

Instituto de

computação



Desenvolvimento Web

JSP

Java Server Pages

Prof. Leonardo Cruz.

leonardocruz@id.uff.br

Departamento de Computação, UFF

Contexto

```
import java.io.*;
import java.text.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class EchoServlet extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws IOException, ServletException {
        response.setContentType("text/html");
        ServletOutputStream out = response.getOutputStream();
        out.println("<HTML><BODY>");
        out.println("the value of parameter " + request.getParameter("parameter"));
        out.println("</BODY></HTML>");
    }
}
```

HTML + JAVA

Problema

- A interface HTML está difusa dentro do código Java
- Qualquer alteração visual implica em recompilação da Servlet
- Quem modifica os fontes: o Webmaster ou o programador?

```
package imc;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class ImcView extends HttpServlet {
    protected void processRequest(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        try {
            Double imc;
            imc = (Double)request.getAttribute("imc");
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Resultado do IMC</title>");
            out.println("</head>");
            out.println("<body>");
            out.println("<h1>Seu IMC é:</h1>");
            //out.println("<p>" + imc + "</p>");
            out.printf("<p>%2.5f</p>", imc);

            if (imc < 18.0) {
                out.println("<p>Cuidado! Abaixo!</p>");
            } else if (imc < 25.0) {
                out.println("<p>Parabéns! Peso ideal</p>");
            } else {
                out.println("<p>Cuidado! Acima!</p>");
            }

            out.println("</body>");
            out.println("</html>");
        } finally {
            out.close();
        }
    }
}
```

Servlet com código html

nada confortável!

Solução: JSP

■ Java Server Pages

- É uma página HTML com trechos de programa Java embutidos
- Simplificam a geração de conteúdo dinâmico para Web Designers
- A página JSP é automaticamente transformada em servlet

Separação: Lógica x Apresentação

- Uma página JSP pode tratar a geração de conteúdo, deixando o Servlet responsável apenas pela lógica de aplicação;
- Vantagens da tecnologia JSP:
 - Estende a tecnologia Servlet, herdando suas vantagens;
 - É portátil;
 - É escalável;

Como funciona uma página JSP

- Uma página JSP consiste em uma página HTML com alguns elementos especiais (tags) que conferem o caráter dinâmico da página;
- Código JSP é delimitado entre `<%` e `%>` (em PHP, delimitávamos o código com `<?php` e `?>`);
- Tais elementos podem tanto realizar um processamento por si, como podem recuperar o resultado do processamento realizado em um Servlet;

JSP: exemplo

■ Java Server Pages

```
<HTML>
<HEAD>
  <TITLE>JSP exemplo</TITLE>
</HEAD>
<BODY>
  <%
    out.println("Bom Dia "+ request.getParameter("name"));
  %>
</BODY>
</HTML>
```


JSP: exemplo

```
<%@page language = "java" %>
```

```
<HTML>
```

```
<HEAD>
```

```
<TITLE> Exemplo de Scriptlet </TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
<H1> Exemplo de Scriptlet </H1>
```

```
<%
```

```
    for (int i=0; i<10; i++)
```

```
    {
```

```
        out.print( "<b> Rodada numero: " + i + "</b>" );
```

```
        out.print( "<p>" );
```

```
        System.out.println( "Execucao " + i );
```

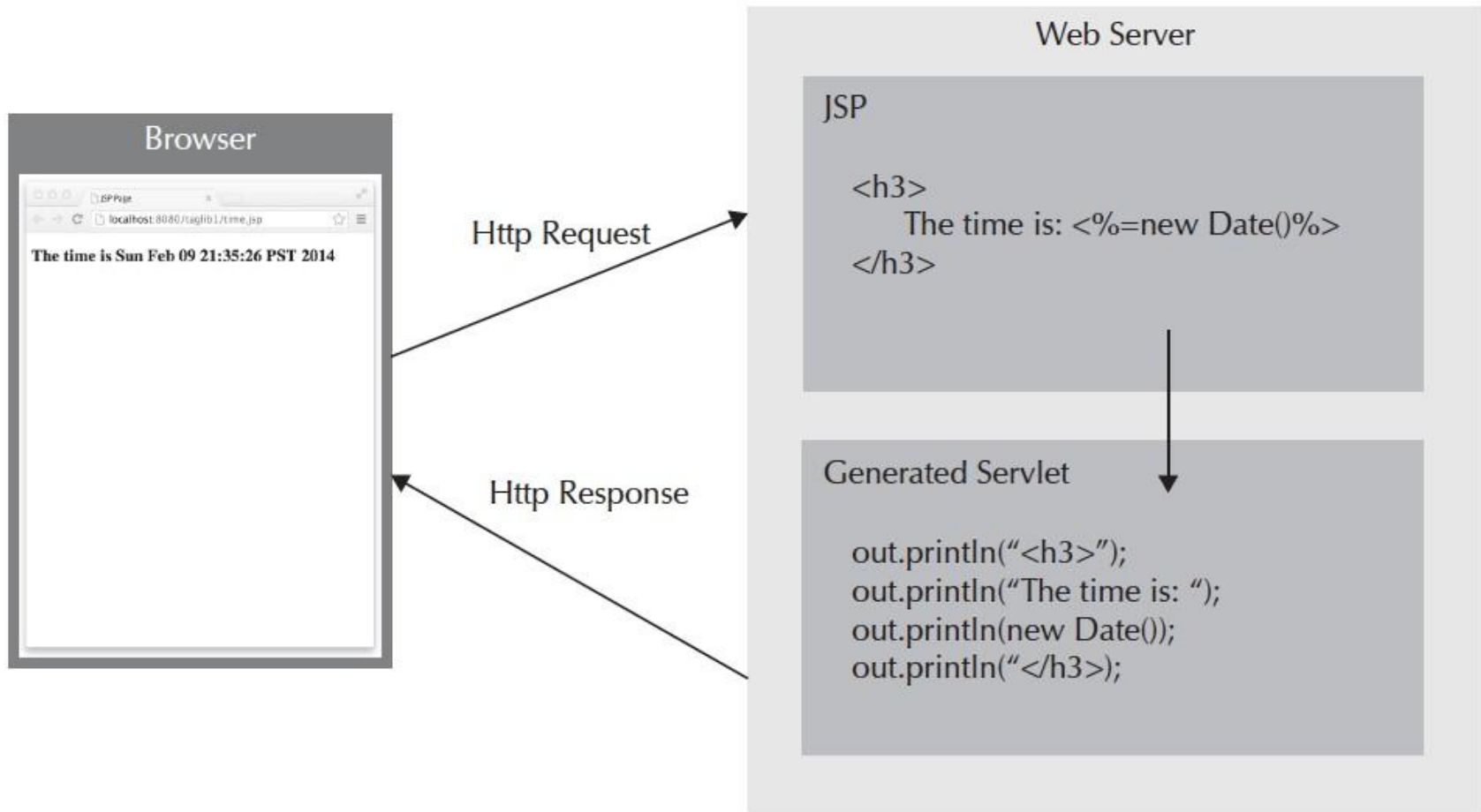
```
    }
```

```
%>
```

```
</BODY>
```

```
</HTML>
```

Transformação JSP → Servlet



JSP

- Facilita o desenvolvimento
 - Criar páginas JSP é mais fácil do que criar aplicações completas
 - Código Java é escrito no HTML (no Servlet é o oposto)
 - Mais fácil de alterar do que um Servlet (classe) e não precisam ser compiladas pelo desenvolvedor antes de entrarem em produção, como em um Servlet
- Indicado para apresentação, quando se tem **muito HTML e pouco Java**

JSP

- Se Servlet é muito Java com pouco HTML...
- JSP é muito HTML com pouco Java

JSP: Olá Mundo versão 0

The image shows a development environment with two windows. The top window is a code editor showing the source code of a JSP file named `ola.jsp`. The code is as follows:

```
1 <%@page contentType="text/html" pageEncoding="UTF-8"%>
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   </head>
8   <body>
9     <h1>Oi Mundo com JSP</h1>
10  </body>
11 </html>
```

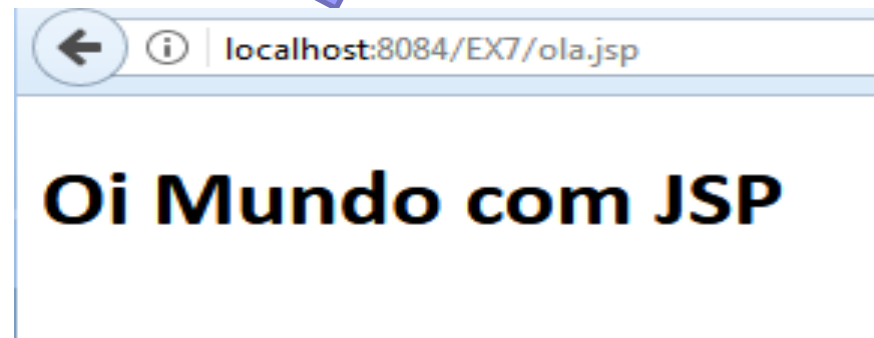
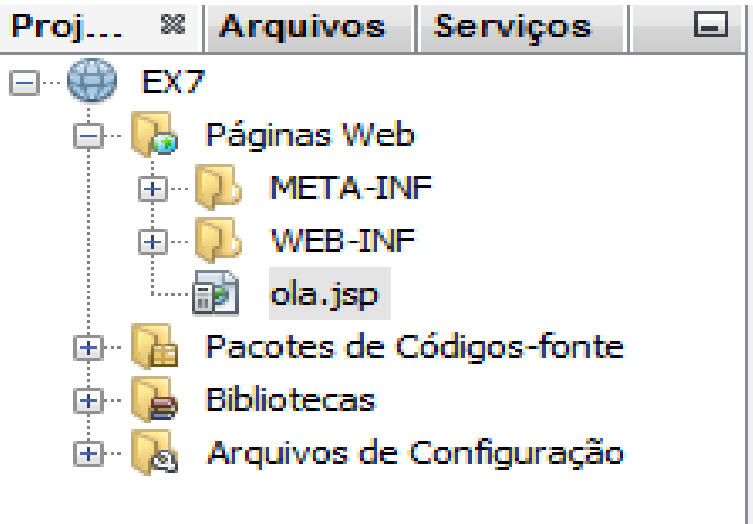
Two blue arrows point from the code to the browser window below. One arrow points from the `<@page>` tag on line 1 to the browser's address bar, and the other points from the `<h1>` tag on line 9 to the main content area of the browser.

The bottom window is a web browser displaying the output of the JSP page. The address bar shows `localhost:8084/EX7/ola.jsp`. The main content area displays the text:

Oi Mundo com JSP

JSP: Olá Mundo versão 1

```
5      <meta http-equiv="Content-Type" content="text/html";
6      <title>JSP Page</title>
7  </head>
8  <body>
9      <h1>
10     <%
11     out.println("Oi Mundo com JSP");
12     %>
13 </h1>
14 </body>
15 </html>
```



Vide fonte ex7

Elementos de script

■ Comentários

- `<%-----` isto é um comentário escondido `-----%>`

■ Declarações

- Definem variáveis ou métodos para uso subsequente
- `<%! int totalVisitas = 0; %>`
- `<%! public int getTime() { return time; } %>`

Elementos de script

■ Scriptlets

- Código Java a ser executado

```
<% for (int i = 0; i < 10; i++) { %>
```

```
<p>Teste de script</p>
```

```
<% } %>
```

■ Expressões

- Valores inseridos na página HTML gerada

O total de visitas é <%= ++total %>

A data atual é <%= new java.util.Date() %>

Elementos de script

■ Diretivas

- Permite definir propriedades gerais do JSP processadas no momento da tradução para Servlet

<%@ ... %>

■ Include

- Inclusão de arquivos na página

<%@ include file="rodape.htm" %>

■ Page

- Importação de pacotes de classes Java utilizadas

<%@ page import="java.util.*,java.io.*" %>

- Informações de configuração para geração

<%@ page contentType="text/plain" %>

<%@ page language="java" %>

<%@ page session="true" %>

Exemplo com vários jsp

- Vide fonte Ex8

Index.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Exemplo Ex.8</title>
  </head>
  <body>
    <jsp:include page="Cabecalho.jsp" />
    <a href="FormContato">Solicite aqui o Contato</a>
  </body>
</html>
```

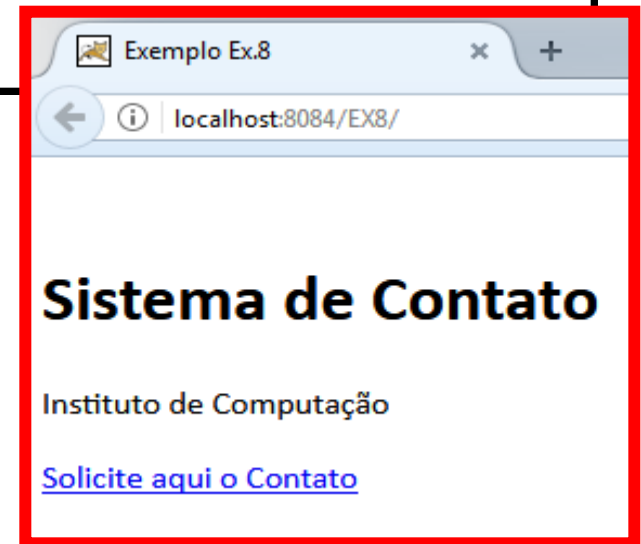


Inclui

Cabecalho.jsp



```
<br>
<h1>Sistema de Contato</h1>
<p>Instituto de Computação</p>
```



```
public class ProcessarContato extends HttpServlet {  
    @Override  
    protected void doGet(HttpServletRequest request, HttpServletResponse response)  
        throws ServletException, IOException {  
        RequestDispatcher rd = request.getRequestDispatcher("FormularioContato.jsp");  
        rd.forward(request, response);  
    }  
}
```

Servlet ProcessarContato

**Instância um request
dispatcher para o jsp**

**Usa o request dispatcher para pedir o Container que
inicialize o FormularioContato.jsp**

```

</title>Formulário Contato</title>
<script type="text/javascript">
    function validarInteiro(valor) {
        valor = parseInt(valor)
        if (isNaN(valor)) {
            return ""
        }else{
            return valor
        }
    }

function valida_enviar() {
    //valido o nome
    if (document.fvalida.nome.value.length==0) {
        alert("Tem que escrever seu nome")
        document.fvalida.nome.focus()
        return 0;
    }
}

```

The screenshot shows a web browser window with the title 'JSP Page'. The address bar displays 'localhost:8084/EX8/FormContato'. The page content includes a heading 'Sistema de Contato', a subheading 'Instituto de Computação', and a form with two text input fields labeled 'Nome:' and 'Idade:'. Below the fields is a button labeled 'Enviar'. An arrow points from the 'Enviar' button in the browser to the corresponding HTML code in the bottom block.

```

<jsp:include page="Cabecalho.jsp" />
<form method="post" action="http://localhost:8084/EX8/ProcessarContato">
<table>
<tr>
    <td>Nome: </td>
    <td><input type="text" name="nome" id="nome" size="30" maxlength="1" />
</tr>
<tr>
    <td>Idade: </td>
    <td><input type="text" name="idade" id="idade" size="3" maxlength="3" />
</tr>
</table>
<input type="button" value="Enviar" onclick="valida_enviar()" />
</form>
</body>

```

**FormularioContato.jsp
gerado**

Servlet ProcessarContato

@Override

```
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    // pegando os parâmetros do request
    String nome = request.getParameter("nome");
    int idade = Integer.parseInt(request.getParameter("idade"));
    if ((idade < 18) || (nome.isEmpty())) // verifica os dados
    { // retorna para o formulario de contato
        RequestDispatcher rd = request.getRequestDispatcher("FormularioContato.jsp");
        rd.forward(request, response);
    }
    else{
        request.setAttribute("nomerecebido", nome);
        request.setAttribute("idaderecebida", idade);
        RequestDispatcher rd = request.getRequestDispatcher("/RespostaContato.jsp");
        rd.forward(request, response);
    }
}
```

Dados inválidos
Retorna ao
Formulário



Dados Válidos => RespostaContato.jsp

Servlet ProcessarContato

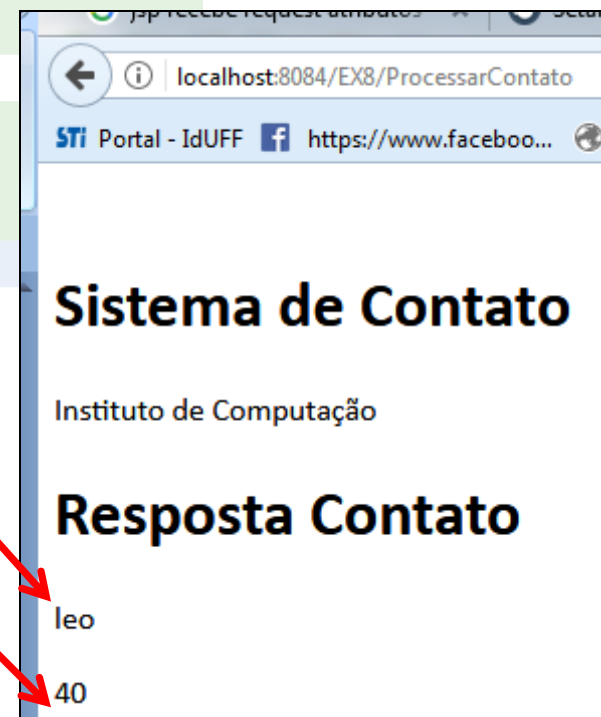
@Override

```
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    // pegando os parâmetros do request
    String nome = request.getParameter("nome");
    int idade = Integer.parseInt(request.getParameter("idade"));
    if ((idade < 18) || (nome.isEmpty())) // verifica os dados
    { // retorna para o formulario
        RequestDispatcher rd = request.getRequestDispatcher("/RespostaContato.jsp");
        rd.forward(request, response);
    }
    else{
        request.setAttribute("nomerecebido", nome);
        request.setAttribute("idaderecebida", idade);
        RequestDispatcher rd = request.getRequestDispatcher("/RespostaContato.jsp");
        rd.forward(request, response);
    }
}
```

Adiciona dois atributos ao objeto request para o JSP usar

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Resposta Contato</title>
</head>
<body>
<jsp:include page="Cabecalho.jsp" />
<h1>Resposta Contato</h1>
<%
    out.println(request.getAttribute("nomerecebido"));
%>
<p></p>
<%
    out.println(request.getAttribute("idaderecebida"));
%>
</body>
</html>
```

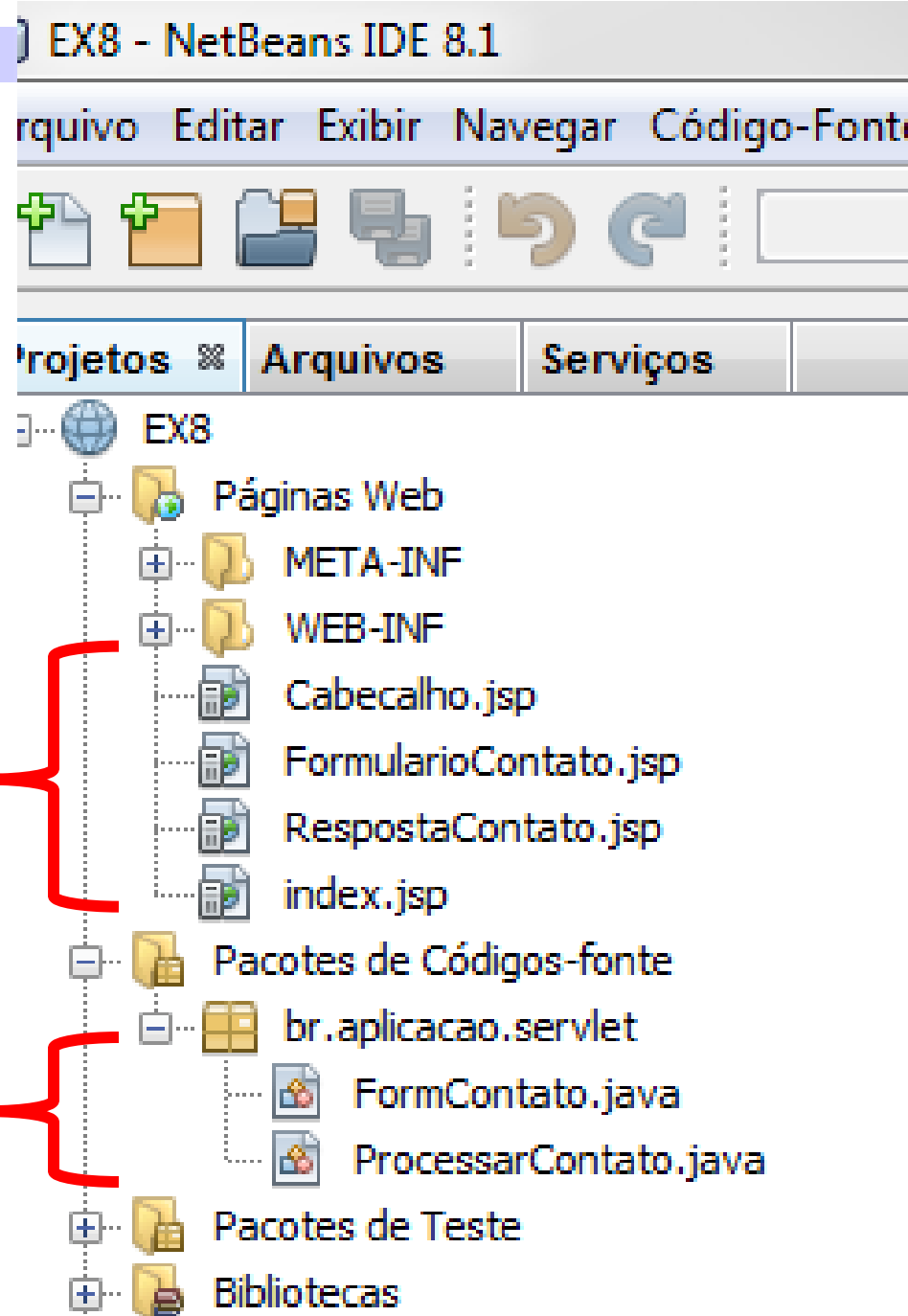
RespostaContato.jsp



Exemplo: Ex8

As páginas jsp

As servlets no pacote



Scriptlet Tag `<% ... %>`

- Expression Tag `<%= ... %>`
 - Servem para imprimir um valor na página
 - **NÃO deve haver ponto-e-vírgula!**

```
<html>
  <head>
    <meta http-equiv="Content-Type" content=
    <title>JSP Page</title>
  </head>
  <body>
    <h1>Olá Mundo!</h1>
    <p>
      <%= 2+5 %>
    </p>
  </body>
</html>
```

Scriptlet Tag <% ... %>

■ Expression Tag <%= ... %>

- Servem para imprimir um valor na página
- **NÃO deve haver ponto-e-vírgula!**


```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/"
    <title>JSP Page</title>
  </head>
  <body>
    <h1>Olá Mundo!</h1>
    <p>
      <%= new java.util.Date() %>
    </p>
  </body>
</html>
```



Action Tag <jsp: ... />

■ Transferindo a requisição para outra JSP

```
<jsp:forward page="outra pagina.jsp" />
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html;" />
    <title>Exemplo Ex.8</title>
  </head>
  <body>
    <jsp:include page="Cabecalho.jsp" />
    <a href="FormContato">Solicite aqui o Contato</a>
  </body>
</html>
```



Action Tag **<jsp: ... />**

- Transferindo a requisição para outra Servlet

```
<jsp:forward page="/OlaMundo" />
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html;
    <title>Exemplo Ex.8</title>
  </head>
  <body>
    <jsp:include page="Cabecalho.jsp" />
    <a href="FormContato">Solicite aqui o Contato</a>
  </body>
</html>
```

Action Tag <jsp: ... />

■ Incluindo um outro JSP

```
<jsp:include page="Cabecalho.jsp" />
<form method="post" action="http://localhost:8084/EX8/ProcessarContato"
<table>
<tr>
    <td>Nome: </td>
    <td><input type="text" name="nome" id="nome" size="30" maxlength="1
</tr>
<tr>
    <td>Idade: </td>
    <td><input type="text" name="idade" id="idade" size="3" maxlength=
</tr>
</table>
<input type="button" value="Enviar" onclick="valida_enviar()">
</form>
</body>
```



Declaration Tag `<%! ... %>`

- Declarar uma variável da classe

```
<%! int contador = 0; %>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html;"
    <title>Exemplo Ex.8</title>
  </head>
  <body>
    <jsp:include page="Cabecalho.jsp" />
    <%= contador %>
    <a href="FormContato">Solicite aqui o Contato</a>
  </body>
</html>
```

Directive Tag `<%@ ... %>`

■ Indicar características do JSP

- ☐ Codificação de Caracteres
- ☐ Bibliotecas usadas (imports)

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

```
<!DOCTYPE html>
```

```
<html>
```

```
    <head>
```

```
        <meta http-equiv="Content-Type" content="text/html;
```

```
        <title>Exemplo Ex.8</title>
```

```
    </head>
```

```
    <body>
```

```
        <jsp:include page="Cabecalho.jsp" />
```

```
        <a href="FormContato">Solicite aqui o Contato</a>
```

```
    </body>
```

```
</html>
```


Directive Tag `<%@ ... %>`

■ import

```
<%@page contentType="text/html" pageEncoding="UTF-8" import="java.util.*"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Exemplo Ex.8</title>
  </head>
  <body>
    <jsp:include page="Cabecalho.jsp" />
    <%= new Date() %>
    <a href="#">Contato</a> Solicite aqui o Contato</a>
  </body>
</html>
```



Directive Tag `<% @ ... %>`

- `<%@ include file="outra_pagina.jsp">`
 - Incluindo um outro JSP
- `<%@ include file="outra_pagina.jsp">`
- `<jsp:include page="Cabecalho.jsp" />`
- Fazem a mesma coisa?

Directive Tag `<%@ ... %>`

- `<%@ include file="outra_pagina.jsp">`
 - Usado para conteúdo estático
- `<jsp:include page="Cabecalho.jsp" />`
 - Usado para conteúdo dinâmico