

Sistemas Distribuídos e
Tolerância a Falhas
Engenharia Informática

JSP Java Server Pages

Bruno Ferreira | m4004
Duarte Gomes | e7333

Introdução

- É uma tecnologia de construção de scripts que correm do lado do servidor;
- É constituído por instruções baseadas na sintaxe do Java;
- Similar às tecnologias Active Server Pages (ASP) da Microsoft e PHP;
- Desenvolvida pela Sun Microsystem;

Jsp ou Java Server Pages é uma tecnologia utilizada no desenvolvimento de aplicações Web, desenvolvida pela Sun Microsystem. É uma Tecnologia similar às tecnologias Active Server Pages (ASP) da Microsoft e PHP. Estas tecnologias estão fundamentadas na arquitectura SSI (Server Side Includes) que são comandos extensivos à linguagem HTML, os quais podem conter conteúdo estático (HTML) e dinâmico (ASP, PHP, JSP, etc).

Introdução (cont.)

- Fundamentado na arquitectura SSI (Server Side Includes) que são comandos extensivos à linguagem HTML;
- Pode conter conteúdo estático (HTML) e dinâmico (ASP, PHP, JSP, entre outras);
- Os ficheiros com JSP têm a extensão .jsp;

Com JSP podemos criar aplicações Web que se executam em vários servidores Web, de múltiplas plataformas, já que Java é na sua essência uma linguagem multiplataforma. As páginas JSP estão compostas por código HTML/XML misturado com etiquetas especiais para programar scripts de servidor em sintaxe Java. Portanto, poderemos escrever JSP com o nosso editor HTML/XML habitual.

Páginas JSP

- Compostas por código HTML/XML;
- O código JSP insere-se nos próprios ficheiros HTML, estando as instruções misturadas com as marcas HTML;
- A identificação das zonas JSP e HTML é feita pelo interpretador Java, através de sequencias sintácticas;
- ```
<html>
<...>
<% ... Código JSP %>
<...>
</html>
```

## Directivas

- Directivas JSP afectam a estrutura geral do Servlet resultante da compilação de uma página JSP. Entre outras coisas, directivas podem ser usadas para definir a linguagem usada no documento JSP, arquivos a serem incluídos, bibliotecas de tags a serem usadas, etc.
- São mensagens dirigidas ao motor JSP.
- Normalmente não geram nenhum resultado visível.

## Atributos das Directivas

- o **Language** – Linguagem de scripting a ser utilizada (normalmente usa-se Java).
- o **Extends** – Classe que depois de compilada estende-se a outra.
- o **Import** – Lista de pacotes Java a importar.
- o **Session** – Define se a página está envolvida numa sessão HTTP (true ou false).
- o **ContentType** – Tipo MIME da resposta.

Os Scripts são excertos de código embutidos no código HTML do JSP. Os scripts numa página JSP são interpretados no lado do servidor, ao contrário do Java Scripts que é interpretado pelo browser (lado cliente).

## Exemplos de Directivas

- `<%@page contentType="Text/HTML" %>`
- `<%@page import="java.util.Date" %>`

## Objectos Implícitos

- Um objecto é uma entidade construída com base no modelo definido por uma classe, que consiste no modelo abstracto para representar dados.
- No JSP estão definidos alguns objectos implícitos, que podem ser utilizados pelos programadores para vários fins.

## Objectos Implícitos

- **Request** – representa um objecto HttpServletRequest, o pedido HTTP.
- **Response** – representa HttpServletResponse, a resposta HTTP.
- **PageContext** – Disponibiliza acesso aos atributos da página.
- **Application** – Representa um objecto ServletContext, o contexto da servlet.
- **Out** – Objecto jspWriter, de escrita.
- **Config** – ServletConfig, parâmetros de configuração.
- **Page** – Representa a própria JSP
- **Session** – Representa HttpSession, o objecto de sessão http.
- **Exception** – Objecto de erro.

O “ambiente” de uma página JSP disponibiliza 9 objectos instanciados.

## Formulários

- O tratamento de formulários é feito através da manipulação de parâmetros enviados na requisição (seja pelo método POST ou GET);
- Parâmetros ajudam nas decisões de processamento da página (controlo de login, password's etc.);

## Formulários

- Os parâmetros de um formulário são enviados na requisição HTTP, sendo assim, estarão acessíveis através do objecto implícito **REQUEST**.
- Parâmetros são acedidos através dos métodos:**
  - `request.getParameter("nome");`
  - `request.getParameterValues(String paramName);`

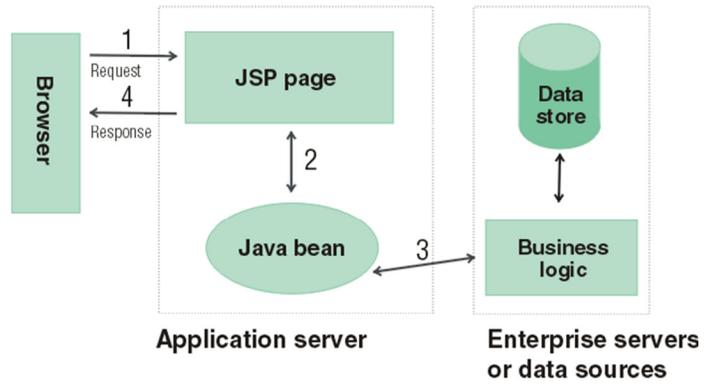
## Formulários

- EXEMPLO BÁSICO: (página que contém o form)

...

```
<form action="trataForm.jsp" method="post">
 Nome: <input name="nome" type="text">
 Sobrenome: <input type="text" name="sobreNome">
 <input type="submit" value="Enviar">
</form>
```

# Arquitectura JSP



## Alguns componentes importantes JSP

- `<%! int contador= 0 %>` - Métodos
- `<%=contador %>` - Resultado da variável *contador*
- `<%@ include file="ficheiro.jsp" %>` - Directivas
- `<%-- comentário --%>` - Comentários
- `out.println(k);` - Escrever no ecrã

### Declaração de expressão

`<%= %>` - O resultado é retornado como um String

Exemplo: O seu endereço IP é: `<%= request.getRemoteAddr() %>`

`<%= 2+2%>` // Será retornado o String "4"

### Declaração de sintaxe:

`<% %>` - Declaração livre.

Exemplo: `<% for( int i= 0; i< 10; i++){ out.println(" numero: "+ i); }%>`

## Exemplo simples JSP

```
<html>
 <head>
 <title>Exemplo JSP</title>
 </head>
 <body>
 <%@ include file="ficheiro.jsp" %>
 <h3>
 <% out.println("Hello World"); %>
 </h3>

 <%! int contador= 10; %>
 <%=contador ;

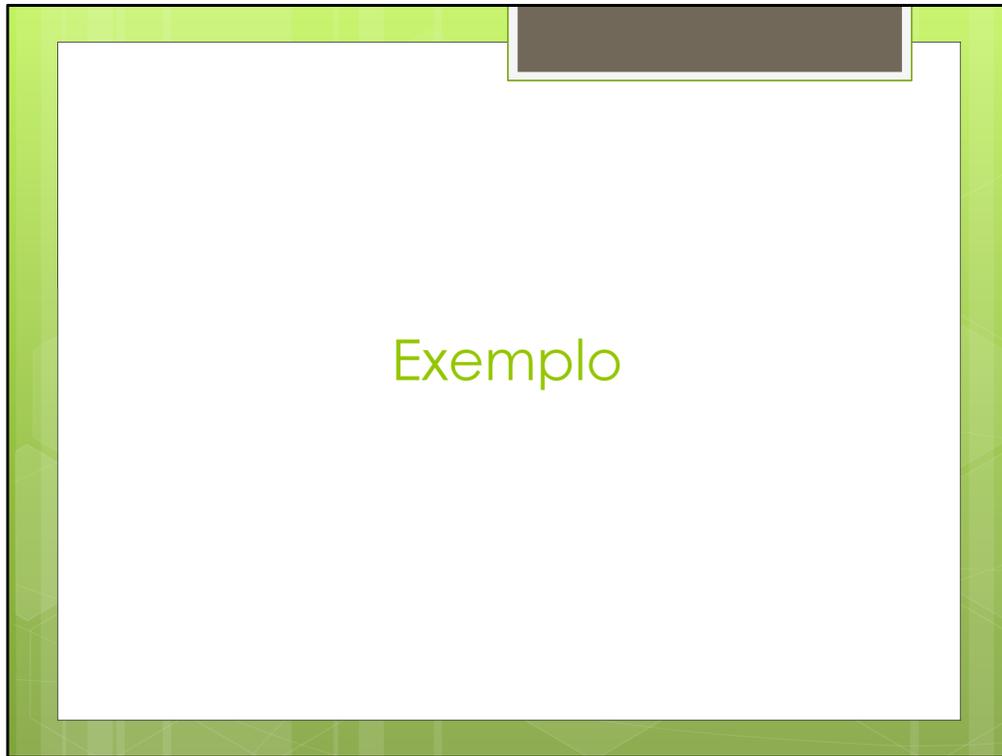
 out.println("
");

 Date d = new Date();
 String hoje = DateFormat.getDateInstance().format(d);
 %>
 Hoje é: <%=hoje%>
 </body>
</html>
```

## Desenvolvimento em JSP

- Servidor (Tomcat);
- Java Development Kit (JDK);
- Integrated Development Environment (IDE);





Explicar como adicionar o tomcat.

Como criar uma aplicação JSP.

Como adicionar um package, para poder aceder a métodos de outras classes.

Mostrar o nosso exemplo e explicar passo a passo.