

VÍDEO 4.1: Utilizando a JSP em um projeto web

Olá, seja bem-vindo(a)!

Avançando mais um pouco nos conhecimentos sobre ferramentas e tecnologias existentes para o desenvolvimento em Java Web, nesta videoaula, você irá estudar sobre a JSP (Java Server Pages). Esta tecnologia proporcionou maior flexibilidade aos programadores e grandes avanços no quesito dinamismo e interações com o usuário.

A tecnologia JSP foi lançada pela empresa SUN na década de 1990. Era uma grande empresa do ramo de tecnologia e foi, posteriormente, comprada pela ORACLE. As Java Server Pages são páginas web baseadas em HTML, em que você pode escrever código Java. Essas páginas são interpretadas pelo servidor de aplicações e possuem o seu código convertido 100% em HTML no momento em que ocorre uma requisição para essa página.

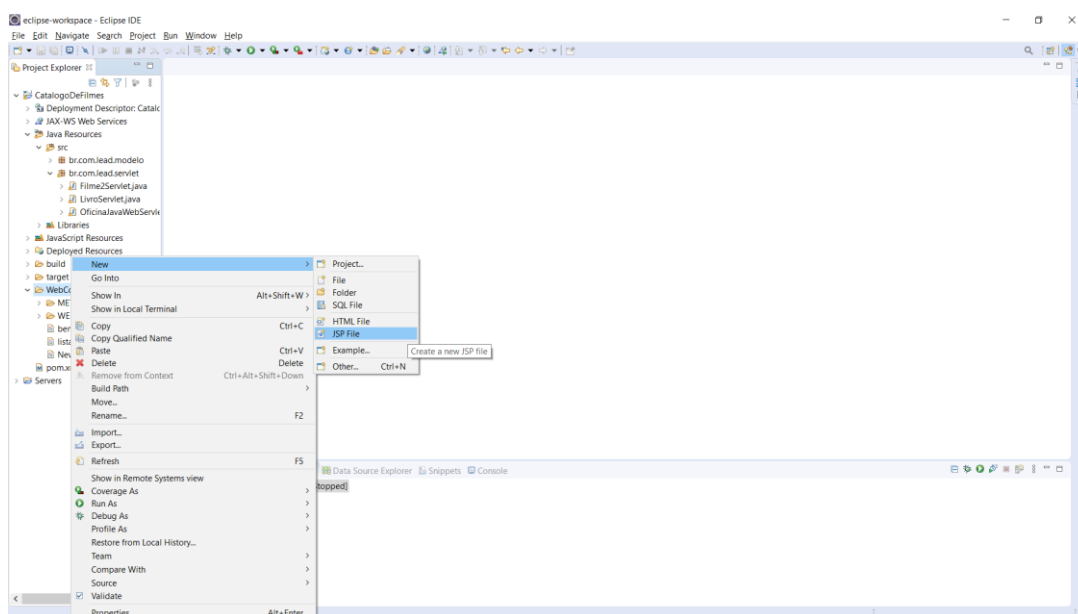
Diferentemente da classe Servlet, que possui o seu código compilado previamente antes de ser executado, a JSP não necessita de uma classe Servlet para ser processada, pois a Java Server Pages pode receber requisição diretamente do servidor de aplicação, e, assim, o conteúdo da Java Server Pages é processado. Uma outra característica da JSP é que essa tecnologia roda no lado do servidor e não no lado do cliente.

Os arquivos referentes a JSP ficam localizados no mesmo local que as páginas HTML do seu projeto, ou seja, no diretório WebContent da sua aplicação. Para isso, você deve criar os arquivos da JSP utilizando a extensão .jsp. Como dito anteriormente, na JSP, você conseguirá escrever código Java, e, para isso, deve-se utilizar os Scriptlets, que são pequenos trechos de código Java que devem ser escritos entre as tags<% e %>. Além disso, possuem esse nome, pois se comportam como verdadeiros scripts. Dentro dos Scriptlets, você pode acessar os objetos de classe, os atributos, inserir blocos de decisão como if e else, inserir blocos de loop

como for e while e ainda utilizar as demais operações como se fosse um arquivo Java.

Uma das vantagens do uso da JSP é que você tem acesso a um objeto chamado out, do tipo JspWriter, sem a necessidade de declarar o objeto. É com ele que você consegue imprimir dados na sua página, por exemplo, utilizando o seguinte trecho de código dentro de um Scriptlet: `out.println("Olá.");` será exibido a informação Olá.

Agora que você já sabe da existência da JSP, é necessário praticar. Para isso, você irá criar a sua primeira página JSP. Acesse o programa Eclipse no seu computador. Depois que abri-lo, navegue até a pasta WebContent, e, com o menu de opções aberto, navegue em New para escolher a opção JSP File. Em seguida, informe o nome do seu arquivo. Para isso, use o seguinte nome `hellojsp.jsp`.



#Audiodescricao: A imagem mostra a interface do Eclipse IDE. No topo, temos a barra de ferramentas. Abaixo, à esquerda, temos a barra “Project Explorer”. Nela, temos vários arquivos. Dentre eles, a pasta “WebContent” que está expandida e selecionada. Em destaque, há um menu aberto a partir dessa pasta em que a opção “New” está selecionada. Por sua vez, a opção “New” gerou um menu lateral em que a opção “JSP File” está selecionada.

Com o arquivo criado, através do Eclipse, você irá perceber que o arquivo .jsp é praticamente um arquivo HTML, com todas as tags do HTML previamente preparadas. Contudo, possui a extensão diferente.



```
hellojsp.jsp
1 <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2   pageEncoding="ISO-8859-1"%>
3 <!DOCTYPE html>
4 <html>
5 <head>
6   <meta charset="ISO-8859-1">
7   <title>Insert title here</title>
8 </head>
9 <body>
10
11 </body>
12 </html>
```

#Audiodescricao: A imagem mostra o código:

```
"<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
  pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here </title>
</head>
<body>

</body>
</html>".
```

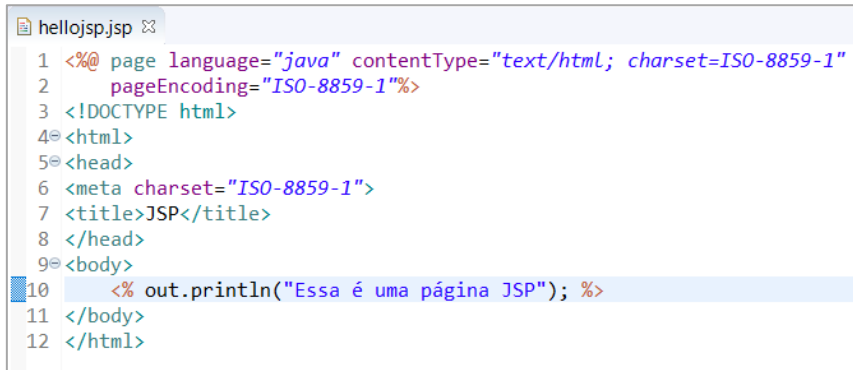
Feito isso, o próximo passo será a edição desse arquivo. Localize a tag title, que, como em um arquivo HTML, geralmente, fica na parte superior do arquivo; delete o seu conteúdo através do teclado; e escreva apenas JSP. Na tag body, você irá imprimir a informação Essa é uma página JSP. Para isso, você utilizará o Scriptlet

Centro de Pesquisa, Desenvolvimento e Inovação Dell

Telefone:(85)3492-1062 | www.leadfortaleza.com.br

Av. Santos Dumont, 2456 - 1906 | 60150162 - Fortaleza. CE

com o código Java. Fique atento(a), quando você for inserir o código Java, deve-se colocá-lo entre as tags `<%` e `%>`. E para imprimir alguma informação de dentro de um trecho de Scriptlet, você irá utilizar o objeto `out`. O trecho deve ficar assim `<% out.println("Essa é uma página JSP."); %>`.

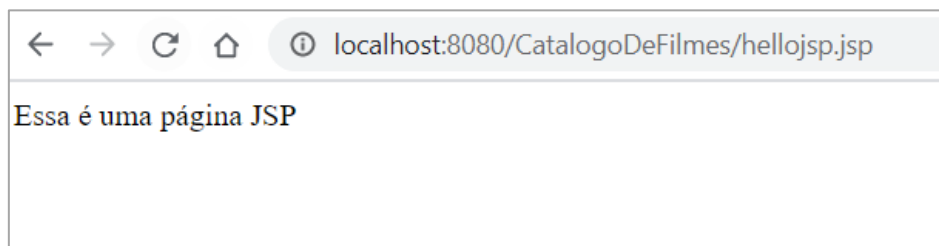


```
hellojsp.jsp
1 <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2   pageEncoding="ISO-8859-1"%>
3 <!DOCTYPE html>
4 <html>
5 <head>
6 <meta charset="ISO-8859-1">
7 <title>JSP</title>
8 </head>
9 <body>
10 <% out.println("Essa é uma página JSP"); %>
11 </body>
12 </html>
```

#Audiodescricao: A imagem mostra o código:

```
"<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
  pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>JSP</title>
</head>
<body>
<% out.println("Essa é uma página JSP"); %>
</body>
</html>".
```

Para verificar como sua aplicação será exibida na web, você deve acessar o endereço <http://localhost:8080/CatalogoDeFilmes/hellojsp.jsp> no seu navegador. A tela a ser exibida deve conter a seguinte informação: Essa é uma página JSP. Caso não apareça, deve ser verificado se a URL foi digitada corretamente e se o arquivo está nomeado como `hellojsp.jsp` corretamente.



#Audiodescricao: a imagem mostra um recorte de um navegador de internet. Nele, temos a barra de ferramentas, com as opções “Voltar”, “Avançar”, “Recarregar” e “Página inicial”, além da barra de endereço, com o seguinte endereço URL: “localhost:8080/CatalogoDeFilmes/ hellojsp.jsp”. Logo abaixo, temos o texto “Essa é uma página JSP”.

Com os passos realizados anteriormente, você deve ter percebido que a página foi exibida sem a necessidade de passar por uma classe Servlet. Isso implica que é possível implementar tudo o que foi implementado, usando a sua Servlet, só que agora utilizando a página JSP.

Entretanto, ainda assim, você continuaria com um dos problemas já existentes anteriormente, que era aquela grande mistura de código Java com o HTML, e é nisso que você trabalhará aqui a fim de minimizá-los. O melhor cenário que você poderia ter era a classe Servlet possuir apenas código Java, o que é necessário para suas regras de negócio e tratamento de fluxo da aplicação, enquanto que a página JSP se preocuparia apenas em exibir informações para o usuário, capturar dados e interagir com quem está utilizando a aplicação. Dessa forma, a página JSP teria a menor quantidade de código Java possível, podendo, assim, ter uma pessoa especialista em designer para trabalhar nessas páginas, pois, da maneira que está o designer, também necessitaria entender de Java para poder manipular a página. Porém, conhecer Java não é uma obrigação do designer.

Ter uma camada responsável apenas na apresentação dos dados e outra camada apenas em definir regras e fluxos da aplicação é um dos princípios de um padrão de

projeto bastante conhecido no desenvolvimento de aplicações, que é o padrão Model View Controller, o MVC. Em que o Model é representado pelas classes que possuem a responsabilidade de armazenar e buscar dados do banco de dados; a View é constituída pelos elementos responsáveis pela apresentação da aplicação, sendo, neste caso, representada pelas páginas JSP; e o Controller geralmente constituído pelos elementos responsáveis por redirecionar as requisições e ter algumas regras de negócio, sendo representado pelas classes Servlet.

A próxima etapa em seus estudos sobre a JSP é aplicar essa tecnologia ao seu projeto, aquele que vem sendo trabalhado desde a videoaula da aula 1. Contudo, isso será realizado no próximo vídeo.

Até lá!