

Servlets e JSP

Navegador



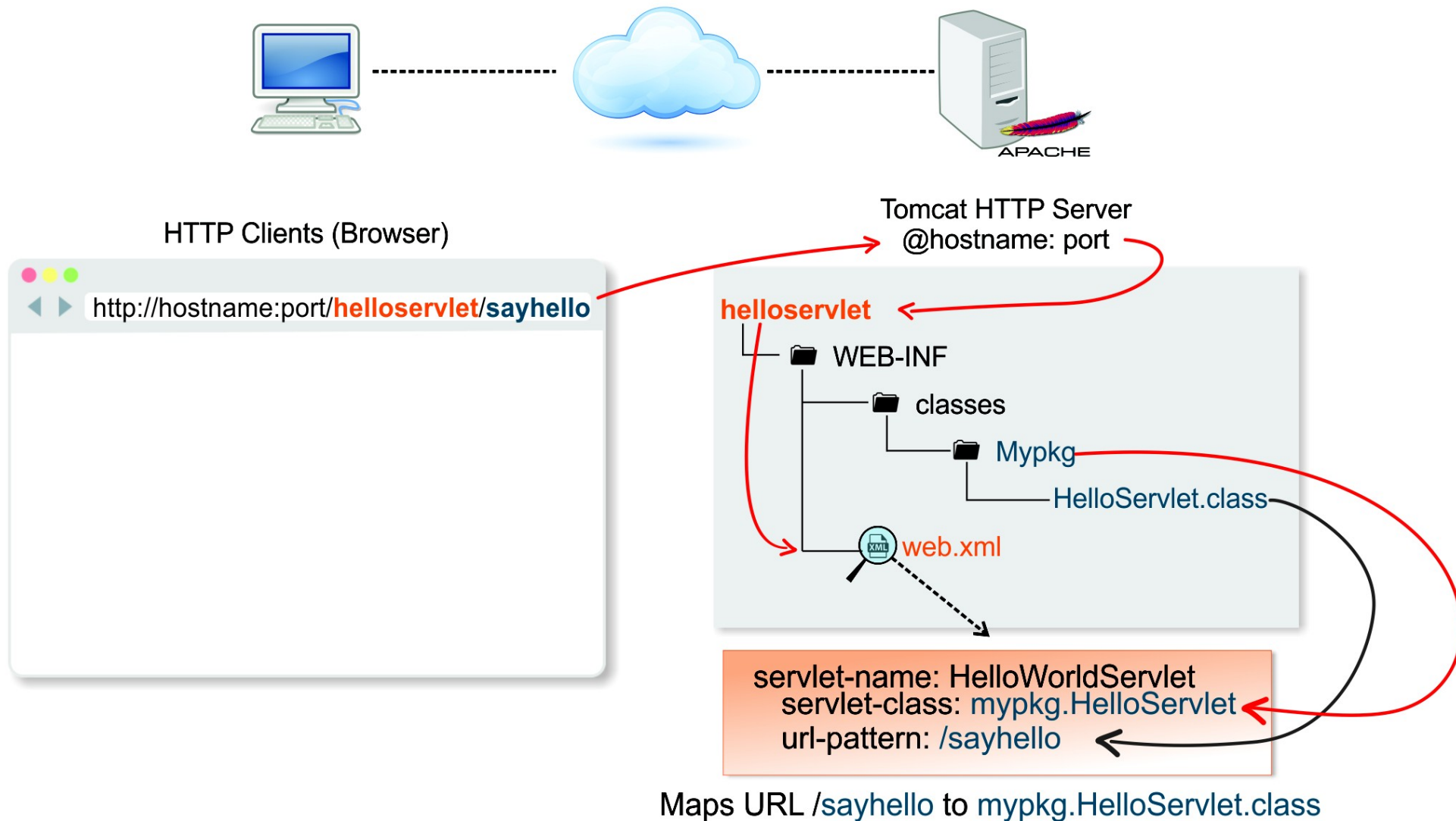
Web
Server

Servlets

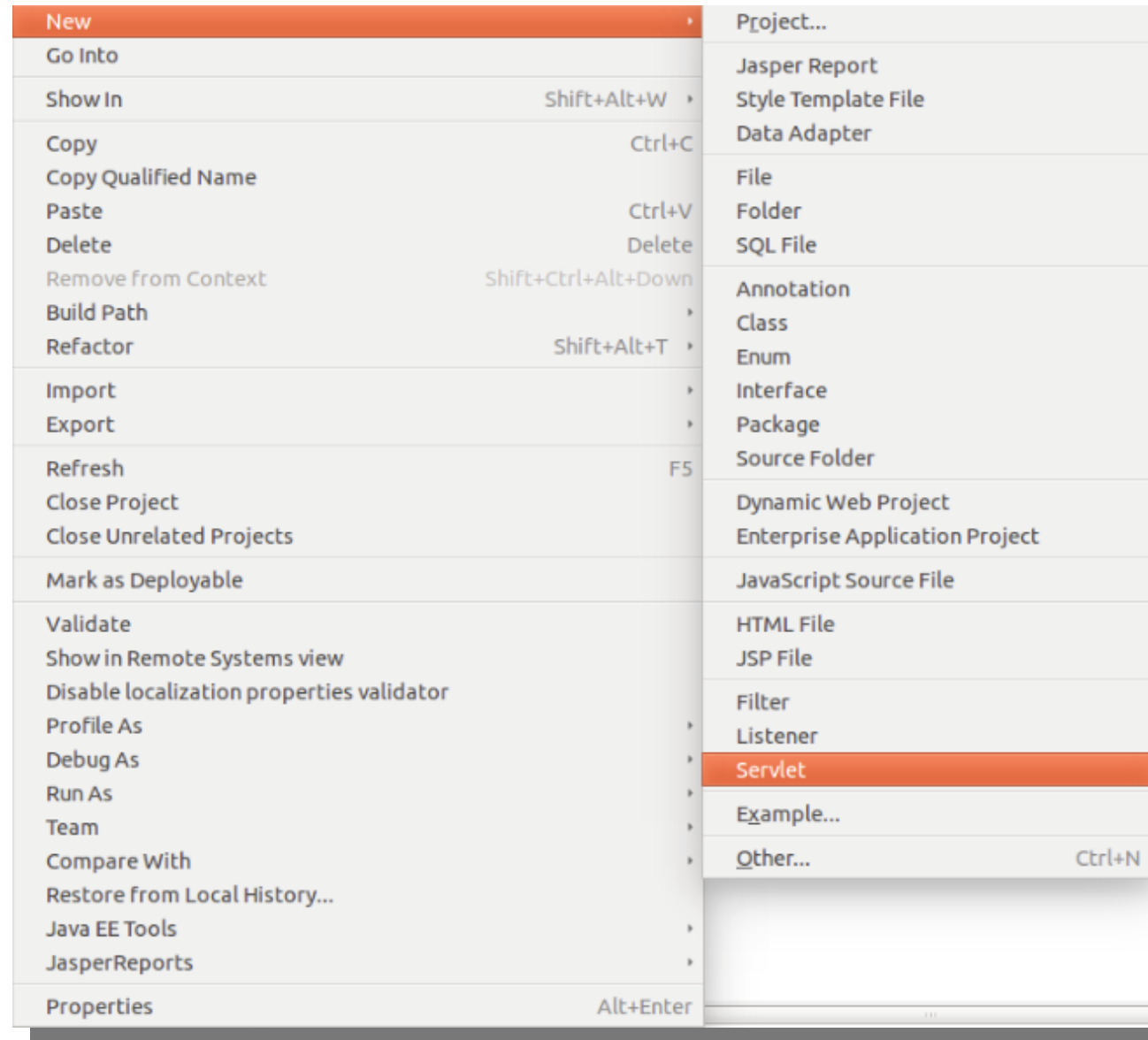
**Banco de
Dados**



Estrutura de uma Aplicação Web

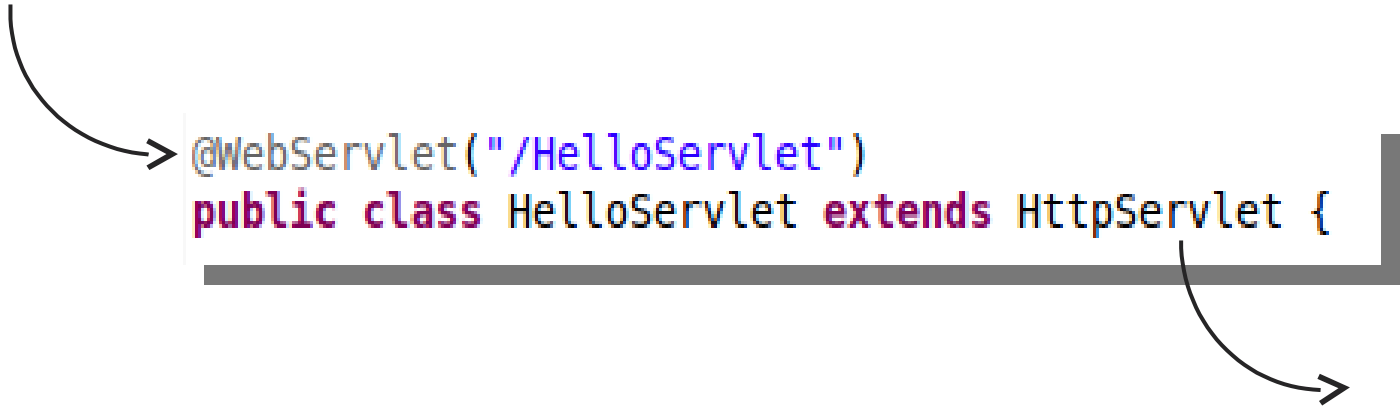


Criação de um Servlet



Criação de uma Servlet

Devemos anotar a classe com `@WebServlet`
para indicar qual a URL



```
@WebServlet("/HelloServlet")  
public class HelloServlet extends HttpServlet {
```

Uma classe servlet deve
estender de `HttpServlet`

Implementação de uma classe Servlet

```
@WebServlet("/ServletExemplo")
public class ServletExemplo extends HttpServlet {

    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public ServletExemplo() {
        super();
        // TODO Auto-generated constructor stub
    }

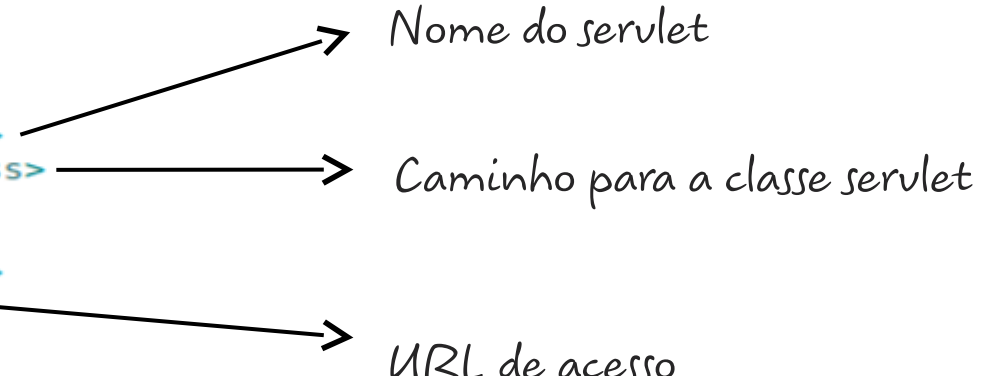
    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
     *      response)
     */
    protected void doGet(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException {
        // TODO Auto-generated method stub
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
     *      response)
     */
    protected void doPost(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException {
        // TODO Auto-generated method stub
    }
}
```

Mapeamento no web.xml (versões < 3.0)

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://java.sun.com/xml/ns/javaee"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
  id="WebApp_ID" version="3.0">
  <display-name>Hello</display-name>
  <welcome-file-list>
    <welcome-file>index.jsp</welcome-file>
  </welcome-file-list>

  <servlet>
    <servlet-name>HelloServlet</servlet-name>
    <servlet-class>HelloServlet</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>HelloServlet</servlet-name>
    <url-pattern>/HelloServlet</url-pattern>
  </servlet-mapping>
</web-app>
```



Nome do servlet

Caminho para a classe servlet

URL de acesso

Anotação `@WebServlet` faz a mesma coisa

Implementação

O método doGet é responsável por manipular requisições GET.

```
@WebServlet("/HelloServlet")
public class HelloServlet extends HttpServlet {

    private static final long serialVersionUID = 1L;

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

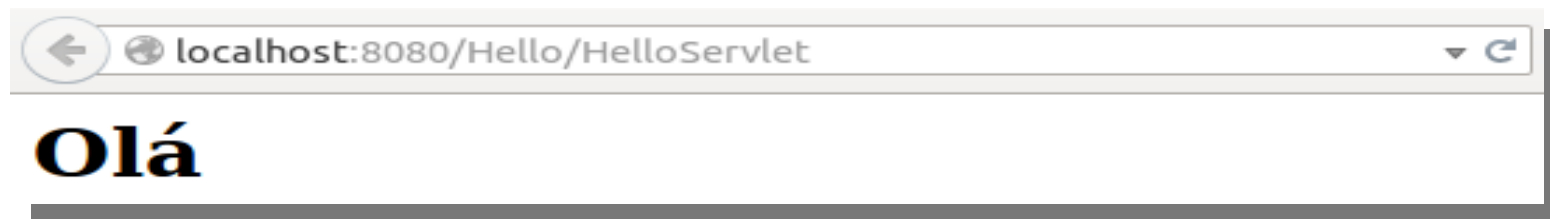
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println("<html>"
            + "<head><title>Hello Servlet</title></head>"
            + "<body>"
            + "<h1>Olá</h1>"
            + "</body></html>");

    }

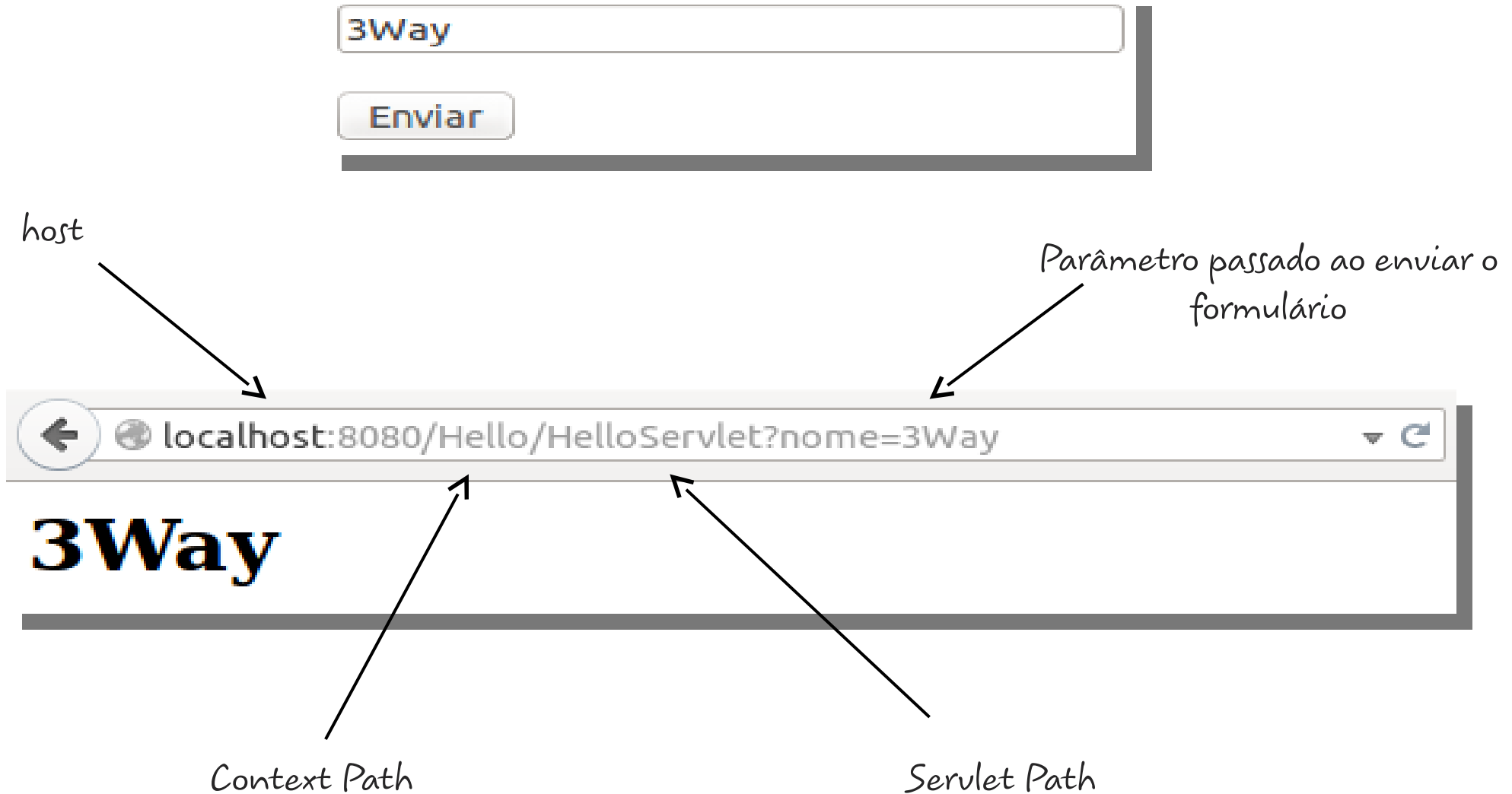
    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException {

    }

}
```



Recebendo Parâmetros



Dados de Formulário



3Way

```
@WebServlet("/HelloServlet")
public class HelloServlet extends HttpServlet {

    private static final long serialVersionUID = 1L;

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

        response.setContentType("text/html");
        PrintWriter out = response.getWriter();

        String nome = request.getParameter("nome");
        out.println("<html>"
            + "<body>"
            + "<h1>" + nome + "</h1>"
            + "</body></html>");

    }

    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException {

    }

}
```

```
<body>
  <form method="get" action="HelloServlet">
    <input type="text" name="nome" size="25">
    <p>
      <input type="submit" value="Enviar">
    </p>
  </form>
</body>
```

Expression Language \${ expressão }

Operador	Função
.	Acessar alguma propriedade
[]	Acessar algum elemento de uma lista
()	Agrupamento de subexpressões
+	Adição
-	Subtração
*	Multiplicação
/ ou div	Divisão
% ou mod	Resto
== ou eq	Igualdade
!= ou ne	Diferença
< ou lt	Menor que
> ou gt	Maior que
<= ou le	Menor ou igual a
>= ou ge	Maior ou igual a
&& ou and	E lógico
ou or	OU lógico
! ou not	Boleano
empty	Testa por variáveis vazias

Expression Language \${ expressão }

Números

Expressão	Resultado
\${ 1 < 2 }	true
\${ 1 lt 2 }	true
\${ 1 > (4/2) }	false
\${ 4.0 >= 3 }	false
\${ 4.0 ge 3 }	true
\${ 4 <= 3 }	true
\${ 4 <= 3 }	false
\${ 4 le 3 }	false
\${ 100.0 == 100 }	true
\${ 100.0 eq 100 }	true
\${ (10*10) != 100 }	false
\${ (10*10) ne 100 }	false

Alfanuméricos

Expressão	Função
\${ 'a' < 'b' }	true
\${ '4' > '3' }	true

Expressão	Resultado
\${ 1 }	1
\${ 1.2 + 2.3 }	3.5
\${ 1.2E4 + 1.4 }	12001.4
\${ -4 -2 }	-6
\${ 21 * 2 }	42
\${ 3/4 }	0.75
\${ 3 div 4 }	0.75
\${ 3/0 }	Infinity
\${ 10%4 }	2
\${ 10 mod 4 }	2
\${ (1==2) ? 3:4 }	4

Expression Language \${ expressão }

Objeto Implícito	Mapeia
pageScope	Atributo no escopo da página
requestScope	Atributo no escopo da requisição
sessionScope	Atributo no escopo da sessão
param	Parâmetro da requisição
paramValues	Parâmetro da requisição
header	Cabeçalho da requisição
headerValues	Cabeçalho da requisição
cookie	Cookie
initParam	Parâmetro de inicialização de contexto

Alterar Parâmetro

foo =

Enviar dados

EL Expression	Result
<code>\${param.foo}</code>	3way
<code>\${param["foo"]}</code>	3way
<code>\${header["host"]}</code>	localhost:8080
<code>\${header["accept"]}</code>	text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
<code>\${header["user-agent"]}</code>	Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:39.0) Gecko/20100101 Firefox/39.0

Página JSP

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ page import="java.util.Date"%>
<!DOCTYPE html>
<%!
    int contador = 0;
    Date agora;
%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>JSP Essencial</title>
</head>
<body>
    <form method="get">
        <label for="nome">Nome</label>
        <input type="text" id="nome" name="nome" size="25">
        <label for="numero">Numero</label>
        <input type="text" id="numero" name="numero" size="10">
        <input type="submit" value="Enviar">
    </form>
    <%
        String pNome = request.getParameter("nome");

        if(pNome != null && ! pNome.isEmpty()){
            contador ++;
            agora = new Date();
        }
    %>
    <h3>Número: ${param.numero}</h3>
    <h2>${param.nome}</h2>
    <h4><%= contador %> em <%= agora %></h4>
    <% } %>
    <jsp:include page="rodape.html" />
</body>
</html>
```

Diretivas

Declaração

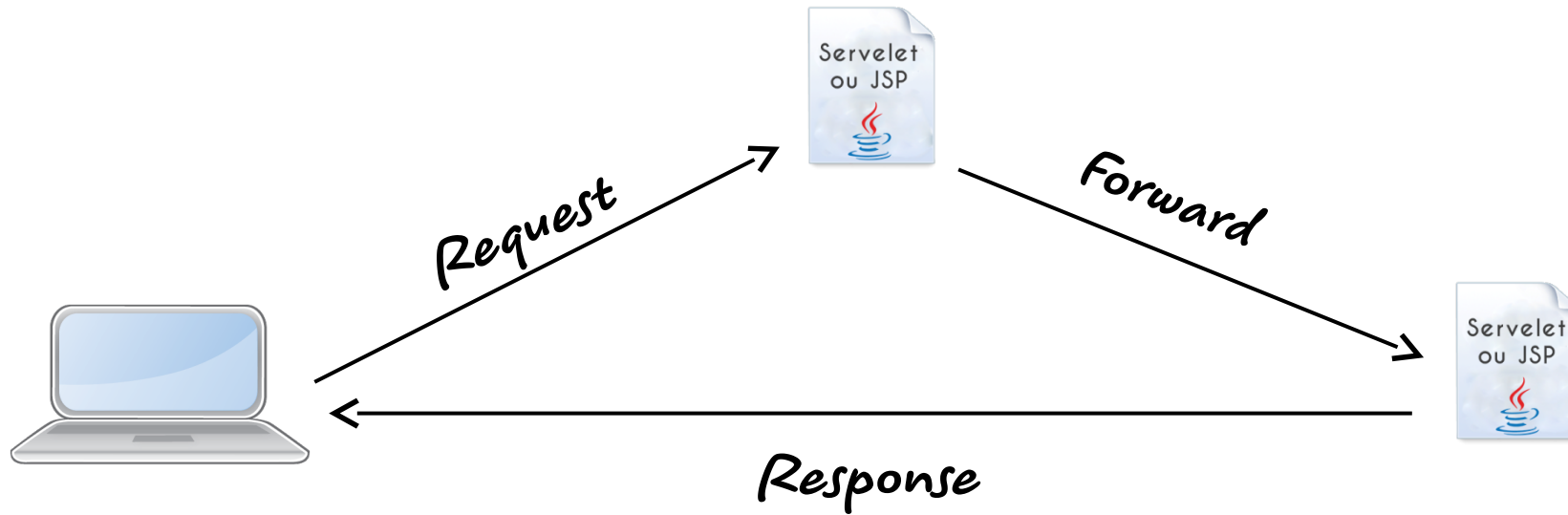
Scriptlet são pedaços de código java em sua página JSP. Deve estar entre as tags `<% %>`.

Expressão EL

Expressão JSP

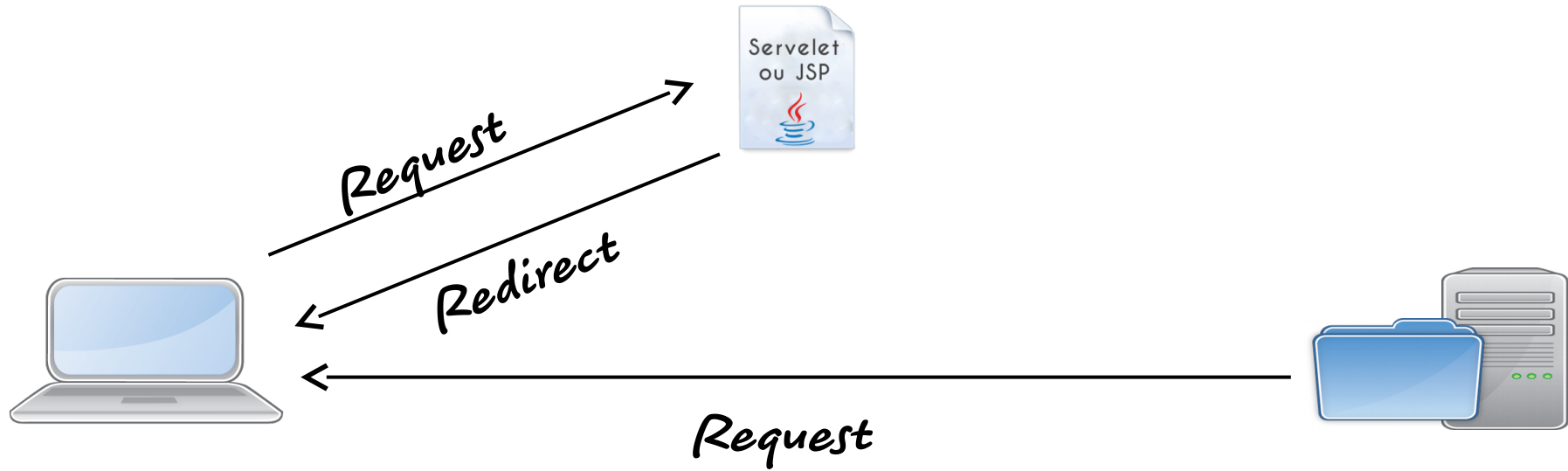
Ação JSP

Encaminhar (forward)



```
Public [doPost | doGet] (...) throws ...{  
  
Request.getRequestDispatcher("/resultListagem.jsp")  
    .forward(request, response);  
  
}
```

Redirecionar (redirect)



```
Public [doPost | doGet] (...) throws ...{  
  
response.sendRedirect("http://www.google.com.br");  
return;  
  
}
```

Paginas de Erro

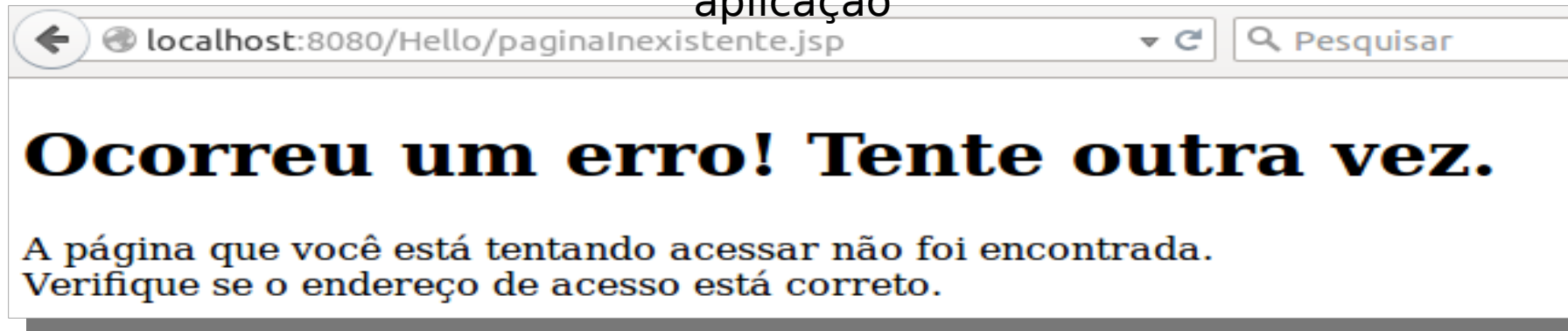
Declare a página de erro no web.xml

```
<error-page>  
  <error-code>404</error-code>  
  <location>/errorPages/erro404.jsp</location>  
</error-page>
```

Código do erro HTTP

Caminho para a página de erro

Acessando uma página inexistente na aplicação



JSTL JavaServer Pages Standart Tag Library

Adicionar os jars da JSTL
no build path do projeto



**Download da
Implementação**

<http://jstl.java.net/>

usa

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>  
<%@ taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt" %>  
<%@ taglib uri="http://java.sun.com/jsp/jstl/sql" prefix="sql" %>  
<%@ taglib uri="http://java.sun.com/jsp/jstl/xml" prefix="x" %>  
<%@ taglib uri="http://java.sun.com/jsp/jstl/functions" prefix="fn" %>
```

JSTL Core Tags

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
```

Podemos importar
outras páginas jsp
(cabeçalhos, rodapés).

```
<c:import url="http://www.google.com.br"/>  
<c:import url="foreach.jsp"/>
```

Testa se o parâmetro
nome está vazio e exibe
uma mensagem.

```
<c:if test="${empty param.nome}">  
    Você não preencheu o seu nome!  
</c:if>  
<c:if test="${not empty param.nome}">  
    O seu nome é ${param.nome}  
</c:if>
```

JSTL Core Tags

Funciona como um
for em java.

```
<c:forEach var="item" begin="1" end="10">
    ${item}
</c:forEach>
```

1 2 3 4 5 6 7 8 9 10

Podemos pegar um atributo
passado na sessão.

```
<table border="1">
  <tbody>
    <c:forEach items="${listPessoas}" var="pessoa">
      <tr>
        <td>${pessoa.nome}</td>
      </tr>
    </c:forEach>
  </tbody>
</table>
```

Servlet com <c:forEach>

Variável de sessão

Passamos uma lista como
atributo da sessão

```
@WebServlet("/HelloServlet")
public class HelloServlet extends HttpServlet {


    private static final long serialVersionUID = 1L;

    private ArrayList<Pessoa> listPessoas = new ArrayList<Pessoa>() {
        {
            add(new Pessoa("Joao"));
            add(new Pessoa("Maria"));
        }
    };

    @Override
    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        listPessoas.add(new Pessoa(request.getParameter("nome")));
        request.getSession().setAttribute("listPessoas", listPessoas);
        response.sendRedirect("listpessoas.jsp");
    }
}
```

```
<table border="1">
<tbody>
<c:forEach items="${listPessoas}" var="pessoa">
<tr>
<td>${pessoa.nome}</td>
</tr>
</c:forEach>
</tbody>
</table>
```



Joao
Maria
Lucas
3way

Outras tags JSTL core

