



# **(✓) Curso Técnico**

Quem Investe no futuro faz QI

## **Informática para Internet**

**Desenvolvimento de Sistemas**

**Web III**

**Unidade I**

## A linguagem Java

Ao decorrer da disciplina a linguagem Java será utilizada, portanto é importante entender, mesmo que resumidamente, o que é a linguagem Java.

Em 1995 a Sun Microsystems criou a linguagem Java que é hodiernamente mantida pela Oracle. Uma linguagem amplamente utilizada para o desenvolvimento de sites e aplicativos, utilizando o paradigma da programação orientada a objetos, detendo as seguintes características:

- Orientação a objetos - classes, métodos, atributos e objetos poderão ser utilizados no desenvolvimento de software.
- Sistema multiplataforma - A JVM permite a execução do software desenvolvido em diferentes sistemas operacionais.
- Tipada - especifica os tipos de dados das variáveis utilizadas nos algoritmos.
- Compilada - o código de um programa escrito em Java é compilado para uma forma intermediária de código denominada bytecode, que é interpretada pela JVM, isto é, do arquivo.java é gerado arquivo.class .
- Interpretada - os bytecodes são interpretados pelo interpretador Java, o qual pode operar por si mesmo ou como parte de um navegador Web.
- Imperativa - o algoritmo desenvolvido determina como o programa irá se comportar ou funcionar.

## A linguagem Java é multiplataforma

Significa que um software/sistema/aplicação desenvolvida em java poderá ser compilada e interpretada em vários sistemas operacionais, pois é utilizada uma máquina virtual Java - JVM - Java Virtual Machine - para executar os software/sistema/aplicação desenvolvida em Java.

## Os arquivos Java

Para produzir um algoritmo Java é necessário utilizar um arquivo da extensão .java, logo é comum dizer que o arquivo de extensão .java detém o código fonte da aplicação Java.

Após produzir e compilar o arquivo da extensão .java, um arquivo .class é criado, viabilizando a interpretação pela JVM - Java Virtual Machine.

Para permitir o agrupamento de arquivos necessários para execução de um software/sistema/aplicação Java, a extensão .JAR poderá ser utilizada, pois a sigla JAR significa Java ARchive e é o formato utilizado para distribuir aplicações e bibliotecas na plataforma Java. É comumente entendido como executável Java ou arquivo compactador de outros arquivos.

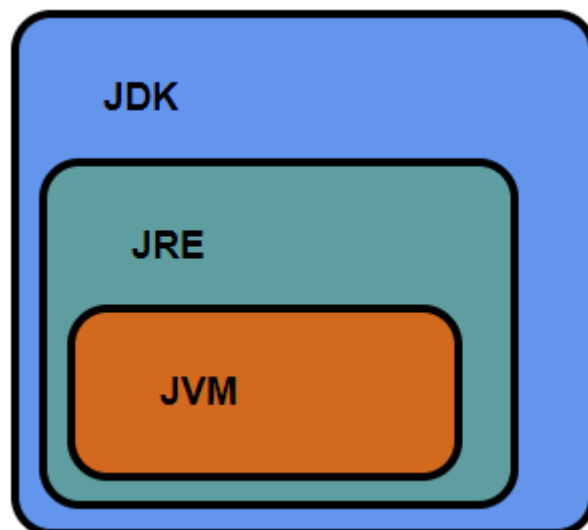
A extensão .war permite, via de regra, a compactação de todos os arquivos necessários para o funcionamento de um software/sistema/aplicação, mesmo que não sejam originalmente Java, na mesma linha, mas para aplicação Java EE a extensão .ear permitirá a compactação dos arquivos de um software/sistema/aplicação.

## Os Jotas Java

JDK ou Java Development Kit é o kit oficial de desenvolvimento para a linguagem Java. É um conjunto de programas que engloba compilador, interpretador, APIs e utilitários. Fornece ferramentas básicas para o desenvolvimento de aplicações Java. Vale ressaltar que não é possível utilizar recursos essenciais sem o JDK, como o comando javac para compilar ou java para interpretar ou executar o código. Ainda é importante registrar que a escolha do JDK é feita com base na versão ou edição da tecnologia Java, como JEE, JSE ou JME. Por fim, todo JDK possui um JRE.

Para executar qualquer código java o Java Runtime Environment ou JRE, será necessário, pois é uma camada de software que é executado sobre um sistema operacional, como Windows, Linux ou Mac, fornecendo as bibliotecas de classes e outros recursos necessários a execução do código.

Significativo salientar que os três componentes inter-relacionados para desenvolvimento, criação e execução de software/sistema/aplicação Java são: JDK, JRE e JVM.



Para o desenvolvimento de aplicativos de desktop ou até mesmo escrever scripts usando ferramentas para testes de automação o JSE Java Standard Edition é o mais indicado.

Para o desenvolvimento de aplicações, inclusive web, corporativas ou de média/alta complexidade o JEE Java Enterprise Edition é o mais indicado.

Para o desenvolvimento de aplicações para dispositivos móveis e sistemas embarcados o JME Java Micro Edition é o mais indicado.

## **Servlets**

Os servlets são programas Java do lado do servidor que usam a API do Sun Microsystems Java Servlet e suas classes e seus métodos associados, como definido na Especificação do Sun Microsystems Java Servlet 2.3. Esses programas Java estendem a funcionalidade de um servidor da web gerando o conteúdo dinâmico e respondendo às solicitações do Web client. Quando o navegador envia um pedido para o servidor, o servidor pode enviar as informações do pedido para um servlet, para que o servlet possa construir a resposta que é enviada de volta para o navegador.

Assim como applets são executados em um navegador da Web e estendem os recursos do navegador, Servlets são executados em um servidor da Web habilitado para Java e estendem os recursos do servidor. Devido à sua flexibilidade e escalabilidade, geralmente, mas não exclusivamente, os Servlets são utilizados para permitir que as empresas conectem-se a bancos de dados à Web.

Não obstante possa um programa servlet ser completamente independente, é viável organizar o desenvolvimento do aplicativo em duas partes:

- A lógica de negócios ou geração de conteúdo, que permite o controle do relacionamento entre a entrada, o processamento e a saída;
- A lógica de apresentação do conteúdo ou regras usabilidade e design gráfico, que determina como as informações são apresentadas ao usuário.

Pode-se utilizar um paradigma para viabilizar a manipulação da lógica de negócios por Javabeans, a lógica de apresentação por JavaServer Pages (JSP) ou arquivos HTML, e o protocolo HTTP manipulado por um servlet.

## **JSP - JavaServer Page**

É uma tecnologia que utiliza a linguagem de programação Java para permitir a criação de páginas dinâmicas de sistemas da Web. JSP é construído sobre a especificação Java Servlet. As duas tecnologias normalmente trabalham juntas, especialmente em aplicativos Web Java mais antigos.

A JSP facilita a criação de páginas que interagem perfeitamente com servlets Java, como o contêiner de servlet Tomcat. A JSP está presente em aplicativos da Java Web mais antigos e, sendo útil na criação de páginas Java Web simples e dinâmicas.

Uma página JSP é um documento com dados estáticos, que podem ser expressos em qualquer formato baseado em texto como HTML e XML, e elementos JSP ou Java, que possibilitam o conteúdo dinâmico, ou seja, tem forma de uma página HTML com trechos de código Java embutidos, visto que em geral a parte dinâmica da página é gerada pelo código Java.

Os documentos JSP, são via de regra da extensão .jsp, observando que quando há uma composição hierárquica de páginas para o alcance de um propósito, de modo que um arquivo superior inclua outros arquivos que contenham uma página JSP completa ou um fragmento de uma página JSP, a extensão recomendada para o arquivo de origem de um fragmento de uma página JSP é .jspf.

Ainda vale lembrar que quando você implementa um arquivo JSP em um servidor da Web que fornece um mecanismo de servlet, ele é pré-processado em um servlet executado no servidor da Web, como Tomcat. Importante atentar que no JavaScript o processamento ocorre no lado do cliente, especificamente num navegador que interpretará o código abrigado dentro das tags <script>. Por isso se infere que uma página JSP é mais adequada para tarefas propícias à execução no servidor, como conexão ao banco de dados.

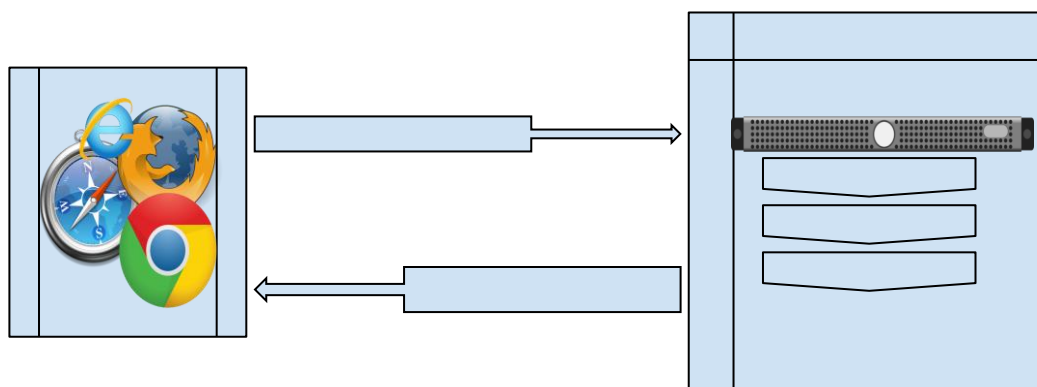
Uma página JSP é traduzida em um servlet Java antes de ser executada e processa solicitações HTTP e gera respostas de maneira semelhante a qualquer outro servlet. A tecnologia JSP facilitou a codificação servlet. A tradução geralmente ocorre na medida da necessidade, i.e., à medida que o aplicativo é executado. O conversor JSP normalmente é acionado pela extensão .jsp do nome do arquivo em uma URL.

As páginas JSP são totalmente interoperáveis com servlets, visto que uma página JSP ou uma página com tecnologia JSP pode incluir saída de um servlet ou encaminhar para um servlet, e um servlet pode incluir saída de uma página JSP ou uma página com tecnologia JSP ou encaminhar para uma página JSP ou uma página com tecnologia JSP.

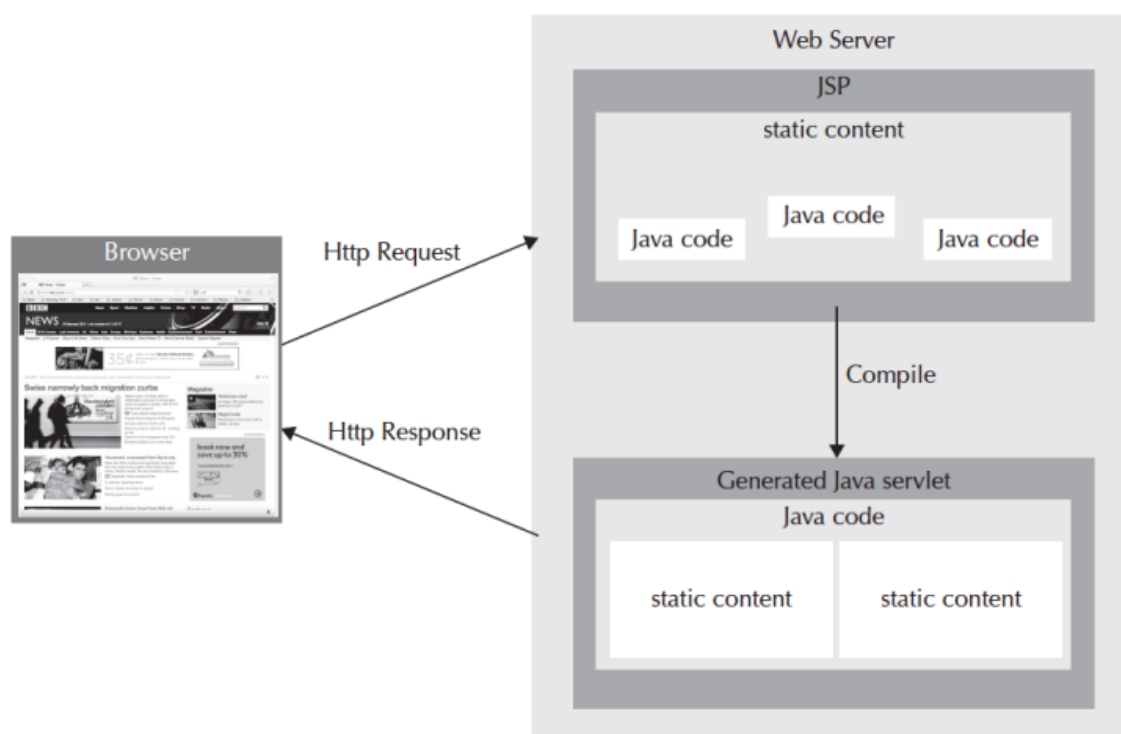
O fluxo de tempo de execução de uma página JSP ou uma página com tecnologia JSP perpassa pelas seguintes etapas:

1. Navegador faz a requisição da página JSP para servidor Web
2. Servidor Web encaminha requisição para container Servlet/JSP
3. Container Servlet/JSP traduz a requisição ou página JSP
4. Container Servlet/JSP compila a requisição ou página JSP
5. Container Servlet/JSP instancia e executa bytecode Servlet
6. Container Servlet/JSP envia a resposta da requisição para o navegador

O fluxo de tempo de execução de uma página JSP ou uma página com tecnologia JSP também pode ser observado na imagem a seguir:



O fluxo de tempo de execução de uma página JSP ou uma página com tecnologia JSP também poderá ser observado na imagem a seguir:



Fonte: Fonte: livro Java EE 7: The Big Picture

## Servlet versus JSP

Percebe que a tecnologia JSP é utilizada na criação de sistemas web, podendo ser pensada como uma extensão da tecnologia Servlet, pois fornece mais

funcionalidades do que a tecnologia servlet, como linguagem de expressão, JavaServer Pages Standard Tag Library - JSTL **3**.

Depreende-se que a tecnologia JSP utiliza tags HTML e tags JSP, portanto as páginas JSP são mais fáceis de manter do que as construídas com Servlet, pois pode-se separar design e desenvolvimento.

A servlet evolui para JSP trazendo os seguintes recursos adicionais:

- a) Pode-se considerar a tecnologia JSP como uma extensão da tecnologia Servlet, pois todos os recursos do Servlet funcionam em JSP, não obstante, pode-se utilizar objetos implícitos, tags pré-definidas, linguagem de expressão e tags personalizadas em JSP, o que facilita o desenvolvimento de JSP.
- b) Ainda pode-se considerar que a tecnologia JSP é mais fácil de manter, pois pode ser gerenciado de uma forma mais fácil, permitindo a separação facilitada da lógica de negócios da lógica de apresentação, o que não acontece na tecnologia Servlet, visto que mistura-se a lógica de negócios com a lógica de apresentação.
- c) Há ganho de velocidade, pois não há necessidade de recompilação para as situações em que há modificação da página(aparência) JSP, diferentemente do código Servlet que precisa de recompilação e atualização para os casos de alteração, mesmo que seja apenas da aparência do sistema.
- d) A possibilidade de usos de tags de ação, JSTL, tags personalizadas, objetos implícitos e EL -Expression Language, permitem a redução do código ou alcance do mesmo objetivo com menos código.

## **Páginas JSP e o ciclo de vida**

No processo de produção de um algoritmo ou sistema é interessante que o desenvolvedor entenda as etapas ou fases que definem o ciclo de vida do documento algorítmico produzido, o que se pode nominar como ciclo de vida. No JavaServer Page as etapas são:

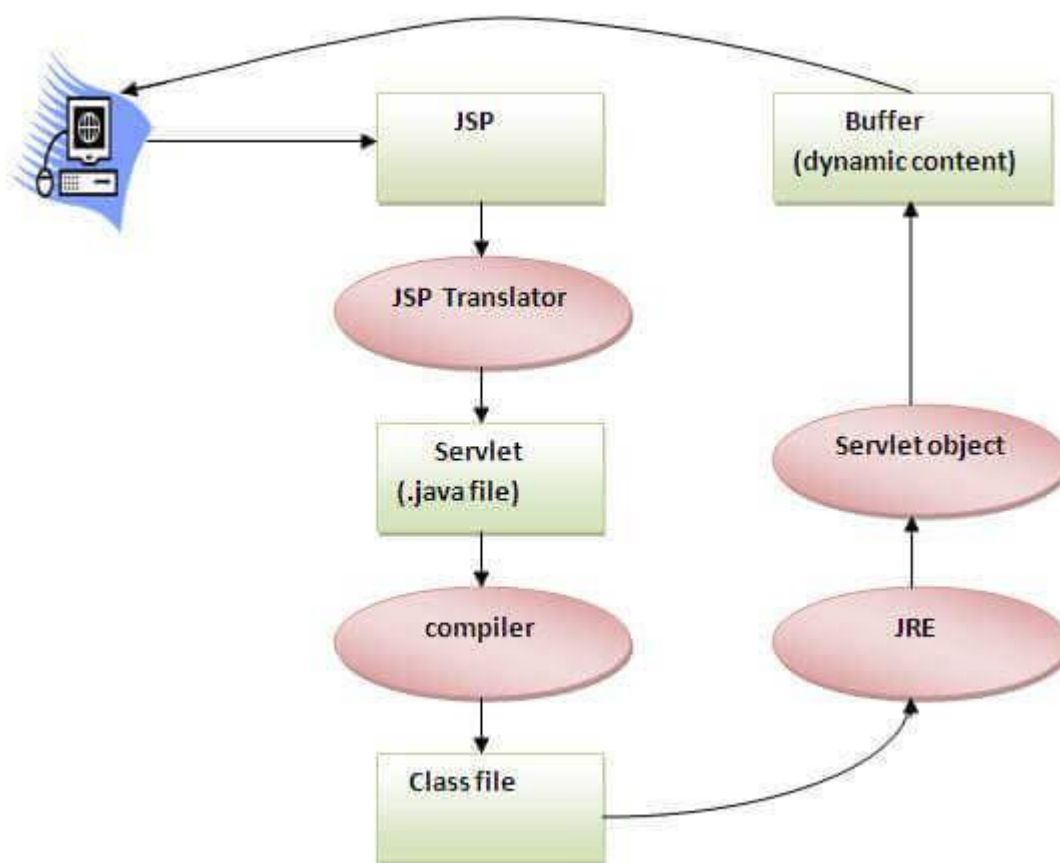
- Tradução da página JSP
- Compilação da página JSP



## CURSO TÉCNICO EM INFORMÁTICA PARA INTERNET

- Carregamento de classe (o carregador de classe carrega o arquivo de classe)
- Instanciação (Objeto do Servlet Gerado é criado).
- Inicialização (o contêiner invoca o método `jspInit()`).
- Processamento de solicitação (o contêiner invoca o método `_jspService()`).
- Destroy (o contêiner invoca o método `jspDestroy()`).

A imagem abaixo diagrama as etapas listadas:



Ao analisar a imagem acima, percebe-se que a página JSP é convertida em Servlet com a ajuda do tradutor JSP, lembrando que o tradutor JSP é uma parte do servidor web (como Tomcat com JEE) que é responsável por traduzir a página JSP em Servlet. Em seguida, a página Servlet é compilada e convertida no arquivo java. Ainda vale mencionar que os processos que acontecem no Servlet como inicialização, confirmação de resposta ao navegador e destruição, são executados no JSP posteriormente.

## A necessidade de HTML no JSP

A tecnologia HTML é fundamental para a construção de software web, pois pode-se dizer que os documentos que trafegam na web são estruturados pela HTML. Dito isso vale entender que HTML. é acrônimo de HyperText Markup Language ou Linguagem de Marcação de HiperTexto, pois na Web não há só texto, mas músicas, fotos, vídeos e outros elementos que necessitam de marcação para eficiência e bom funcionamento da Web.

Quando menciona-se a necessidade de uso de HTML nos sistemas Java Web refere-se a necessidade de marcação e apresentação do conteúdo nos navegadores, por isso entender a mecânica de funcionamento do HTML é importante.

A estrutura básica de um documento HTML é bastante simples, como mostrado:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta charset="UTF-8">
```

```
<title>Título do Documento</title>
```

```
</head>
```

```
<body>
```

```
Conteúdo do documento...
```

```
</body>
```

```
</html>
```

## DOCTYPE

O **<!DOCTYPE>** é uma declaração que auxilia o navegador a mostrar a página corretamente. Como há vários documentos diferentes na Web, o navegador só

consegue apresentar uma página HTML de maneira correta se souber o tipo e a versão utilizada.

## Tag ou seção **head**

O elemento **<head>** é um agrupador para todos os elementos de cabeçalho. Os elementos dentro de **<head>** podem incluir scripts, links para folhas de estilos, meta-informações da página, entre outros, salientando-se que os elementos dentro da seção **<head>** não são mostrados na página, eles apenas descrevem alguma característica do documento ou fazem referência a arquivos externos.

## Tag ou seção **body**

Diferentemente da **<head>**, a seção definida pelo elemento **</body>** corresponde a todo o conteúdo visível da página (corpo), ou seja, os demais elementos que compõem a página como os parágrafos, links, listas, imagens, tabelas, etc devem ficar dentro da seção **</body>**.

## Tags

As tags HTML são palavras-chave escritas entre os sinais de **< >**. Elas definem um elemento e, normalmente, vêm em pares como, por exemplo, **<p>** e **</p>**.

A primeira tag é a inicial e a segunda é a final. A tag final é escrita como a inicial, com uma barra (/) antes do nome da tag. Elas também podem ser chamadas de tag de abertura e tag de fechamento, respectivamente. Exemplo: **<p>**Parágrafo**</p>**

Há, ainda, algumas tags que não possuem tag de fechamento, ou seja, elas são chamadas de tags autocontidas, ou elementos vazios. É o caso dos elementos **<br>** e **<input>**, por exemplo: **<br>** e **<input type="text" name="cidade">**

## Atributos

Os atributos fornecem informações adicionais sobre os elementos HTML e eles são sempre especificados na tag inicial.

# CURSO TÉCNICO EM INFORMÁTICA PARA INTERNET

As tags possuem alguns **atributos** genéricos e outros específicos. Os atributos são escritos dentro da tag, separados por espaço e seu valor sempre deve estar entre aspas.

Exemplo:

```
<p class="big">Parágrafo grande</p>
```

ou

```
<input type="text" name="nomecompleto">
```

No exemplo acima, temos um elemento `<p>` com o atributo `class` de valor `big`, e um elemento `<input>` com um atributo `type` de valor `text` e outro atributo `name` de valor `nomecompleto`.

## Scriptlet JavaServer Pages - JSP

A leitura deste material permite inferir que uma página JSP pode ser considerada um arquivo baseado em HTML, cuja extensão é `.jsp`, sendo que dentro de uma página ou arquivo `.jsp` podemos escrever além do código de marcação HTML, código Java.

Para escrever o código java dentro de um arquivo com extensão `.jsp` deve-se utilizar a seguinte sintaxe:

```
<% código java %>
```

Numa estrutura HTML ficará:

```
<html>
```

```
<head>
```

```
<meta charset="UTF-8">
```

```
<title>Título do Documento</title>
```

```
</head>
```

```
<body>
```

```
<% código java %>
```

```
</body>
```

```
</html>
```

Vale registrar que scriptlet é o código escrito entre `<% código java %>` que é nominado como **scriptlet**, pois a palavra **script** (conjunto de instruções algorítmicas) com o sufixo **let**, que indica algo pequeno.

Na tecnologia JSP as tags de expressão ou *expression tag* são utilizadas para “exibir” um conteúdo ou dado num elemento de saída(navegador).

A forma de uso ou sintaxe das tag de expressão ou *expression tag* JSP é a seguinte:

```
<%= expressão do dado ou valor %>
```

Documento com a utilização de tag de declaração JSP:

```
<html>
```

```
<head>
```

```
<meta charset="UTF-8">
```

```
<title>Título do Documento</title>
```

```
</head>
```

```
<body>
```

```
<%= “Olá mundo!” %>
```

```
Hora atual:<%= java.util.Calendar.getInstance().getTime() %>
```

```
</body>
```

```
</html>
```

No exemplo acima as tags de expressão são utilizadas duas vezes, sendo que na primeira vez, uma frase do tipo String será exibida, e na segunda a classe Calendar do pacote util é utilizada para exibir a hora atual extraída pelo método getTime() do método getInstance().

Na tecnologia JSP também existem as tags de declaração ou *declaration tag* que são utilizadas para declaração de campos e métodos. Insta consignar que o código escrito dentro da tag de declaração ou *declaration tag* JSP é disponibilizado fora do método `service()` do servlet gerado automaticamente, destarte, não obtém memória a cada solicitação.

A forma de uso ou sintaxe das tag de declaração JSP é a seguinte:

`<%! declaração de campo ou método %>`

Documento com a utilização de tag de declaração ou *declaration tag* JSP:

`<html>`

`<head>`

`<meta charset="UTF-8">`

`<title>Título do Documento</title>`

`</head>`

`<body>`

`<%! int variavelValor=200; %>`

`<%= "O valor da variável é:" + variavelValor %>`

`</body>`

`</html>`

No exemplo acima é declarada a variável `variavelValor` que é inicializada com o valor `200`.

## Referências

- ★ <https://www.javatpoint.com/jsp-declaration-tag>
- ★ <https://www.alura.com.br/apostila-java-web/jaserver-pages>
- ★ <https://www.geeksforgeeks.org/jsp-declaration-tag/>
- ★ <https://www.javatpoint.com/jsp-expression-tag>
- ★ <https://www.javatpoint.com/jsp-tutorial>
- ★ [https://docs.oracle.com/cd/A97688\\_16/generic.903/a97681/jspprim.htm](https://docs.oracle.com/cd/A97688_16/generic.903/a97681/jspprim.htm)
- ★ <http://docente.ifrn.edu.br/fellipealeixo/disciplinas/tads-2012/desenvolvimento-de-sistemas-web/material/antigo/jsp01.pdf>
- ★ <https://www.devmedia.com.br/introducao-ao-java-server-pages-jsp/25602>