

# **DESENVOLVIMENTO DE SISTEMAS WEB**

## **TECNOLOGIA JSP**

# Olá!

Ao final desta aula, você será capaz de:

1. Conhecer os princípios básicos do desenvolvimento para ambiente web, com a tecnologia JSP. 2. Ter uma vivência prática com a criação de aplicativos Web, baseados nesta tecnologia.

Nesta aula, abordaremos a tecnologia JSP, descrevendo suas características e benefícios. Também, serão apresentados: a estrutura da página JSP, o conceito de diretivas (com a configuração de parâmetros) e include (com a inclusão de classes de outros pacotes Java).

## 1 Fundamentos da tecnologia JSP

**JSP** (Java Server Pages) é uma tecnologia web-scripting para desenvolvimento de aplicações Web, similar às tecnologias **Active Server Pages** (ASP) da Microsoft® e PHP.

Faz parte da família de tecnologias Java. Páginas JSP são compiladas em servlets e podem chamar componentes JavaBeans (**beans**) ou componentes Enterprise JavaBeans (**enterprise beans**), para a realização do processamento no servidor. Pode-se afirmar que a tecnologia JSP é uma componente chave na arquitetura de alto escala para aplicações baseadas na Web, provendo um modo simplificado e dinâmico de gerar páginas Web que contêm um código dinamicamente gerado.

### Características Gerais:

Sendo baseada na plataforma Java (uma das linguagens mais populares atualmente), seu código escrito em uma determinada arquitetura pode ser portado para qualquer outra. Além de ter uma enorme comunidade de desenvolvedores, ampla documentação e diversas bibliotecas e códigos prontos, dos quais o desenvolvedor pode usufruir.

Permite ao desenvolvedor Web produzir aplicações que acessem banco de dados, manipulem arquivos no formato texto, capturem informações, a partir de formulários e captem informações sobre o visitante e sobre o servidor.

Facilmente codificado, facilitando a elaboração e manutenção da aplicação Web.

Permite a separação da programação lógica (parte dinâmica) da programação visual (parte estática).

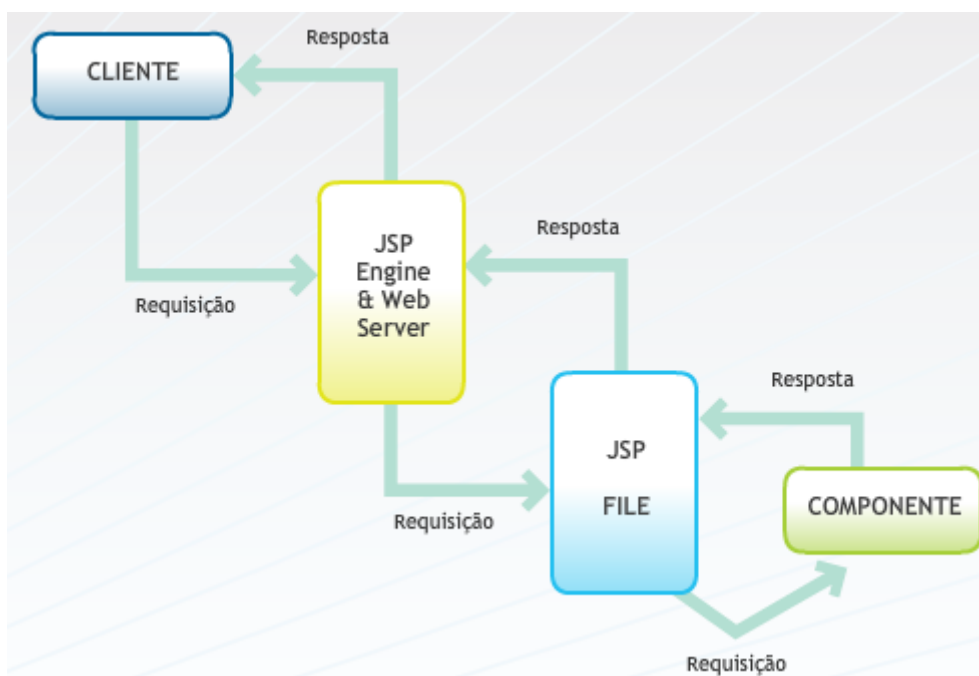
Torna possível uma distribuição de carga de processamento de aplicações desenvolvidas em diversos servidores, sendo que servidores podem ser adicionados ou removidos de maneira a acompanhar o aumento ou decréscimo dessa carga de processamento (escalabilidade).

## 2 Arquitetura JSP

Página JSP é uma página construída por um desenvolvedor *Web* que inclui a tecnologia JSP e *tags* específicas, combinadas com outras *tags* estáticas (HTML ou XML). Uma página JSP tem a extensão **.jps** ou **.jspx**; o quê informa ao servidor *Web* que um **engine** (motor) irá processar e codificar os elementos presentes nessa página. Utilizam *tags* XML e *scriptlets* escritos na linguagem Java para encapsular a lógica que gera o conteúdo para a página. Isso envia qualquer *tag* de formatação (HTML ou XML) de volta à página de resposta. Assim, páginas JSP separam a lógica da página de seu design e de sua exibição.

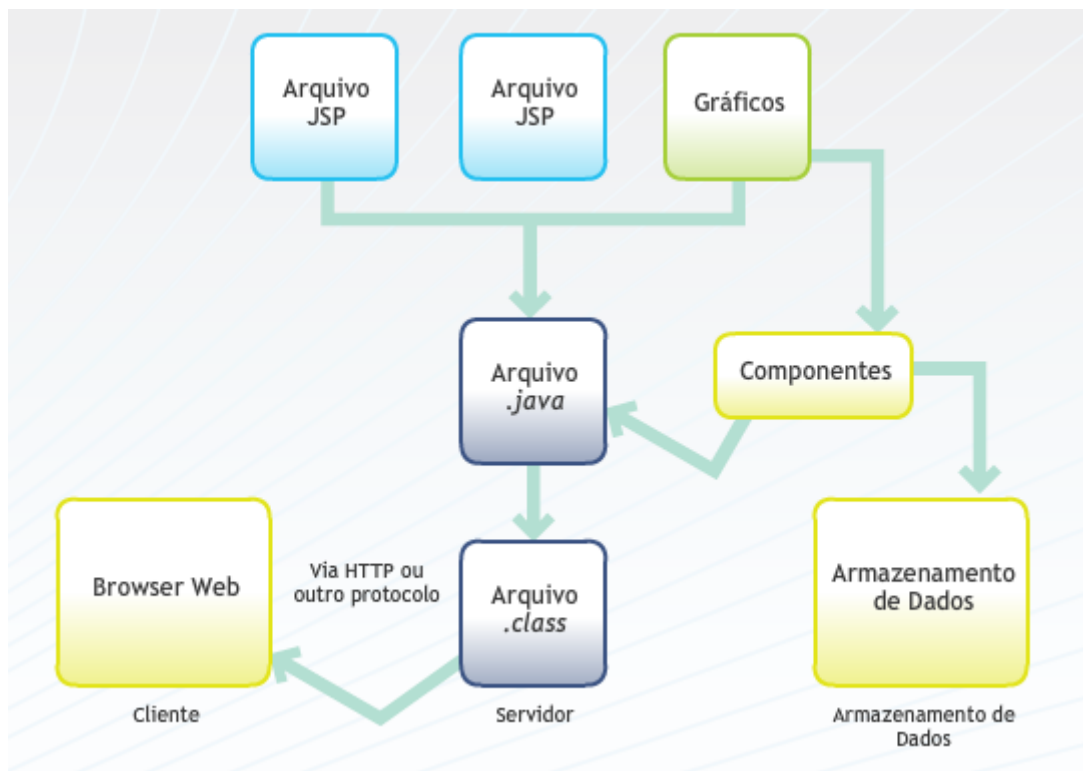
Saiba Mais

Para mais informações, [clique aqui](#)



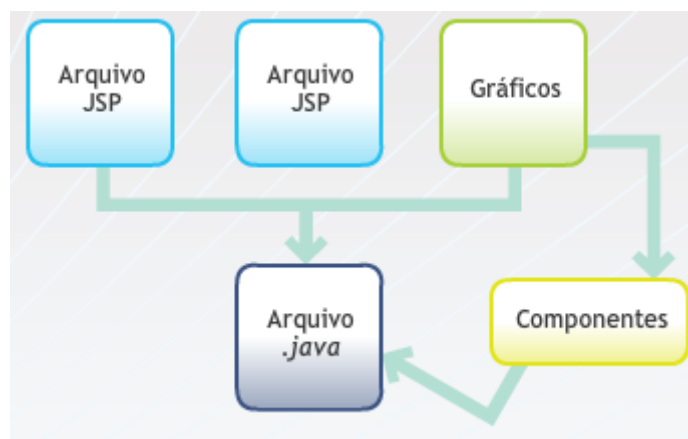
Fonte: Arquitetura JSP – Modelo Básico. (Fonte da imagem: SANTOS, R.C.; JORGE, E.M.F. Java Server Pages – JSP).

Uma página JSP é compilada quando o usuário a carrega em *Web browser*. O processo de compilação é ilustrado na figura a seguir.



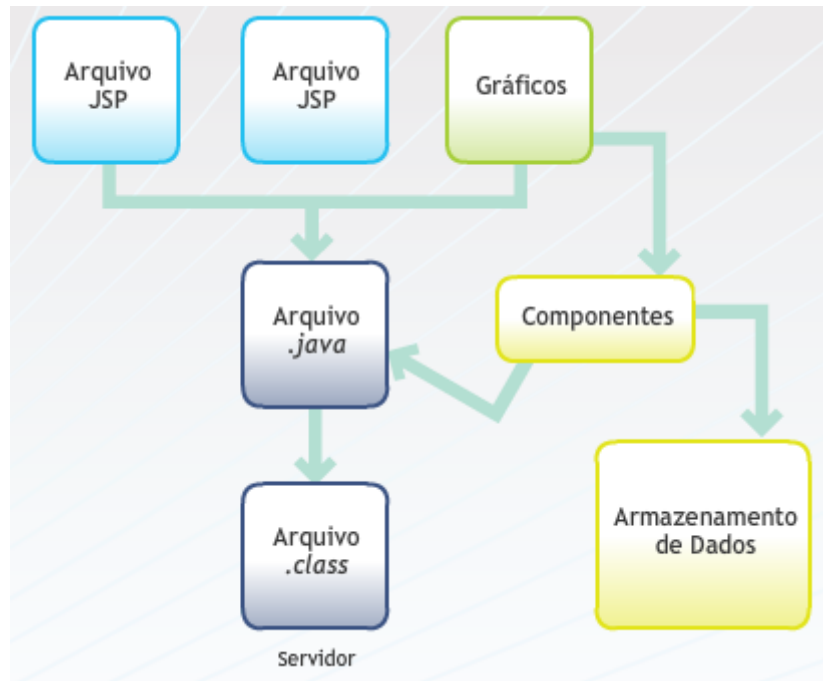
Fonte: Processo de Compilação de uma Página JSP. (Fonte da imagem)

## 2.1 Passo a passo para a compilação de uma página JSP

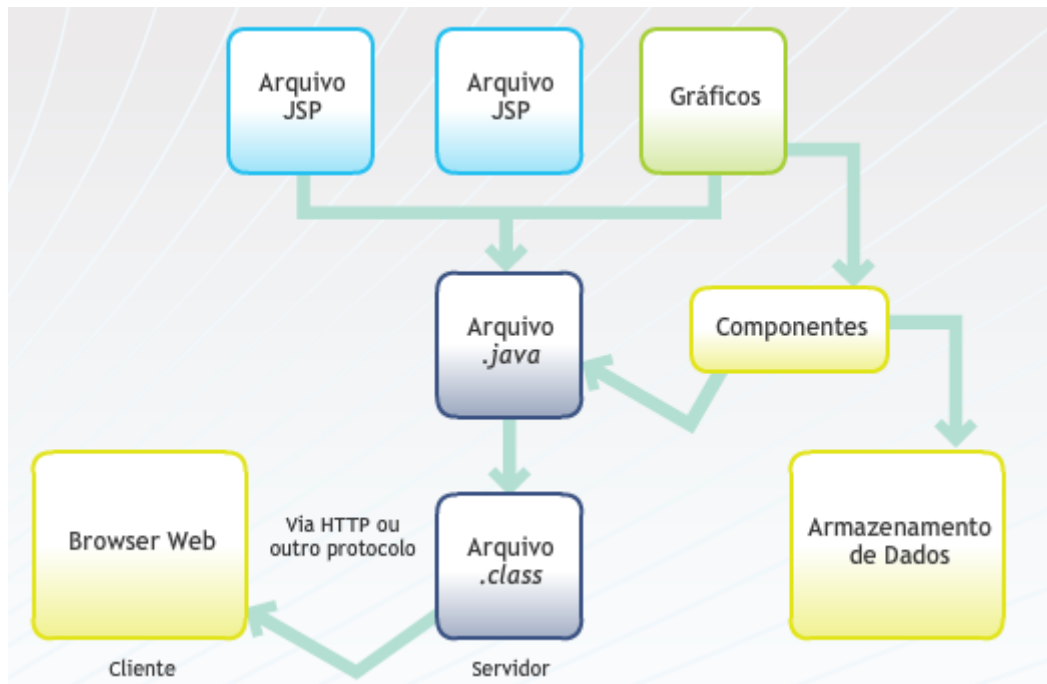


Pode-se afirmar que uma aplicação JSP, normalmente, é uma coleção de arquivos JSP, arquivos HTML, gráficos e outras fontes. Então, quando o usuário carrega a página pela primeira vez, os arquivos (os quais compõem a aplicação) são traduzidos (sem qualquer dado dinâmico), para um único arquivo fonte Java (**.java**), com o nome definido pela sua aplicação JSP.

Posteriormente, o arquivo **.java** é compilado em um arquivo **.class**. Usualmente, o arquivo **.java** é um servlet Java, que está de acordo com a API Java Servlet. Essa fase é conhecida como "tempo de tradução" (*translation time*).

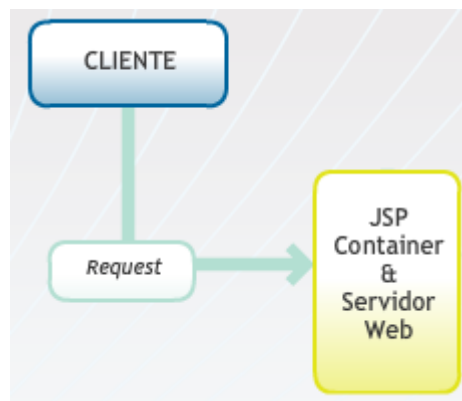


Quando um usuário realiza uma solicitação (*request*) à aplicação JSP, um ou mais de um dos componentes da aplicação (*bean, enterprise bean ou servlet*) manipula (m) os dados que o usuário submeteu; ou recupera (m) os dados dinamicamente de um repositório de dados e retorna (m) os mesmos para um arquivo **.java**. Esse último arquivo, por sua vez, é recompilado em um arquivo **.class**.



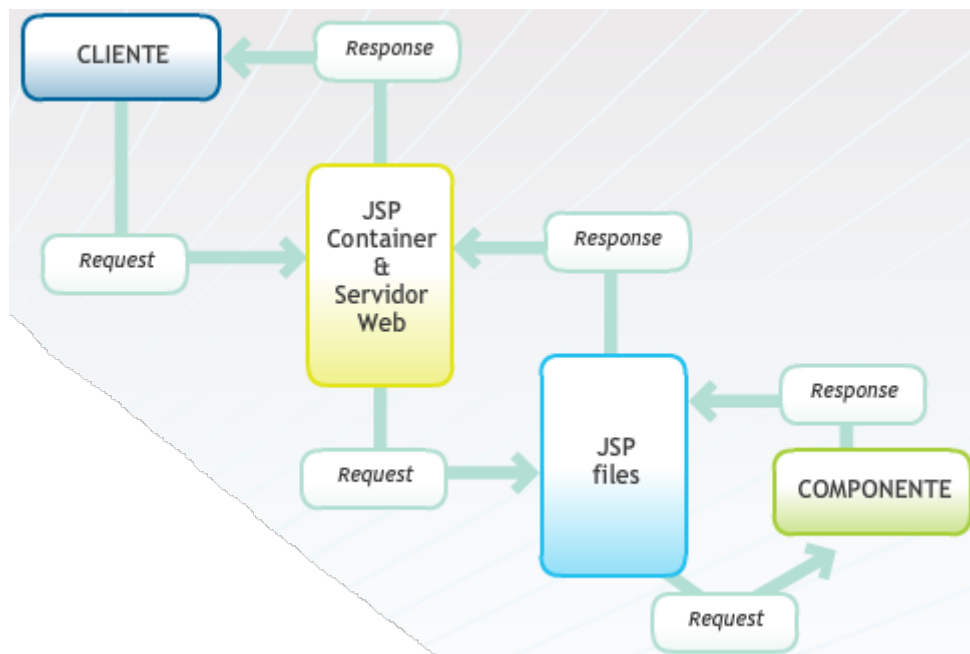
O arquivo **.class**, sendo um *servlet* Java, retorna os dados para o *Web browser* do cliente através do seu método *service*.

Quando o usuário faz uma nova solicitação, o componente obtém ou manipula os dados novamente e os retorna para o arquivo **.java**. Esse último arquivo, mais uma vez é compilado em um arquivo **.class**. Essa etapa é conhecida como "tempo de solicitação" (*request time*).



Fonte: Processo de Compilação de uma Página JSP II. (Fonte da imagem)

As informações que um usuário envia, por exemplo, através de um formulário HTML são armazenadas em um objeto *request*, que as envia do cliente para um *container* JSP.



Fonte: Processo de Compilação de uma Página JSP II. (Fonte da imagem)

O *container* JSP envia o objeto **request** para o componente (*beans, enterprise bean ou servlet*) especificado pelo arquivo JSP.

O componente manipula o **request**, possivelmente recuperando dados de um banco de dados ou de outro tipo de repositório, e, então, envia um objeto **response** de volta para o *container* JSP.

## 2.2 Elementos básicos do JSP

Páginas JSP podem ser configuradas por vários tipos de elementos que aperfeiçoam a apresentação dos dados para o cliente, assim como decrescem o labor da manutenção. Veremos agora os elementos básicos da tecnologia JSP:

### Diretiva

DIRETIVA PAGE: Permite informar ao servidor, em tempo de tradução, algumas propriedades específicas da página, quando esta estiver disponível. Pode-se afirmar que a forma pela qual esse processo é realizado é por meio dos atributos da diretiva. tais como:

- **CONTENTTYPE**: Define o tipo de conteúdo MIME para a resposta da página JSP. O valor padrão é "`\ text / html`", isto é, se este atributo não for definido, o tipo de conteúdo de resposta será este. EXEMPLO: `<% @ page contentType = `` xml `` %>`
- **IMPORT**: Possibilita a realização da importação de pacotes para que seja possível o acesso às classes em uma página JSP. EXEMPLO: `<% @ page import = `` br.estacio.exatas.hipermedia.beans. *, Java.util. * `` %>` <sup>1</sup>
- **SESSION**: Definir os objetos de uma sessão que podem ser acessados pela página JSP. O valor padrão é verdadeiro. EXEMPLO: `<% @ page session = `` true `` %>`

<sup>1</sup> No exemplo, dois pacotes estão sendo importados: **br.estacio.exatas.hipermedia.beans** e **java.util**. Uma página JSP pode ter várias diretivas **page** com o atributo de **import**. Entretanto, é possível a realização do **import** de todos os pacotes em uma única diretiva separando-os através de uma vírgula como no exemplo.

- **ERRORPAGE**: Define o destino de qualquer exceção lançada na página JSP que está sendo executada. EXEMPLO: `<% @ page errorPage = `` errors / error.jsp ``%>`
- **ISELIGNORE**: Define se uma página JSP suporta ou não o uso de linguagens de expressão (Expression Language - **EL** <sup>2</sup>). O valor padrão é falso. EXEMPLO: `<% @ page isELIgnored = `` true ``%>`
- **isErrorPage**: define se uma página JSP pode expor uma exceção enviada para ela através do atributo **errorPage**. O valor padrão é falso. EXEMPLO: `<% @ page isErrorPage = `` true ``%>`

<sup>2</sup> Linguagem de Expressão (**EL- Expression Language**): tem por objetivo facilitar o acesso a objetos que seguem as regras de um componente **JavaBean** e seus atributos, por meio de uma sintaxe própria e fácil de ser entendida e usar. Assim, o desenvolvedor não precisa, obrigatoriamente, saber a sintaxe da linguagem Java. É possível desabilitar uma definição do atributo **isELIgnored** da *página* diretiva como **true**.

**DIRETIVA INCLUDE**: Em tempo de tradução, essa diretiva é utilizada para incluir um código ou texto oriundo de outra página JSP, como, por exemplo, uma barra de navegação comum a várias páginas de uma aplicação. O único atributo dessa diretiva é o arquivo qualificado como informações do arquivo que será incluído. EXEMPLO: `<% @ include file = `` menuBarNaveg.jsp ``%>`

## Scriptlets

**SCRIPTLET**S: São trechos de códigos Java que podem ser inseridos em qualquer lugar da página e servem, entre outras coisas, para definir valores para as variáveis a serem usadas nas expressões. Seguem a delimitação de caracteres especiais. Dentre os principais *scriptlets*, podem-se citar: comentários, declarações e expressões.

**COMENTÁRIOS**: Podem ser de 2 (dois) tipos: Comentários HTML: `<!-- comentario HTML -->` Comentários JSP: `<% -- comentario JSP --%>`

## DECLARAÇÕES:

Permite a definição de variáveis e métodos em qualquer local de uma página JSP.

Há 2 (dois) tipos: Locais: `<% int y = 27; %>` Globais: `<%! int y = 27; %>`

**EXPRESSÕES**: Utilizadas combinando tags `<%` e `%>` com o símbolo `=`, tornando possível a escrita, por exemplo, de uma saída para a página JSP como no exemplo abaixo:

`<% = y%>`

## Atenção

As *tags* utilizadas para diretivas, *scriptlet* e expressões são muito semelhantes. Tome cuidado ao utilizá-las!

## Saiba mais





Para essa aula, sugerimos que acesse:

Saiba mais sobre JSP:

GONÇALVES, E. Desenvolvendo Aplicações Web com JSP, Servlets, JavaServer Faces, Hibernate, EJB 3 Persistence e Ajax. Rio de Janeiro: Ciência Moderna, 2007.

SIEIRA, K.; BASHAM, B. Use a Cabeça! Servlets & JSP. Rio de Janeiro: Alta Books, 2008.

## O que vem na próxima aula

Na próxima aula, você vai estudar:

- Middleware JDBC: fundamentos, banco de dados em Java, uso do NetBeans para fora de banco JavaDB.

## CONCLUSÃO

Nesta aula, você:

- Conheceu a tecnologia jsp: seus benefícios, sua arquitetura.
- Aprendeu o mecanismo para o desenvolvimento de soluções cliente-servidor web-based.