

INTRODUÇÃO AO JSP

PROF. ME. HÉLIO
ESPERIDIÃO

CLIENT-SIDE

O cliente-side é uma aplicação processada e executa no navegador do cliente que é o responsável por interagir com o Servidor HTTP.

Entre as tecnologias cliente-side temos o HTML que é executado no navegador, o CSS é outra tecnologia cliente que serve para formatar paginas HTML, há ainda o javascript que permite desenvolver ou ampliar o poder ao lado do cliente.

SERVER-SIDE

Server-side por representa tecnologias em que os processos são interpretados/processados diretamente no servidor, retornando como resultado a codificação client-side.

BENEFÍCIOS DO JSP

Custo.

- O JSP não tem custo de licença

JSP é embutido no HTML

- Assim como no php

Pode utilizar todo o poder da linguagem de programação JAVA.

JAVA SERVER PAGES

JSP significa Java Server Pages, uma linguagem criada pela SUN (Atualmente faz parte da empresa ORACLE) gratuita.

É baseada em script com objetivo primário a geração de conteúdo dinâmico para páginas da Internet.

Dentro de um arquivo .JSP é possível escrever código Java.

JSP

Permite adicionar comportamento dinâmico em nossas páginas, como declaração de variáveis, condicionais (if), loops (for, while) entre outros.

É possível instanciar classes e utilizar todo o poder da linguagem JAVA.

Scriptlet

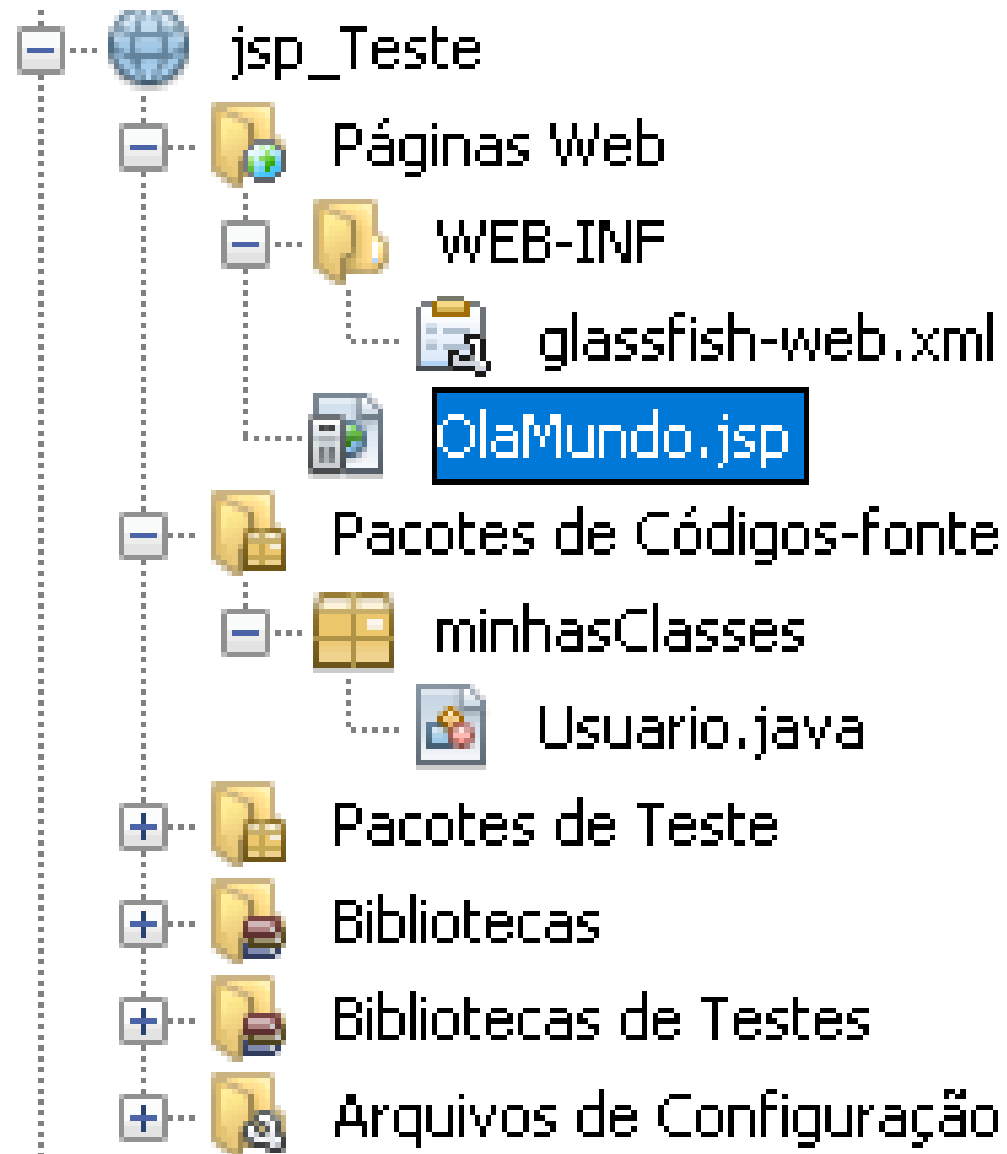
Scriptlet é o código escrito entre `<% e %>` .

Esse nome é composto da palavra script (pedaço de código em linguagem de script) e let significa algo pequeno.

Crie um projeto JAVA WEB

Crie um novo projeto JAVA web.

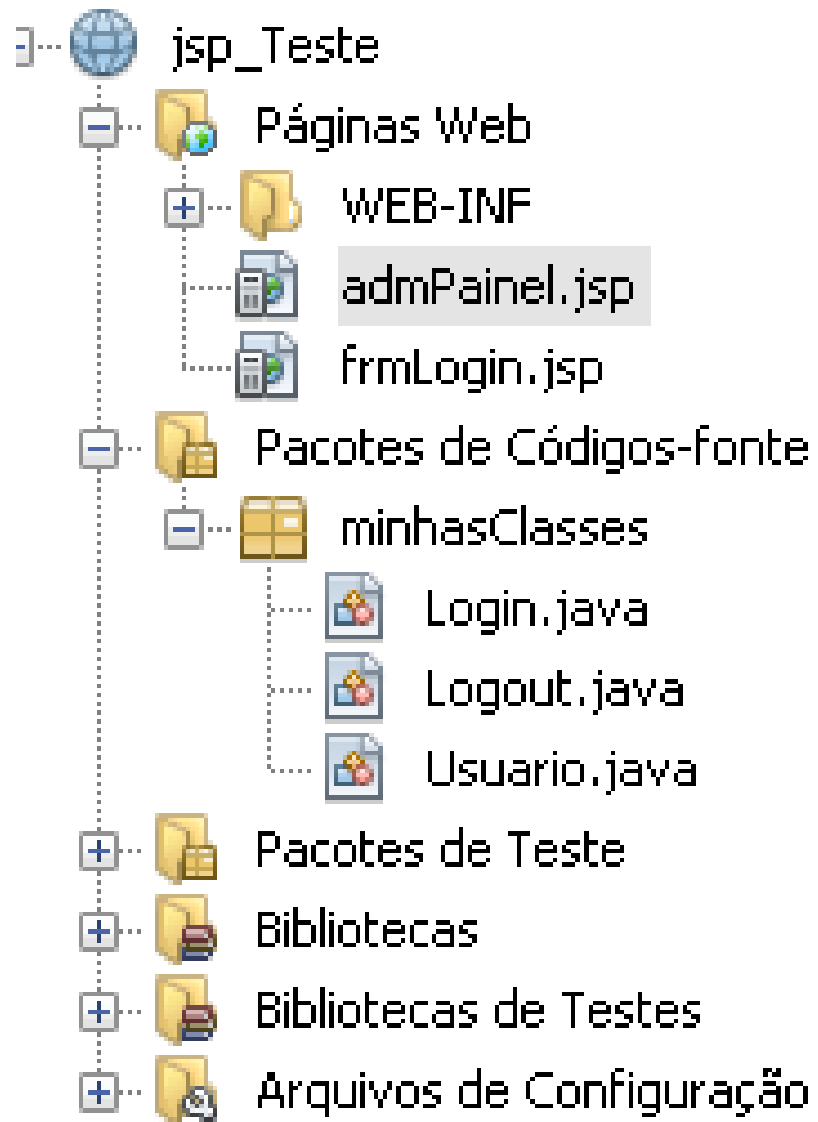
Adicione uma página do tipo jsp ao projeto.




```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
  </head>
  <body>
    <h1> <% out.println("Hello World!");%></h1>
  </body>
</html>
```

Olá mundo
página OlaMundo.jsp

O código é delimitado por <% %>



Integrando Servlets e JSP

A estrutura de arquivos e classes ao lado vai permitir que exista

Uma integração entre servlets e jsp por meio de variáveis de Sessão.

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
  </head>
  <body>
    <form action="/Login" method="post">
      Nome<input type="text" name="txtNome"><br>
      Senha<input type="password" name="txtSenha"><br>
      <input type="submit" value="login">
    </form>
  </body>
```

Login.jsp

PÁGINA JSP QUE ENVIA DADOS DE LOGIN PARA UM SERVLET /LOGIN

Servlet /Login

Verifica se o login está

Correto

Cria uma sessão

Redireciona para a página

admPainel.jsp

```
import javax.servlet.http.HttpSession;

/**
 *
 * @author Helio
 */
public class Login extends HttpServlet {

    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        PrintWriter out=response.getWriter();
        request.getRequestDispatcher("frmLogin.jsp").include(request, response);
        String nome=request.getParameter("txtNome");
        String senha=request.getParameter("txtSenha");

        HttpSession session=null;

        if(nome.equals("helio") && senha.equals("prof")){
            session=request.getSession();
            session.setAttribute("nome", nome);
            session.setAttribute("status", "ok");
            response.sendRedirect("admPainel.jsp");
        }
        else{
            out.print("Indeferido");
        }
    }
}
```

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@page import=" javax.servlet.http.HttpSession" %>

<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset:

    <title>JSP Page</title>
  </head>
  <body>
    <h1>
      <%
        HttpSession s=null;
        s=request.getSession();
        String nome=(String)s.getAttribute("nome");
        String status=(String)s.getAttribute("status");
        out.print(nome + "<br>" +status);
      %>
    </h1>
  </body>
</html>
```

Jsp admPainel.jsp

Importa o HttpSession.

E recupera os dados de sessão

Trabalhando com outras classes

Para trabalhar com suas classes é necessário que elas estejam estruturadas em pacotes.

Veja um exemplo bem simples da classe Usuario que está no pacote minhasClasses

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@page import="minhasClasses.Usuario" %>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

    <title>JSP Page</title>
  </head>
  <body>
    <h1>
      <%
        Usuario u = new Usuario();
        u.setNomeUsuario("Hélio Esperidião");
        out.println(u.getNomeUsuario());
      %>
    </h1>
  </body>
</html>
```