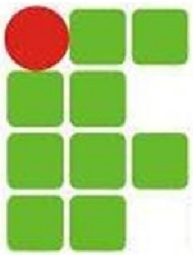
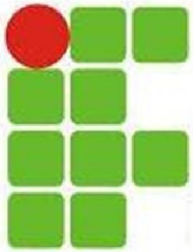
Programação Web

Professor: Diego Oliveira



Conteúdo 02:
JSP e Servlets

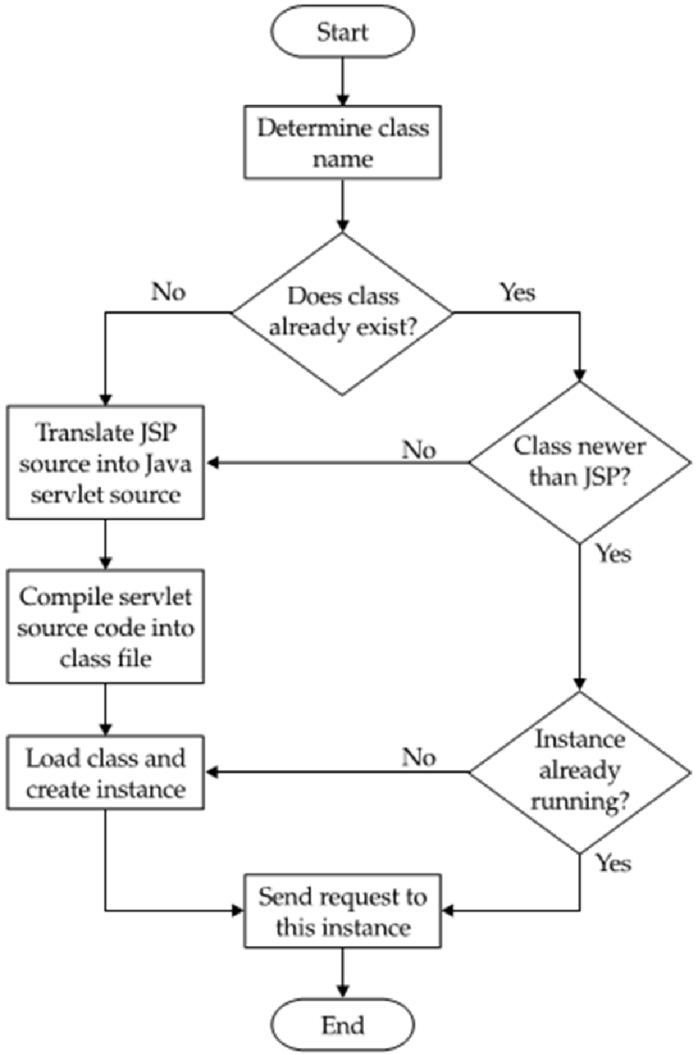


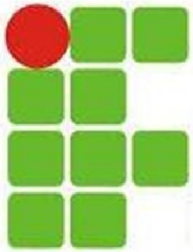
JSP

- JSP é um template de uma página Web que usa Java para gerar HTML dinamicamente
- JSP é considerado server-side e roda em um objeto chamado container, que os transformam em Servlets
- Suporta a parte de rede, acesso a banco de dados e threads do Java
- São recompilados automaticamente quando necessário

JSP

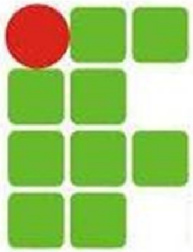
- O funcionamento do conteiner JSP é mostrado ao lado
- Basicamente ele verifica se já há uma classe para aquele JSP, caso negativo ele cria
- No caso de haver uma atualização, ele compila novamente





Diretivas JSP

- São utilizadas para passar informações sobre o JSP para o Tomcat
- Estas informações influenciam em como é feita a compilação dos Servlets
- Há três diretivas:
  - page
  - include
  - taglib

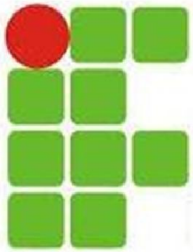


Diretiva page

- Ela funciona como o import do Java
- A sintaxe dela é assim:
  - <%@page import="java.util.ArrayList"%>
  - <%@page import=" java.util.Iterator"%>

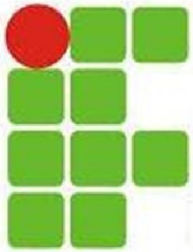
  ou

  - <%@page import="java.util.ArrayList, java.util.Iterator"%>



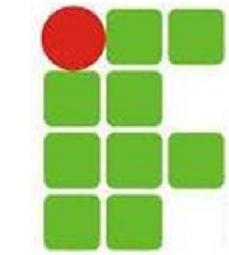
Diretiva include

- Permite a inclusão de trechos não compilados, ou seja, pedaços de JSP
- A sintaxe dela é assim:
  - <%@include file=“trecho1.jspf"%>
  - <%@include file="trecho2.jspf"%>
  - <%@include file="trecho3.jspf"%>
- JSPF = JSP Fragment



Diretiva taglib

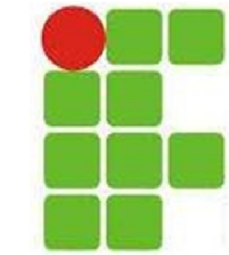
- Permite a criação dos seus próprios conjuntos de tags (Aquelas do HTML)
- Cada conjunto de tags deve ter um endereço e um prefixo
- A sintaxe é assim:
  - <%@taglib uri="http://mysite.com/mytags" prefix="my" %>
- Depois é possível utilizar as tags assim:
  - <my:umaTag> ... </my:umaTag>



Ações JSP

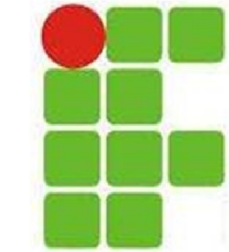
- São executadas enquanto o HTTP Request é processado no Tomcat
- Principais ações:
  - forward
  - include
  - param
- Exemplo de uso:

```
<jsp:forward page=“proximaPagina.jsp">
    <jsp:param name=“salario" value=“4000"/>
</jsp:forward>
```



Ações JSP

- Há mais ações como:
  - useBean
  - setProperty
- O useBean pode associar uma página JSP a um objeto Java:
  - <jsp:useBean id="dataManager" scope="application“ class=“myapp.model.DataManager"/>
- O setProperty configura os campos:
  - <jsp:useBean id="customer" class="eshop.beans.Customer"/>
  - <jsp:setProperty property="*" name="customer"/>

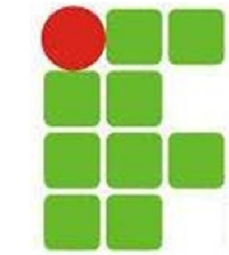


Exemplo JSP

index.jsp | teste.jsp

Source | History

```
1  <%--
2      Document   : teste
3      Created on : Oct 21, 2014, 10:34:37 PM
4      Author     : diego
5  --%>
6
7  <%@page contentType="text/html" pageEncoding="UTF-8"%>
8  <!DOCTYPE html>
9  <html>
10     <head>
11         <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
12         <title>Minha primeira página JSP</title>
13     </head>
14     <body>
15         <% for(int i=1; i<=10; i++){ %>
16             <h1> <%=i%> </h1>
17         <% } %>
18     </body>
19 </html>
20
```

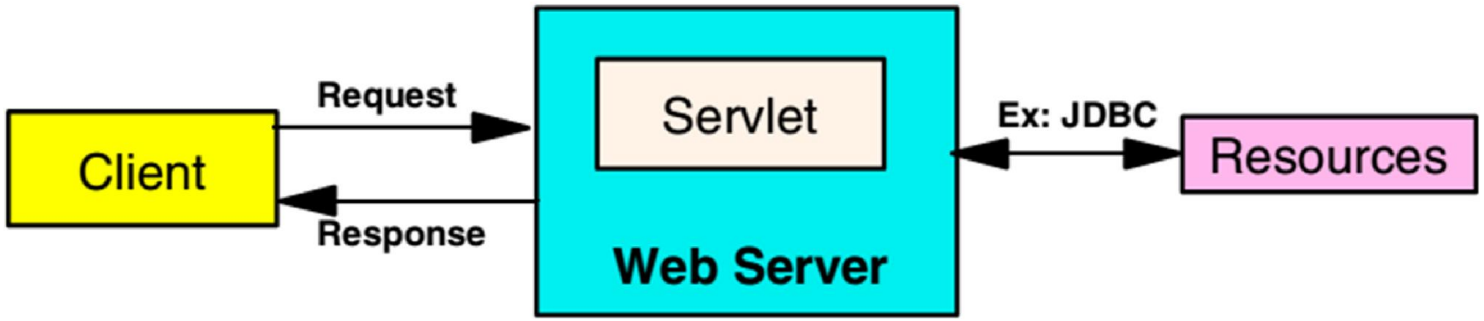


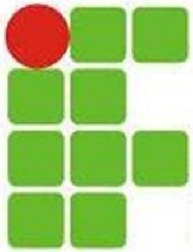
Servlet

- São componentes server-side escritos em Java e independentes de plataforma e de protocolos
- Não possuem interface com o usuário pois são executados dentro do servidor Web
- Podem responder requisições HTML ou construir páginas HTML dinamicamente
- Lembrando que internamente, no servidor Web, todo JSP é traduzido em um Servlet

Servlet

- Fluxo de processo do Servlet:
  1. O Cliente envia uma requisição para o servidor
  2. O servidor envia as informações da requisição para o Servlet
  3. O Servlet monta dinamicamente uma resposta e passa para o servidor
  4. O servidor envia a resposta para o cliente



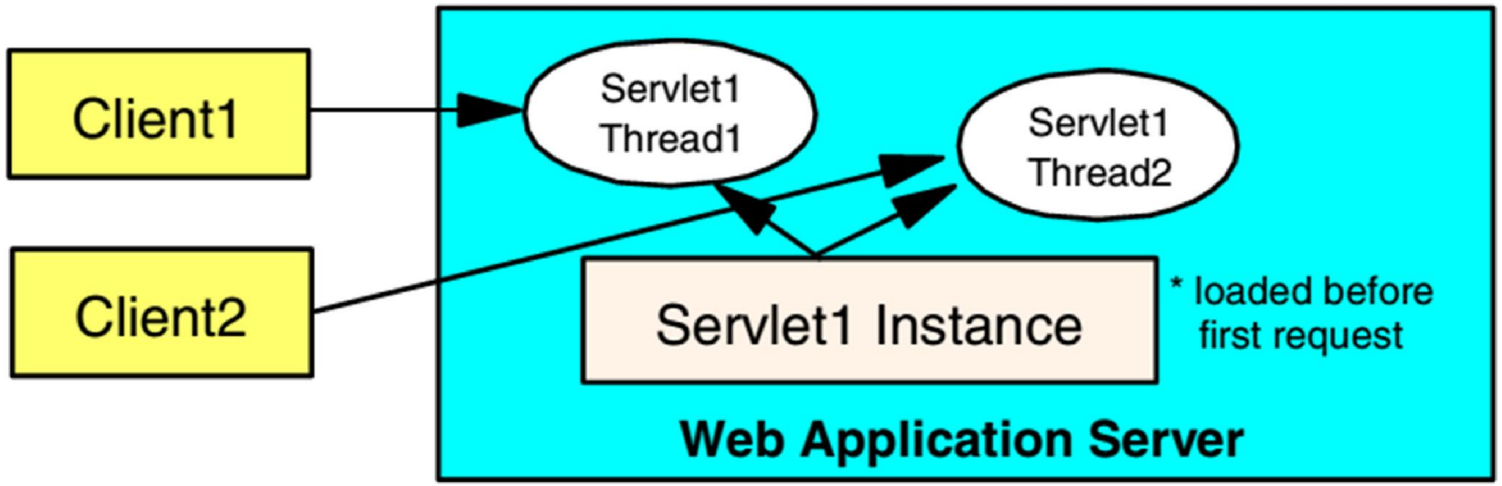


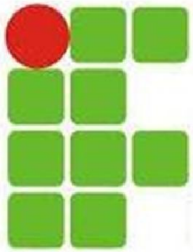
Servlet

- Vantagens dos Servlets:
  - São escrito em Java
    - Orientados a Objetos
    - Fortemente tipados
    - Modularizados
  - Possuem portabilidade
  - São independentes de plataforma
  - O Servlet é carregado apenas uma vez no servidor Web, ou seja, ele pode manter informações de sessão

Servlet

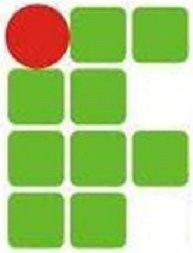
- Exemplo de ciclo de vida de um Servlet:
  - Servlet1 é carregado ao iniciar o servidor Web
  - Dois navegadores requisitam serviços dele
  - Duas threads são criadas, uma para gerenciar cada solicitação





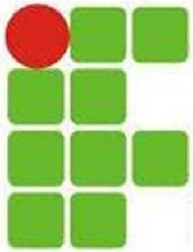
Exemplo de Servlet

```
1  package itso.servjsp.servletapi;
2  import java.io.*;
3  import javax.servlet.*;
4  import javax.servlet.http.*;
5  public class SimpleHttpServlet extends HttpServlet {
6      protected void service(HttpServletRequest req, HttpServletResponse res)
7          throws ServletException, IOException {
8          res.setContentType("text/html");
9          PrintWriter out = res.getWriter();
10         out.println("<HTML><TITLE>Servlet HTTP Simples</TITLE><BODY>");
11         out.println("<H2>Um exemplo simples da API Servlet HTTP</H2><HR>");
12         out.print("<H4>Servlet é uma forma simples de");
13         out.println("construir páginas dinâmicas!</H4>");
14         out.println("</BODY><HTML>");
15         out.close();
16     }
17 }
```



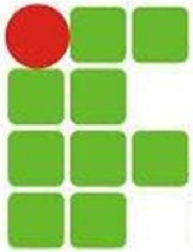
Explicando o Exemplo

- A classe criada é filha de **HttpServlet**
- O método **service()** é declarado abstrato na classe **HttpServlet**, então toda classe filha dela precisa implementá-lo
- A resposta do Servlet é em formato HTML, então a linha 8 declara isso
- A linha 9 cria um **PrintWriter** para que se possa escrever na saída padrão, **response**
- As demais linhas escrevem a página HTML



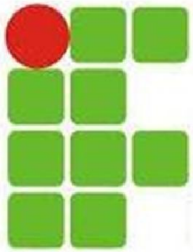
Como chamar um Servlet?

- Através da URL no navegador é possível chamar um Servlet via GET:
  - http://host/servlet/pacote.SimpleHttpServlet
- Ou então através de uma aplicação Web, que é a forma mais comum
- Para rodar em uma aplicação Web, basta criar um Web Project em um IDE qualquer e gerar um arquivo .war, depois colocá-lo no ou qualquer outro Servidor Web



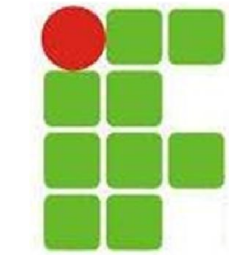
Resumindo!

- Vimos o que são Servlets, para que servem e como criar um Servlet e colocá-lo para rodar
- Também vimos o que é um JSP, para que ele é utilizado e como criar um projeto Web utilizando JSP na camada de apresentação
- Basicamente utilizamos HTML e Java para fazer páginas Web dinâmicas utilizando JSP e Servlets!



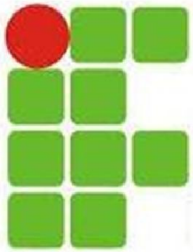
Exercício

- Criar um site JSP que contenha:
  - Elementos HTML convencionais
    - Imagens
    - Texto
    - Campos de Formulário
  - Elementos de repetição do Java
    - For
    - While
  - Elementos de controle de fluxo do Java
    - if, else



Exercício

- Crie um Servlet que escreva uma página contendo:
  - Elementos HTML convencionais
    - Imagens
    - Texto
    - Campos de Formulário
  - Uma contagem de 1 a 1000
  - Todas as letras do alfabeto, uma por linha



Perguntas?