

JSP (JavaServer Pages)

JavaServer Pages (JSP) é uma tecnologia que permite combinar, em uma só página, códigos Java, HTML estático, *Cascading Style Sheets* (CSS), *Extensible Markup Language* (XML) e JavaScript.

Páginas JSP são páginas HTML com códigos Java embutidos em *tags* especiais. Essas páginas devem ser salvas com extensão **.jsp**. Cada página JSP é traduzida para uma classe Servlet com extensão **.java**, compilada para um *bytecode* com extensão **.class** e executada em um servidor de aplicações web enviando ao cliente o conteúdo de resposta no formato HTML.

As principais vantagens oferecidas pela tecnologia JSP são as seguintes:

- Por ser uma especificação e não um produto, JSP permite que diversas empresas construam servidores web concorrentes, aumentando as possibilidades de escolha para os desenvolvedores que utilizam JSP;
- Páginas JSP são compiladas e não interpretadas, o que possibilita um desempenho e um processamento mais eficiente;
- Aplicações construídas com JSP são portáteis para qualquer plataforma;
- Permite desenvolver aplicações web que acessem bancos de dados, manipulem arquivos no formato texto, capturem informações a partir de formulários e captem informações sobre o visitante e sobre o servidor.
- JSP é uma parte integrante do conjunto de APIs J2EE e é compatível com todas as tecnologias incluídas.

É necessário que a página .jsp esteja hospedada em um servidor de aplicações web com suporte a container de Servlets para que a mesma seja executada apropriadamente.

Separar o design da página da lógica de programação – o que é um problema quando se utiliza a tecnologia Java Servlet – é mais fácil quando se utiliza JSP. Se o designer não quiser utilizar *Scriptlets* com código Java ele pode utilizar bibliotecas de tags especiais que embutem código Java na página por meio de tags em um formato semelhante ao HTML. Isso é mais fácil para os web designers fazerem, pois eles estão habituados a utilizar tags HTML. Uma das bibliotecas de tags mais conhecidas e utilizadas para essa finalidade é a *JavaServer Pages Standard Tag Library* (JSTL). Alguns *frameworks*, como o Struts e o *JavaServer Faces* (JSF) também oferecem bibliotecas de tags.

Normalmente, páginas JSP e Servlets são combinados no desenvolvimento de páginas web dinâmicas. Ambas as tecnologias são baseadas na plataforma Java, por isso oferecem independência de plataforma.

Tags (especiais) de JSP

- Tag de Scriptlet
 - Usada para escrever blocos de códigos Java em meio ao HTML;
 - `<% bloco de código %>`
- Tag de Expressão
 - Usada para mostrar (imprimir) o valor de uma variável ou objeto;
 - `<%= variavel %>`
- Tag de Declaração
 - Usada para se declarar variáveis, objetos e métodos globais;
 - `<%! Declaração de variável, objeto ou método %>`
- Tag de Diretiva
 - Usada especificar diretivas ao servidor de aplicações como a importação de bibliotecas ou inclusão de outras páginas .jsp;
 - `<%@ Diretiva %>`
 - Exemplos:
 - `<%@ page import="java.util.*" %>`
 - `<%@ include file="hello.jsp" %>`

Objetos Implícitos (ou Variáveis Predefinidas)

Ao carregarmos um página JSP, alguns objetos já são instanciados automaticamente com nomes predefinidos:

request

Objeto da classe `HttpServletRequest`, associado a requisição (GET ou POST)

response

Objeto da classe `HttpServletResponse`, associado a resposta ao cliente

out

Objeto da classe `JspWriter`, é um escritor utilizado para enviar saída ao cliente

session

Objeto da classe `HttpSession`, associado a requisição

application

Objeto da classe `ServletContext`, associado a contexto do servlet

Usando seu pacote de classes em uma página JSP

Ao criar uma página JSP para usar seu pacote de classes, você deve importar a(s) classe(s), ou o pacote como um todo, fazendo uso da tag diretiva page.

Importando uma única classe de um pacote:

```
<%@ page import="meuspacotes.sistemasdebancos.ContaCorrente" %>
```

Importando todas as classes de um pacote:

```
<%@ page import="meuspacotes.sistemasdebancos.*" %>
```

Importando mais de um pacote:

```
<%@ page import="meuspacotes.sistemasdebancos.*,java.util.*" %>
```

```
<%@ page language="java"
    contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <title>Exemplo de uma página JSP</title>
  </head>
  <body>
    <p>
      Oi, a data e hora de agora é <%= new java.util.Date() %>
    </p>
  </body>
</html>
```

Perceba que no meio do código HTML existe um código em Java delimitado por <%= e %> respectivamente.

Figura 1: Exemplo de uma página JSP que usa a tag de expressão

```
<%@ page language="java"
    contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">

<html>
  <head>
    <title>Exemplo de uma página JSP</title>
  </head>
  <body>
    <%
      // Este é um Scriptlet
      System.out.println("Imprime no console do Tomcat");
      java.util.Date date = new java.util.Date();
    %>
    <p>Oi, a data e hora de agora é <%= date %></p>
  </body>
</html>
```

Bloco de código Java, delimitado pelas tags `<%` e `%>`, chamado de **Scriptlet**

Referência ao objeto declarado no **Scriptlet**

Figura 2: Exemplo de uma página JSP que usa tags de scriptlet e expressão