



PROGRAMAÇÃO IV

Professor Felipe Fontoura

AULA 3 - AGENDA

- JSP, como adicionar lógica na camada de visualização
- Introdução aos Servlets
- Adicionando lógica nos Servlets
- Exercícios de fixação

JSP, COMO ADICIONAR LÓGICA NA CAMADA DE VISUALIZAÇÃO

UM EXEMPLO DE JSP BÁSICO

```
1  <%@page contentType="text/html" pageEncoding="UTF-8"%>
2  <!DOCTYPE html>
3  <html>
4      <head>
5          <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
6          <title>JSP Page</title>
7      </head>
8      <body>
9          <h1>Página simples</h1>
10     </body>
11 </html>
```

LÓGICA NA CAMADA DE VISUALIZAÇÃO

Apesar do JSP possuir uma sintaxe básica que se assemelha a HTML, nele, você pode adicionar partes de código em Java.

No final, “tudo” vira Java para o Servidor Tomcat.

LÓGICA NA CAMADA DE VISUALIZAÇÃO

Tudo que estiver entre as marcações `<%%>` são códigos Java.

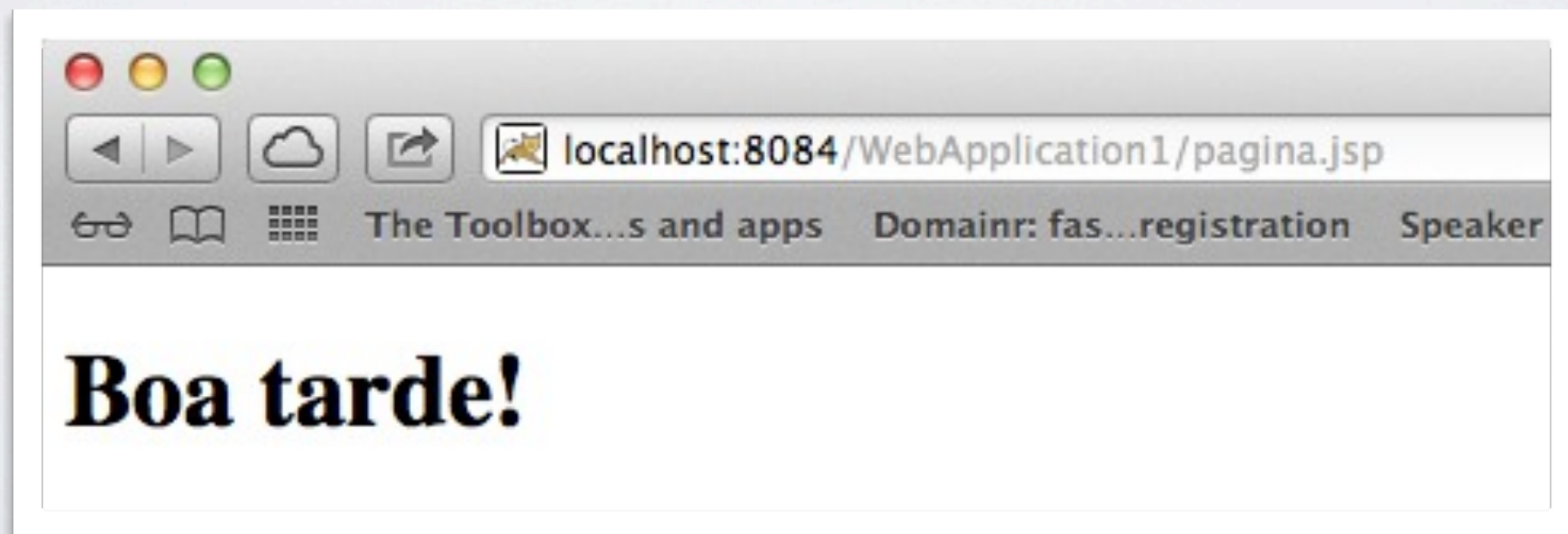
```
1  <%@page contentType="text/html" pageEncoding="UTF-8"%>
2  <!DOCTYPE html>
3  <html>
4      <head>
5          <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
6          <title>JSP Page</title>
7      </head>
8      <body>
9          <% String dizer_oi = "Olá pessoal!"; %>
10
11         <h1><%= dizer_oi %></h1>
12     </body>
13 </html>
```



VAMOS DEIXAR A PÁGINA INTELIGENTE?

```
1  <%@page import="java.util.Calendar"%>
2  <%@page contentType="text/html" pageEncoding="UTF-8"%>
3
4  <!DOCTYPE html>
5  <html>
6      <head>
7          <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
8          <title>Página bem educada</title>
9      </head>
10     <body>
11         <% Calendar hoje = Calendar.getInstance(); %>
12         <% int hora = hoje.get(Calendar.HOUR_OF_DAY); %>
13
14         <% if(hora >= 0 && hora <= 12) { %>
15             <h1>Bom dia!</h1>
16         <% } else if (hora >= 12 && hora <= 18) { %>
17             <h1>Boa tarde!</h1>
18         <% } else if (hora >= 18 && hora <= 0) { %>
19             <h1>Boa noite!</h1>
20         <% } %>
21     </body>
22 </html>
```


VAMOS DEIXAR A PÁGINA INTELIGENTE?



QUE TIPO DE CÓDIGO JAVA
EU POSSO ADICIONAR NO
JSP?

Um sistema inteiro!

SERVLETS, UM BOM LUGAR
PARA O SEU CÓDIGO JAVA
MORAR!

SERVLETS

○ O primeiro erro dos principiantes é adicionar qualquer tipo de lógica de negócio na camada de visualização (JSP).

HISTÓRICO DE SERVLETS

As Servlets são a primeira forma que veremos de criar páginas dinâmicas com Java. Usaremos a própria linguagem Java para isso, criando uma classe que terá capacidade de gerar conteúdo HTML. O nome “servlet” vem da ideia de um pequeno servidor (servidorzinho, em inglês) cujo objetivo é receber chamadas HTTP, processá-las e devolver uma resposta ao cliente.

Uma primeira ideia da servlet seria que cada uma delas é responsável por uma página, sendo que ela lê dados da requisição do cliente e responde com outros dados (uma página HTML, uma imagem GIF etc). Como no Java tentamos sempre que possível trabalhar orientado a objetos, nada mais natural que uma servlet seja representada como um objeto a partir de uma classe Java.

Cada servlet é, portanto, um objeto Java que recebe tais requisições (request) e produz algo (response), como uma página HTML dinamicamente gerada.

SERVLETS

Para escrevermos uma servlet, criamos uma classe Java que estenda **HttpServlet** e sobrescreva um método chamado **doGet** e **doPost** (principais métodos o HTTP). Estes métodos serão o responsáveis por atender requisições e gerar as respostas adequadas. Suas assinaturas:

```
@Override  
protected void doGet(HttpServletRequest request, HttpServletResponse response)  
    throws ServletException, IOException
```

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


SERVLETS

```
1  import java.io.IOException;
2  import java.io.PrintWriter;
3  import javax.servlet.*;
4  import javax.servlet.annotation.*;
5  import javax.servlet.http.*;
6
7  @WebServlet(urlPatterns = {"/MeuServlet"})
8  public class MeuServlet extends HttpServlet {
9      @Override
10     protected void doGet(HttpServletRequest request, HttpServletResponse response)
11         throws ServletException, IOException {
12         response.setContentType("text/html;charset=UTF-8");
13         PrintWriter out = response.getWriter();
14
15         try {
16             out.println("<!DOCTYPE html>");
17             out.println("<html>");
18             out.println("<head>");
19             out.println("<title>Meu Servlet</title>");
20             out.println("</head>");
21             out.println("<body>");
22             out.println("<h1>Olá, seu o seu Servlet!</h1>");
23             out.println("</body>");
24             out.println("</html>");
25         } finally {
26             out.close();
27         }
28     }
29 }
```


SERVLETS



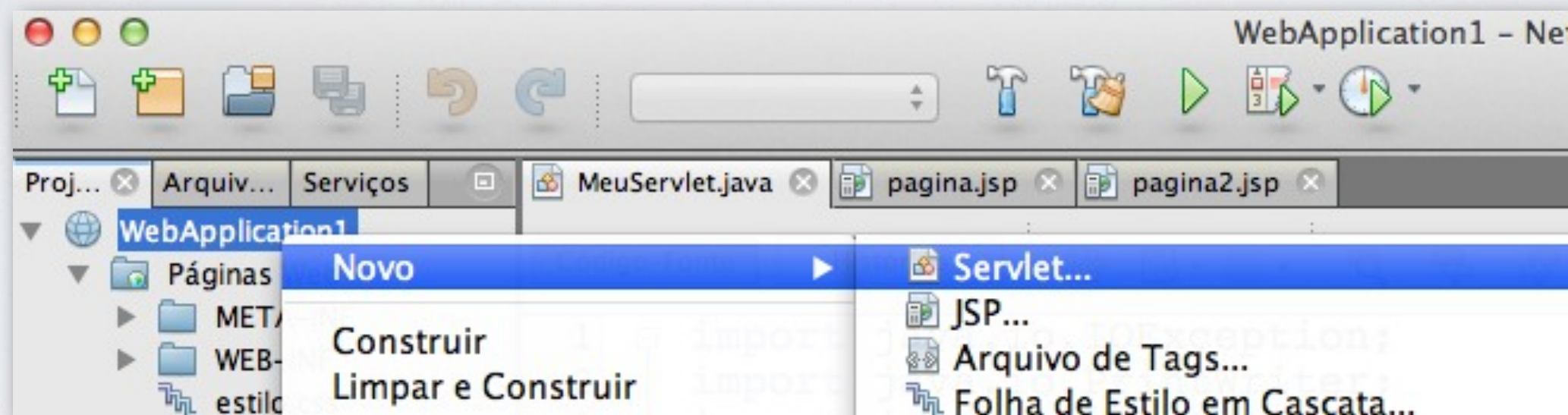
“PÁGINA EDUCADA” IMPLEMENTADA EM SERVLET

```
7  @WebServlet(urlPatterns = {"/MeuServlet"})
8  public class MeuServlet extends HttpServlet {
9      @Override
10     protected void doGet(HttpServletRequest request, HttpServletResponse response)
11         throws ServletException, IOException {
12
13         Calendar hoje = Calendar.getInstance();
14         int hora = hoje.get(Calendar.HOUR_OF_DAY);
15         String mensagem = "";
16
17         if(hora >= 0 && hora <= 12) {
18             mensagem = "Bom dia!";
19         } else if (hora >= 12 && hora <= 18) {
20             mensagem = "Boa tarde!";
21         } else if (hora >= 18 && hora <= 0) {
22             mensagem = "Boa noite!";
23         }
24
25         response.setContentType("text/html;charset=UTF-8");
26         PrintWriter out = response.getWriter();
27
28         try {
29             out.println("<!DOCTYPE html>");
30             out.println("<html>");
31             out.println("<head><title>Página bem educada</title></head>");
32             out.println("<body><h1>" + mensagem + "</h1></body>");
33             out.println("</html>");
34         } finally {
35             out.close();
36         }
37     }
38 }
```

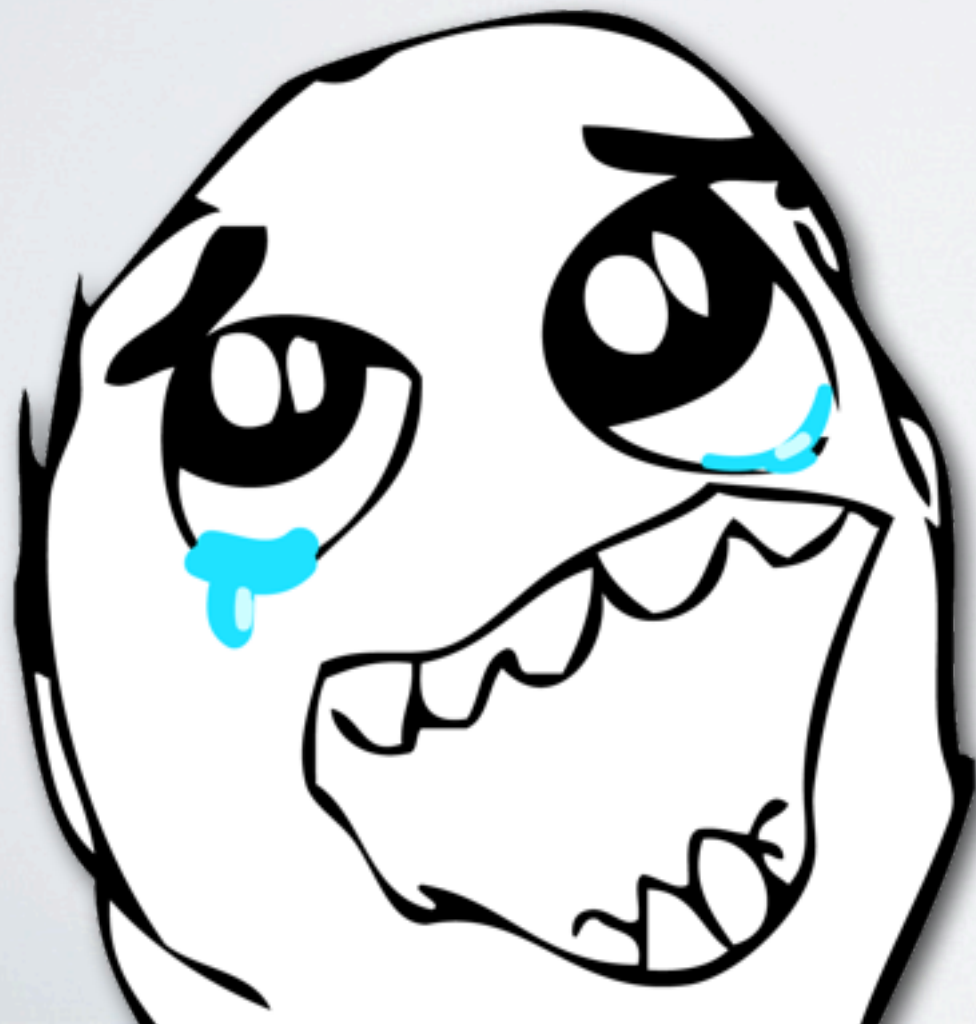

“PÁGINA EDUCADA” IMPLEMENTADA EM SERVLET



COMO CRIAR UM SERVLET NO NETBEANS?

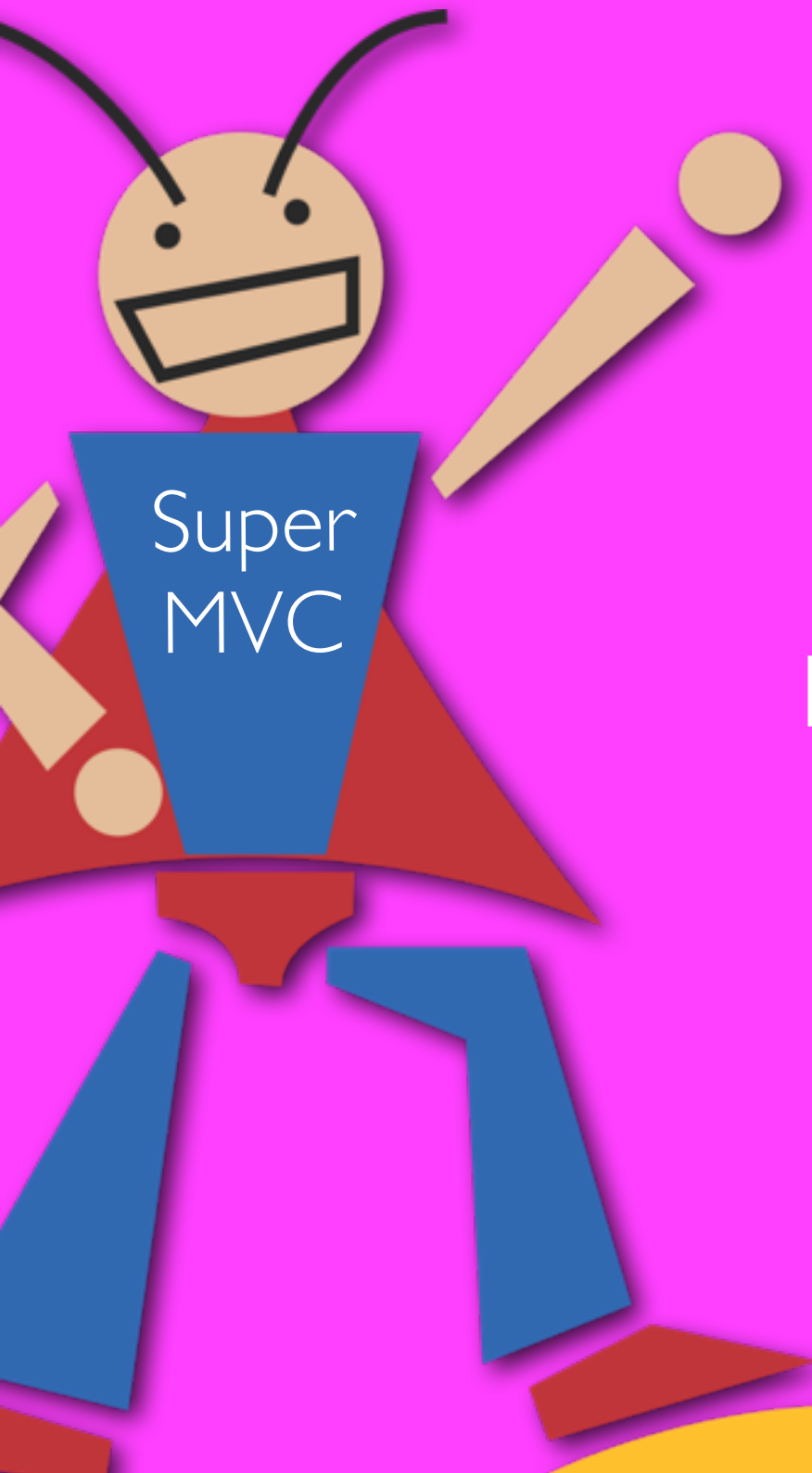


Neste momento, você deve estar pensando que conseguiu resolver o problema de onde adicionar a lógica, mas, utilizando **Servlets**, causou outro problema: Camada de visualização dentro do **Servlet**. E caso, necessite alterar qualquer **HTML**, irá ter que recompilar um código Java.



Web Designer (que odeia programação) pensando que terá que compilar Java.

ENTÃO QUAL É A SOLUÇÃO?



MVC

Model View Controller



Gambiarra

EXERCÍCIOS DE FIXAÇÃO

1. Crie um Servlet que seja capaz de sortear, para um apostador, os números da Mega Sena e exiba-os.

<http://www.kl9.com.br/artigos/numeros-aleatorios-em-java-a-classe-random/>