

## 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](mailto: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](mailto: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.





```
1. <%@page import="Suporte.Unidades"%>
2. <%@page contentType="text/html" pageEncoding="UTF-8"%>
3. <!DOCTYPE html>
4. <html>
5.   <head>
6.     <title>Teste de Dados e Erro em JSP</title>
7.   </head>
8.   <body>
9.     <h3>Cadastro de Item com Tratamento - Exemplo</h3>
10.    <div id="frm">
11.      <form name="frmItens" action="ExTrataErroJSP_Dados.jsp" method="post">
12.        <pre>
13.          *** Inclusão de itens de estoque ***
14.          =====
15.          Código:      <input type="text" name="codigo" size="8" />
16.          descrição:   <input type="text" name="descricao" size="40" />
17.          unidade:     <%=new Unidades().getUnidades() %>
18.          Saldo:       <input type="text" name="saldo" size="10" />
19.          Valor Unitário:<input type="text" name="valorUnitario" size="10" />
20.          Observações: <input type="text" name="obs" size="40" />
21.          <input type="submit" value="Incluir" name="acao">
22.          <input type="reset" value="Limpar">
23.        </pre>
24.      </form>
25.    </div>
26.  </body>
27. </html>
```

Arquivo:  
ExTrataErroJSP\_Form.jsp



# DW – Desenvolvimento Web

Exemplo da tela em execução:

## Cadastro de Item com Tratamento - Exemplo

\*\*\* Inclusão de intes de estoque \*\*\*

Código:

descrição:

unidade:

Saldo:

Valor Unitário:

Observações:



# DW – Desenvolvimento Web

Exemplo da tela em execução:

## Diretivas

Segundo Santos (2007), as diretivas são **TAGs** específicas que ajudam na configuração de páginas JSP. Estas diretivas são três: **page** que configura as propriedades de uma página JSP; **include** que insere o conteúdo de outro arquivo em uma página JSP e; **taglib** que pode estender um conjunto de TAGs que o contêiner pode interpretar.

Em consideração a uma boa prática de programação, colocaremos as diretivas **page** nas primeiras linhas da página.

Exemplo: `<@page import="Suporte.Unidades"`



```
1. <%@page contentType="text/html" pageEncoding="UTF-8" errorPage="TrataErro.jsp" %>
2. <!DOCTYPE html>
3. <html>
4.     <head>
5.         <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
6.         <title>Teste de Dados e Erro em JSP</title>
7.     </head>
8.     <body>
9.         <h3>Cadastro de Item com Tratamento - Com tratamento de Erro</h3>
10.        <%
11.            // Recebendo dados com método getParameter
12.            // através do objeto implícito request:
13.            int codigo = Integer.parseInt(request.getParameter("codigo"));
14.            String descricao = request.getParameter("descricao");
15.            String unidade = request.getParameter("unidade");
16.            float saldo = Float.parseFloat(request.getParameter("saldo"));
17.            float valorUnitario =
18.                Float.parseFloat(request.getParameter("valorUnitario"));
19.            String obs = request.getParameter("obs");
20.            String acao = request.getParameter("acao");
21.        %>
22.        <!-- Apresentação dos dados via HTML -->
23.        <pre>
24.            ----- DADOS OBTIDOS DO BROWSER -----
25.            Ação:          <%=acao%>
26.            Código:        <%=codigo%>
27.            Descrição:     <%=descricao%>
28.            unidade:       <%=unidade%>
29.            Saldo:         <%=saldo%>
30.            Valor Unitário: <%=valorUnitario%>
31.            Observações:   <%=obs%>
32.            -----
33.            Valor Total:   <%= (saldo * valorUnitario) %>
34.        </pre>
35.    </body>
36. </html>
```

Arquivo:  
ExTrataErroJSP\_Dados.jsp



# DW – Desenvolvimento Web

Exemplo da tela em execução:

## Cadastro de Item com Tratamento - Com tratamento de Erro

```
----- DADOS OBTIDOS DO BROWSER -----  
Ação:          Incluir  
Codigo:        1  
Descrição:     Freio KX  
unidade:       PC  
Saldo:         10.0  
Valor Unitário: 55.0  
Observações:   Produto descontinuado.  
-----  
Valor Total:   550.0
```





# DW – Desenvolvimento Web

## Objetos implícitos e explícitos

### Objetos implícitos

Como já foi demonstrado com a classe `unidades`, sabemos que para utilizar uma classe implementada pelo programador, basta colocar a diretiva **page** com o atributo `import`, indicando ao Java onde encontrar a classe desejada.

No entanto, "além dos objetos que o próprio programador pode instanciar em uma página JSP, ele também pode fazer uso de nove objetos que o contêiner instancia e gerencia.". (SANTOS, 2007, p.108)

São os objetos **application**, **config**, **exception**, **out**, **page**, **pageContext**, **request**, **response** e **session**, cada qual com sua funcionalidade, instanciando uma classe específica.





```
1. <%@page contentType="text/html" pageEncoding="UTF-8" isErrorPage="true"%>
2. <!DOCTYPE html>
3. <html>
4.     <head>
5.         <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
6.         <title>JSP Page</title>
7.         <style type="text/css">
8.             #erro {color:red; }
9.             #voltar {background-color: #00FF00}
10.            span {background-color: #FF0000; color: #FFFFFF; font-weight: bold}
11.            body {text-align: center}
12.        </style>
13.    </head>
14.    <body>
15.        <h2>Problemas Encontrados</h2>
16.        <hr>
17.        <div id="erro">
18.            Ocorreu um erro inesperado na execução do sistema.<br />
19.            Favor informar ao suporte técnico sobre o erro:<br>
20.            <span><%=exception%></span>
21.        </div>
22.        <hr>
23.        <form id="voltar">
24.            <input type="button" onclick="history.go(-1)" value="Voltar" />
25.        </form>
26.        <hr>
27.    </body>
28. </html>
```

Arquivo:  
TrataErro.jsp





# DW – Desenvolvimento Web

Com tratamento de erros

## Problemas Encontrados

Ocorreu um erro inesperado na execução do sistema.

Favor informar ao suporte técnico sobre o erro:

**java.lang.NumberFormatException: For input string: "IAAA"**

Voltar





# DW – Desenvolvimento Web

## Sem tratamento de erros



### HTTP Status 500 -

**type** Exception report

**message**

**description** The server encountered an internal error () that prevented it from fulfilling this request.

**exception**

`org.apache.jasper.JasperException: PWC6033: Error in Javac compilation for JSP`

`PWC6197: An error occurred at line: 20 in the jsp file: /TrataErro.jsp`

`PWC6199: Generated servlet error:`

`cannot find symbol`

`symbol: variable exception`

`location: class org.apache.jsp.TrataErro_jsp`

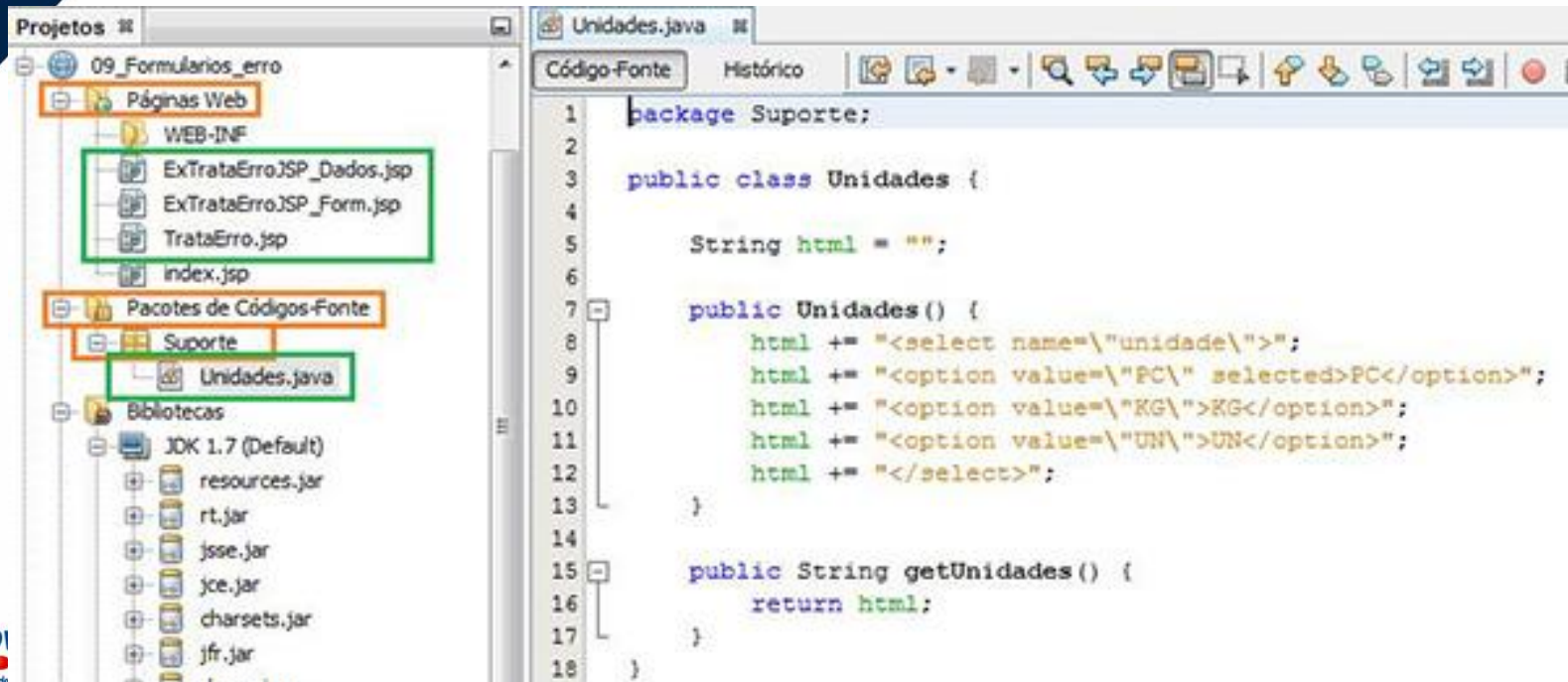
**note** The full stack traces of the exception and its root causes are available in the GlassFish Server Open Source Edition 3.1.2.2 logs.

GlassFish Server Open Source Edition 3.1.2.2



# DW – Desenvolvimento Web

## Classe Java extra e estrutura do projeto



The screenshot displays an IDE interface with two main panels. The left panel, titled 'Projetos', shows a project named '09\_Formularios\_erro'. Under the 'Páginas Web' folder, there are files: 'WEB-INF', 'ExTrataErroJSP\_Dados.jsp', 'ExTrataErroJSP\_Form.jsp', 'TrataErro.jsp', and 'Index.jsp'. The 'Pacotes de Códigos-Fonte' folder contains a 'Suporte' package, which in turn contains the 'Unidades.java' file. The right panel, titled 'Unidades.java', shows the source code of the 'Unidades' class. The code is as follows:

```
1 package Suporte;
2
3 public class Unidades {
4
5     String html = "";
6
7     public Unidades() {
8         html += "<select name=\"unidade\">";
9         html += "<option value=\"PC\" selected>PC</option>";
10        html += "<option value=\"KG\">KG</option>";
11        html += "<option value=\"UN\">UN</option>";
12        html += "</select>";
13    }
14
15    public String getUnidades() {
16        return html;
17    }
18 }
```



# DW – Desenvolvimento Web



## Testar os exemplos das aulas do AVA:

- Manutenção de dados e erros em JSP.
  - Estruturas de desvio condicional e de repetição em JSP:
    - Cálculo do IMC
    - Exibição das tabuadas
- 
- Mais emoções!???
    - Pergunte-me como!







# JavaBean

```
public class Pessoa{  
    private String nome;  
    private int idade;  
    public Pessoa(){  
    }  
    public String getNome(){  
        return nome;  
    }  
    public void setNome(String nome){  
        this.nome = nome;  
    }  
    public int getIdade(){  
        return idade;  
    }  
    public void setIdade(int idade){  
        this.idade = idade;  
    }  
}
```

Classe      Objeto

Atributos

Métodos

Estado

Encapsulamento



Hora de praticar !!!



Atividade



## Atividade - JavaBean

Criar um **novo** projeto chamado: **Folha**

- 1) Na página index, crie um formulário com os seguintes campos:
  - **Nome**
  - **Salário hora**
  - **Horas trabalhadas:**
  - **Horas extras:**
- 2) Criar um JavaBean para receber os campos do formulário e outros novos campos para:
  - **Salário horas normais:**
  - **Salário horas extras:**
  - **Salário bruto total:**
  - **FGTS:**
- 3) Criar um JSP para receber os dados do formulário, acionar o JavaBean e exibir as seguintes informações:
  - **Nome**
  - **Salário hora**
  - **Horas trabalhadas:**
  - **Horas extras:**
  - **Salário horas normais:**
  - **Salário horas extras:**
  - **Salário bruto total:**
  - **FGTS:**



# Atividade - JavaBean

Criar um **novo** projeto chamado: **Folha**

- 1) Na página index, crie um formulário com os seguintes campos:
  - **Nome:** **Jão Souza** **A**
  - **Salário hora:** **50.00** **B**
  - **Horas trabalhadas:** **150** **C**
  - **Horas extras:** **8** **D**
- 2) Criar um JavaBean para receber os campos do formulário e outros novos campos para:
  - **Salário horas normais:** **(B\*C) => 50,00 m\* 150 => 7500.00**
  - **Salário horas extras:** **(D \* (B \* 2) ) => 8 \* (50\*2) => 800,00**
  - **Salário bruto total:** **(7500+800) => 8,300**
  - **FGTS:** **664,00**
- 3) Criar um JSP para receber os dados do formulário, acionar o JavaBean e exibir as seguintes informações:
  - **Nome**
  - **Salário hora**
  - **Horas trabalhadas:**
  - **Horas extras:**
  - **Salário horas normais:**
  - **Salário horas extras:**
  - **Salário bruto total:**
  - **FGTS:**





# 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**

