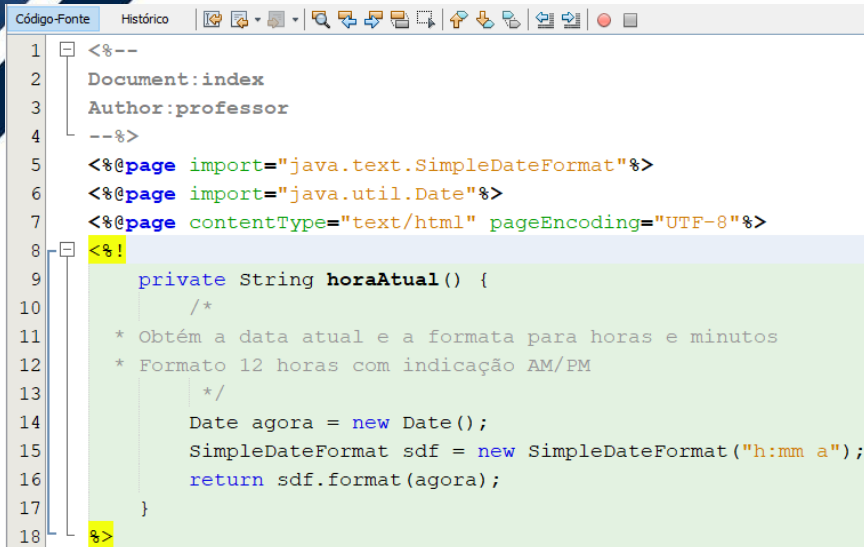
DW – Desenvolvimento Web

Delimitadores JSP - Teste

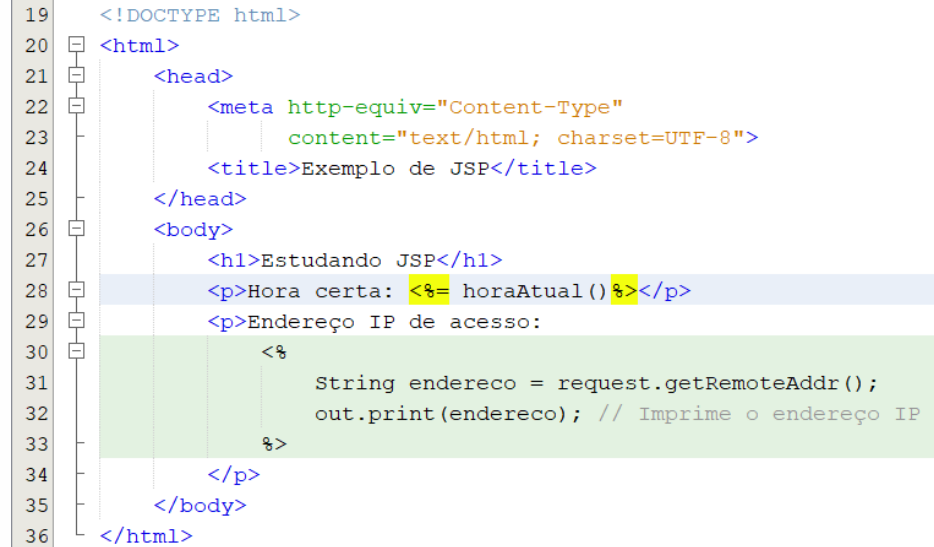


DW – Desenvolvimento Web

Exemplo de utilização dos delimitadores JSP

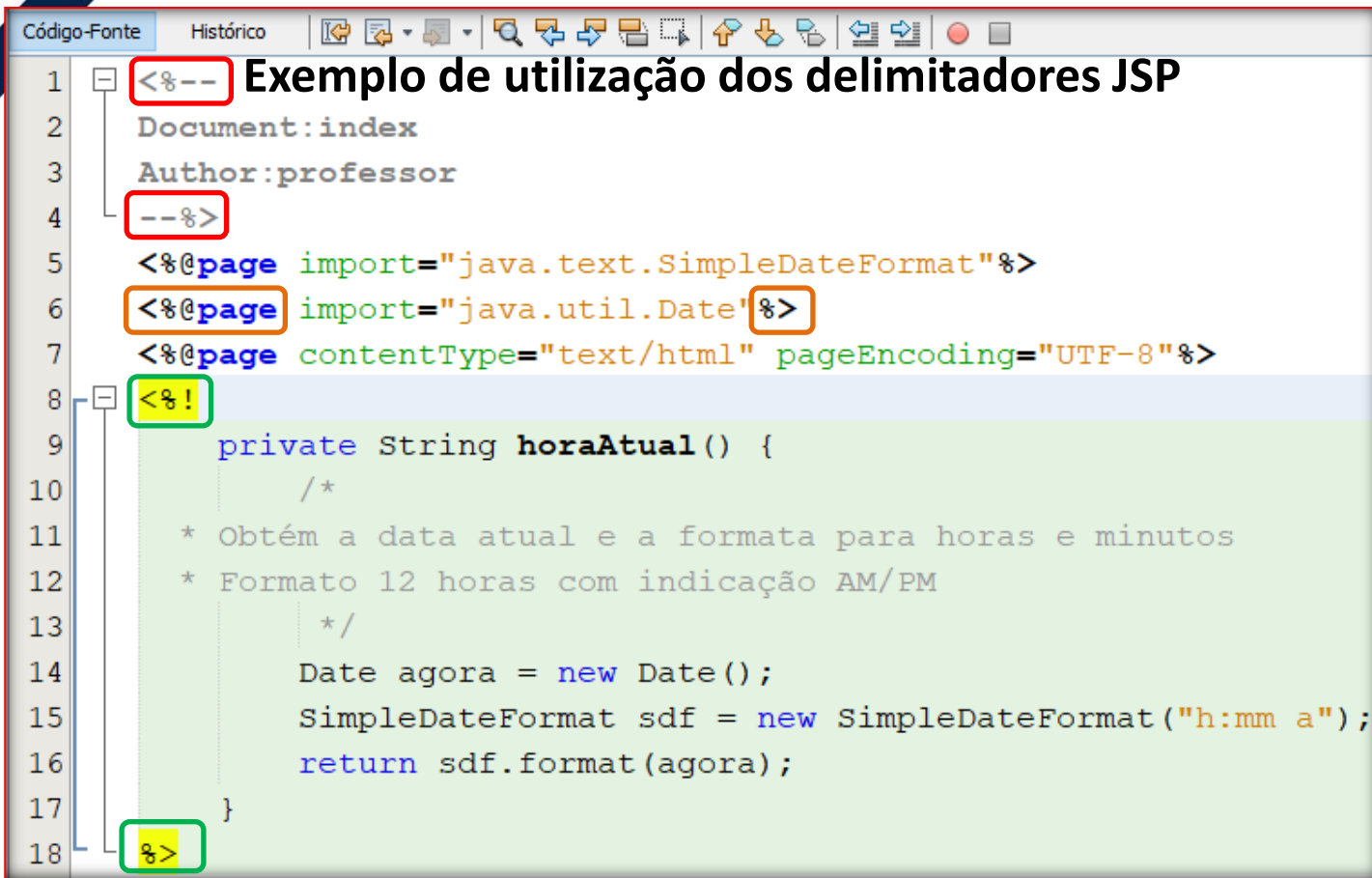


```
Código-Fonte  Histórico
1  <%--
2  Document:index
3  Author:professor
4  --%>
5  <%@page import="java.text.SimpleDateFormat"%>
6  <%@page import="java.util.Date"%>
7  <%@page contentType="text/html" pageEncoding="UTF-8"%>
8  <%!
9      private String horaAtual() {
10         /*
11         * Obtém a data atual e a formata para horas e minutos
12         * Formato 12 horas com indicação AM/PM
13         */
14         Date agora = new Date();
15         SimpleDateFormat sdf = new SimpleDateFormat("h:mm a");
16         return sdf.format(agora);
17     }
18  %>
```



```
19  <!DOCTYPE html>
20  <html>
21  <head>
22      <meta http-equiv="Content-Type"
23          content="text/html; charset=UTF-8">
24      <title>Exemplo de JSP</title>
25  </head>
26  <body>
27      <h1>Estudando JSP</h1>
28      <p>Hora certa: <%= horaAtual() %></p>
29      <p>Endereço IP de acesso:
30
31          <%
32              String endereco = request.getRemoteAddr();
33              out.print(endereco); // Imprime o endereço IP
34          %>
35      </p>
36  </body>
37  </html>
```

DW – Desenvolvimento Web



Código-Fonte | Histórico

1 **<%--** Exemplo de utilização dos delimitadores JSP

2 Document:index

3 Author:professor

4 **--%>**

5 <%@page import="java.text.SimpleDateFormat"%>

6 <%@page import="java.util.Date"%>

7 <%@page contentType="text/html" pageEncoding="UTF-8"%>

8 **<%!**

9 private String horaAtual() {

10 /*

11 * Obtém a data atual e a formata para horas e minutos

12 * Formato 12 horas com indicação AM/PM

13 */

14 Date agora = new Date();

15 SimpleDateFormat sdf = new SimpleDateFormat("h:mm a");

16 return sdf.format(agora);

17 }

18 **%>**

DW – Desenvolvimento Web

```
19 <!DOCTYPE html>
20 <html>
21   <head>
22     <meta http-equiv="Content-Type"
23           content="text/html; charset=UTF-8">
24     <title>Exemplo de JSP</title>
25   </head>
26   <body>
27     <h1>Estudando JSP</h1>
28     <p>Hora certa: <%= horaAtual() %></p>
29     <p>Endereço IP de acesso:
30       <%
31         String endereco = request.getRemoteAddr();
32         out.print(endereco); // Imprime o endereço IP
33       %>
34     </p>
35   </body>
36 </html>
```




DW – Desenvolvimento Web



Escrevendo páginas em JSP – Tipos de Dados

- **byte**: Inteiros de -128 a 127. Utiliza somente um byte em memória.
- **short**: Inteiros de -32.768 a 32.767 e ocupa dois bytes.
- **int**: Inteiros de -2.147.483.648 e 2.147.483.647, ocupando quatro bytes. (padrão utilizado)
- **long**: Não inteiros entre -9.223.372.036.854.775.808 e 9.223.372.036.854.775.807. (8 bytes)
- **float**: Não inteiros (quatro bytes de memória). Capacidade: 10^{38} .
- **double**: Não inteiros (oito bytes de memória). Capacidade: 10^{308} .
- **boolean**: Apenas os valores true ou false.
- **char**: Um byte (16 bits) - Armazenar um caractere. Padrão Unicode.





DW – Desenvolvimento Web

Escrevendo páginas em JSP – Estruturas de Decisão

if-then-else

```
26.         if (hora < 12) {
27.             /*
28.              * Se for antes de meio dia, dizer "Bom dia"
29.              * Utilizamos out.println para enviar um dado para o documento HTML
30.              */
31.             out.println("Bom dia");
32.         } else if (hora < 18) {
33.             /*
34.              * Não sendo antes do meio dia, mas sim antes das 18 horas
35.              * Escrever "Boa tarde"
36.              */
37.             out.println("Boa tarde");
38.         } else {
39.             // Em outras situações, escrever "Boa noite"
40.             out.println("Boa noite");
41.         }
```



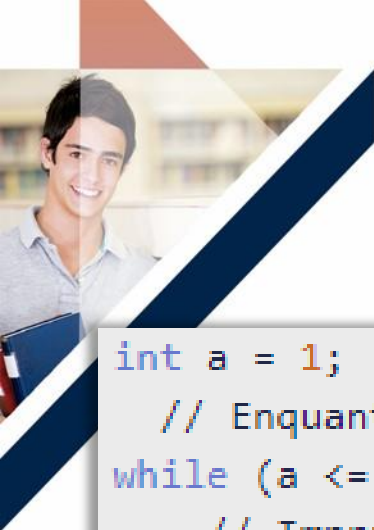
DW – Desenvolvimento Web

Escrevendo páginas em JSP Estruturas de Decisão

switch case

```
30.         switch (mês) {
31.             // Caso seja um dos três primeiros meses
32.             case 1:
33.             case 2:
34.             case 3:
35.                 out.println("Primeiro trimestre");
36.                 /*
37.                  * O break é necessário para ir direto
38.                  * para o fim desta estrutura
39.                  */
40.                 break;
41.             // Fazemos o mesmo para os outros trimestres
42.             case 4:
```

```
54.             case 12:
55.                 out.println("Quarto trimestre");
56.                 break;
57.             // Usamos default para outros valores
58.             default:
59.                 out.println("Valor incorreto");
60.                 /*
61.                  * Este break é desnecessário,
62.                  * mas geralmente utilizado
63.                  */
64.                 break;
65.         }
```



DW – Desenvolvimento Web

Escrevendo páginas em JSP – Estruturas de repetição

while, do while e for

```
int a = 1;
// Enquanto ela for menor ou igual
while (a <= 10) {
    // Imprimir um item da lista
    out.println("<li>" + a + "</li>");
    // Incrementar o valor de controle
    a++;
}
```

```
int a = 1;
// Inicia a repetição
do {
    // Imprimir um item da lista
    out.println("<li>" + a + "</li>");
    // Incrementar o valor de controle
    a++;
    // Decisão de saída no final da estrutura
while (a <= 10);
```

```
<%
// Declara a estrutura for
for (int a = 1; a <= 10; a++) {
    // Imprimir um item da lista
    out.println("<li>" + a + "</li>");
}
%>
```

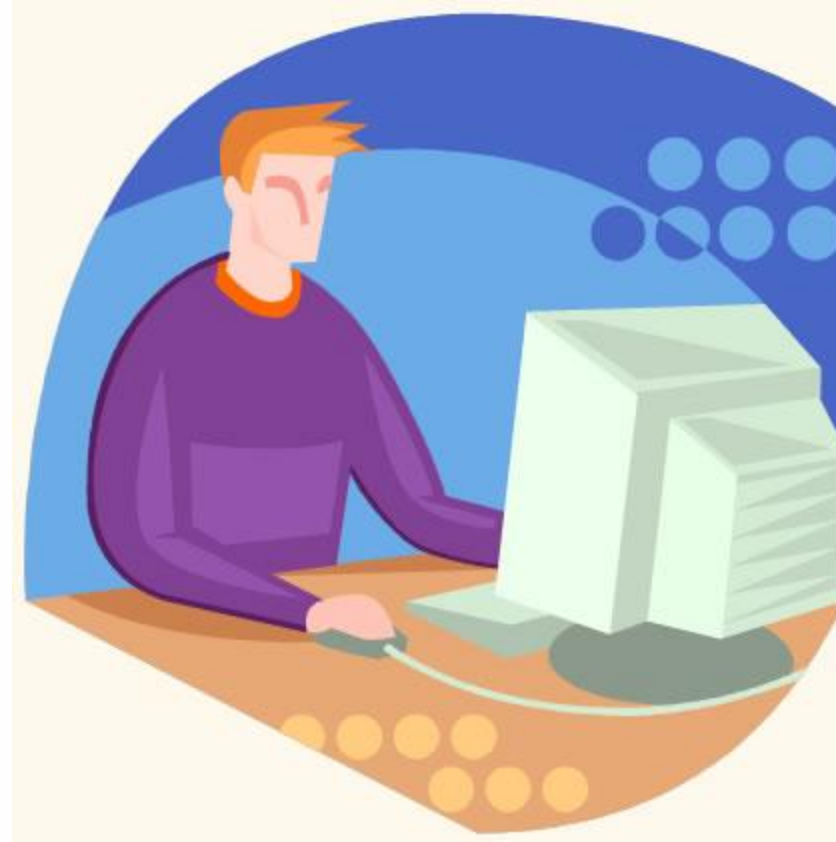


DW – Desenvolvimento Web

Vamos praticar...

Testar os exemplos das aulas do AVA:

- Estruturas de aplicações web
- Escrevendo páginas em JSP





DW – Desenvolvimento Web

ATIVIDADE



Criar um formulário em HTML que solicite Nome e Data de Nascimento. Este formulário deverá ser submetido à um JSP que irá dar como resposta, o dia da semana em que a pessoa nasceu.

Criar um formulário que exiba todos os dias de um determinado mês, e seus respectivos dias da semana.



DW – Desenvolvimento Web

EXERCÍCIO / ATIVIDADE

Com base no tutorial da aula anterior, construa uma aplicação que trabalhe com um produto.

Atividade valerá 3 (três) pontos na AV1, a ser feita e entregue pelo Class no dia 28.

* Mas pode ser adiantada a partir de já!

➤ Nome do Projeto: 2024_1_Produtos

➤ **CONTINUAÇÃO DO TREINAMENTO COM HTML, JSP, SERVLET E CLASSES JAVA**

DW – Desenvolvimento Web

EXERCÍCIO / ATIVIDADE

Classe Produto a ser criada, com seus respectivos atributos e métodos:

```
1 package model;
2 public class Produto {
3     // Atributos com get/set:
4     private int codigo;
5     private String descricao;
6     private int quantidade;
7     private float valorUnitario;
8     private String dataUltimaCompra;
9     // Atributos apenas com get:
10    private float valorTotal;
11    private float valorVendaUnitario;
12    private float valorVendaTotal;
13    // Construtor vazio:
14    public Produto() {...2 linhas }
15    // Construtor com parâmetros:
16    public Produto(int codigo, String descricao, int quantidade,
17        float valorUnitario, String dataUltimaCompra) {...6 linhas }
18    public int getCodigo() {...3 linhas }
19    public void setCodigo(int codigo) {...3 linhas }
20    public void setDescricao(String descricao) {...3 linhas }
21    public int getQuantidade() {...3 linhas }
22    public void setQuantidade(int quantidade) {...4 linhas }
23    public float getValorUnitario() {...3 linhas }
24    public void setValorUnitario(float valorUnitario) {...4 linhas }
25    public String getDataUltimaCompra() {...3 linhas }
26    public void setDataUltimaCompra(String dataUltimaCompra) {...3 linhas }
27    public float getValorTotal() {...3 linhas }
28    public float getValorVendaTotal() {...3 linhas }
29    public float getValorVendaUnitario() {...3 linhas }
30    private void calcular() {...5 linhas }
31 }
```



```
1 package model;
2 public class Produto {
3     // Atributos com get/set:
4     private int codigo;
5     private String descricao;
6     private int quantidade;
7     private float valorUnitario;
8     private String dataUltimaCompra;
9     // Atributos apenas com get:
10    private float valorTotal;
11    private float valorVendaUnitario;
12    private float valorVendaTotal;
13    // Construtor vazio:
14    + public Produto() {...2 linhas }
16    // Construtor com parâmetros:
17    ? public Produto(int codigo, String descricao, int quantidade,
18    +     float valorUnitario, String dataUltimaCompra) {...8 linhas }
26    + public int getCodigo() {...3 linhas }
29    + public void setCodigo(int codigo) {...3 linhas }
32    + public void setDescricao(String descricao) {...3 linhas }
35    + public int getQuantidade() {...3 linhas }
38    + public void setQuantidade(int quantidade) {...4 linhas }
42    + public float getValorUnitario() {...3 linhas }
45    + public void setValorUnitario(float valorUnitario) {...4 linhas }
49    + public String getDataUltimaCompra() {...3 linhas }
52    + public void setDataUltimaCompra(String dataUltimaCompra) {...3 linhas }
55    + public float getValorTotal() {...3 linhas }
56    + public float getValorVendaTotal() {...3 linhas }
61    + public float getValorVendaUnitario() {...3 linhas }
64    + private void calcular() {...5 linhas }
69 }
```



DW – Desenvolvimento Web

Opções:

[Cadastro de produtos](#)

[Consulta](#)

Cadastro de Produtos

Código: 1
Descrição: Coca-Cola
Quantidade: 100
Valor Unitário: 3
Data última compra: 07/03/2018

Enviar

Reposta Cadastro - (Produto)!

Código: 1
Descrição: Coca-Cola
Quantidade: 100
Valor Unitário: R\$ 3,00
Valor Total: R\$ 300,00
Data última compra: 2018-03-07
Valor Unitário Venda: R\$ 3,90
Valor Total de Venda: R\$ 390,00



DICAS PARA ESTUDO:

- Seja “**CURIOSO**”:
 - Procure revisar o que foi estudado.
 - Pesquise as referências bibliográficas.
- Seja “**ANTENADO**”:
 - Leia a próxima aula pelo AVA.
- Seja “**COLABORATIVO**”:
 - Publique assuntos relevantes nos fóruns do AVA.
 - Coloque suas dúvidas relativas às aulas nos fóruns do AVA.
 - Responda à dúvidas dos colegas nos fóruns do AVA.

Prof. Antonio Andrade

a.a.s@uni9.pro.br

