

FIAP GRADUAÇÃO

DESENVOLVIMENTO DE SISTEMAS WEB

PROF. THIAGO T. I. YAMAMOTO
thiagoyama@gmail.com

INTEGRAÇÃO JSPs E SERVLETS

1. Integração entre os componentes: Servlets e JSPs
2. Parâmetros
3. Atributos
4. Escopo dos atributos:
 1. Escopo de página
 2. Escopo de request
 3. Escopo de context (aplicação)
 4. Escopo de sessão
5. Redirecionamento:
 1. Forward
 2. Redirect

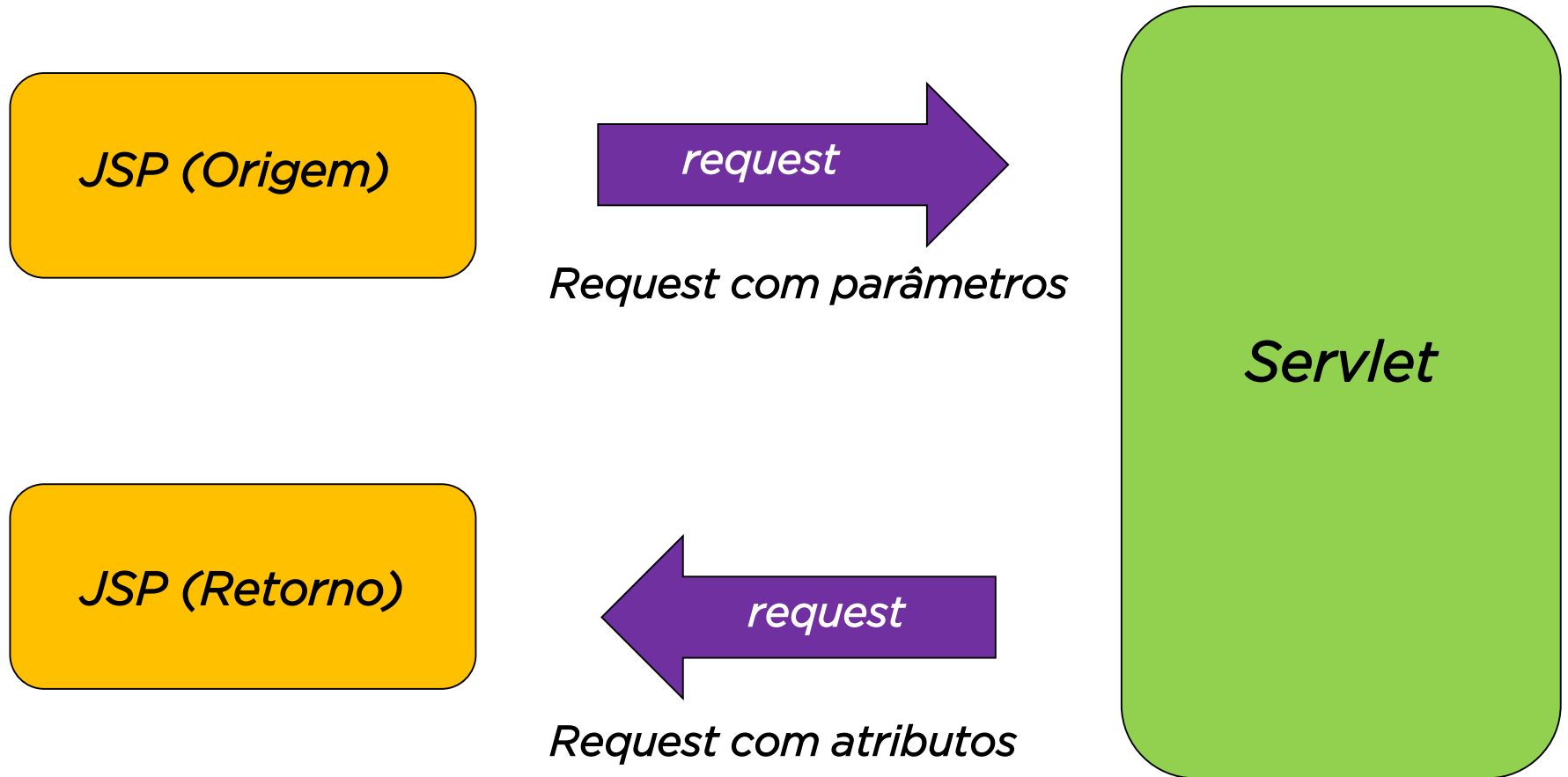
INTEGRAÇÃO ENTRE OS COMPONENTES JAVA WEB – JSPs E SERVLETS

Como podemos passar informações entre os JSPs e Servlets?

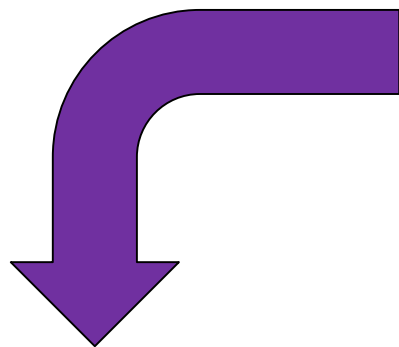


Existem algumas formas, as principais são:

- JSP para a Servlet:
 - Parâmetros no request;
- Servlet para o JSP:
 - Atributos;



- Os parâmetros estão na request, em requisições GET ou POST.

**JSP**

```
<form action="cadastrar" method="get">
  <label>Nome</label>
  <input type="text" name="nome"/>
  <label>Email</label>
  <input type="text" name="email"/>
  <input type="submit" value="Enviar"/>
</form>
```

Servlet

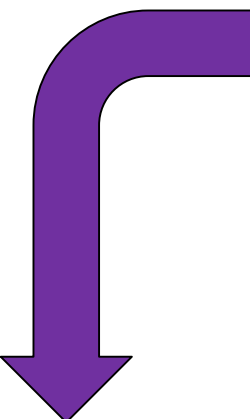
```
public void doGet(HttpServletRequest req,
  HttpServletResponse resp) throws ServletException,
  IOException {
```

```
    String nome = req.getParameter("nome");
    String email = req.getParameter("email");
```

```
}
```

São informações que podem ser colocados em uma “área” para ser compartilhado entre os componentes;

Servlet



```
protected void doGet(HttpServletRequest req,
    HttpServletResponse resp)
    throws ServletException, IOException {

    String nome = "Thiago";
    req.setAttribute("usuario", nome);
}
```

JSP



```
<% String nome = (String) request.getAttribute("usuario"); %>

<h1>Olá <%=nome %></h1>
```


A visibilidade e o tempo de vida dos atributos depende do escopo que ele está inserido:

Escopo	Tempo de vida	Quando é criado	Quando é destruído
PageContext	Enquanto a página está sendo processada	Quando a página é processada	Quando a página para de ser processada
ServletContext	Enquanto a aplicação estiver no ar	Quando a aplicação ou o servidor é inicializado	Quando a aplicação ou o servidor é desativado
HttpSession	Enquanto o usuário estiver logado na aplicação	No primeiro request efetuado pelo usuário	Quando invalidado por timeout ou programaticamente
HttpServletRequest	Durante o processamento de um request	A cada novo request	Após response é retornado

- Utilizados dentro das páginas JSPs
- Um atributo definido no escopo de página só é visível dentro do JSP.

JSP

```
<!--Atribui um valor no escopo de página -->  
<% pageContext.setAttribute("tipoUsuario", "Administrador"); %>  
  
<!-- Recupera o atributo e exibe na página -->  
<h1>Olá <%=pageContext.getAttribute("tipoUsuário") %> </h1>
```

Os atributos estarão presentes no request enquanto durar a requisição:

Servlet

```
protected void doGet(HttpServletRequest req,  
    HttpServletResponse resp)  
    throws ServletException, IOException {  
  
    String nome = "Thiago";  
    req.setAttribute("login", nome);  
}
```

JSP

```
<% String user = (String) request.getAttribute("login"); %>
```



Os atributos são compartilhados por todos os componentes

Servlet

```
ServletContext ctx = getServletContext();  
  
//Coloca um atributo no contexto  
ctx.setAttribute("login", "Thiago");  
  
//Recupera um atributo do contexto  
String nome = ctx.getAttribute("login");
```

JSP

```
<% application.setAttribute("login","Thiago"); %>  
  
<%= application.getAttribute("login"); %>
```

- É representada pela interface **HttpSession**
- Os atributos são exclusivos de um usuário da aplicação, ou seja, a sessão é única para cada cliente
- A sessão do usuário é destruída por **timeout** (configurado no web.xml) ou via **programação** (código)



```
//Destroi a sessão do usuário  
session.invalidate();
```

Servlet

```
//Recupera a sessão do usuário ou cria uma nova se não existe
HttpSession session = request.getSession();
//Adiciona o atributo usuário na sessão
session.setAttribute("usuario", "Thiago");
//Recupera o atributo usuario da sessão
String usuario = (String) session.getAttribute("usuario");
```

JSP

```
<%
//Adiciona o atributo na sessão
session.setAttribute("usuario", "Thiago");
//Recupera o atributo na sessão
String user = (String) session.getAttribute("usuario");
%>
```

- É possível configurar o tempo de timeout, ou seja, o tempo necessário de inatividade do usuário no sistema para que a sua sessão seja destruída.

Web.xml

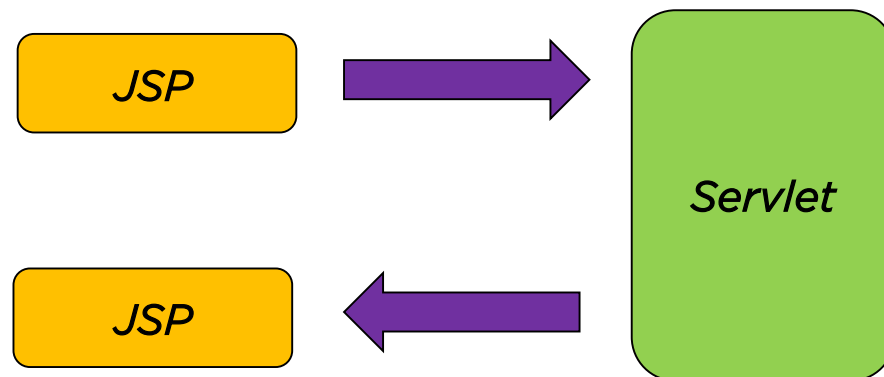
```
<session-config>  
    <session-timeout>30</session-timeout>  
</session-config>
```



Tempo definido
em minutos

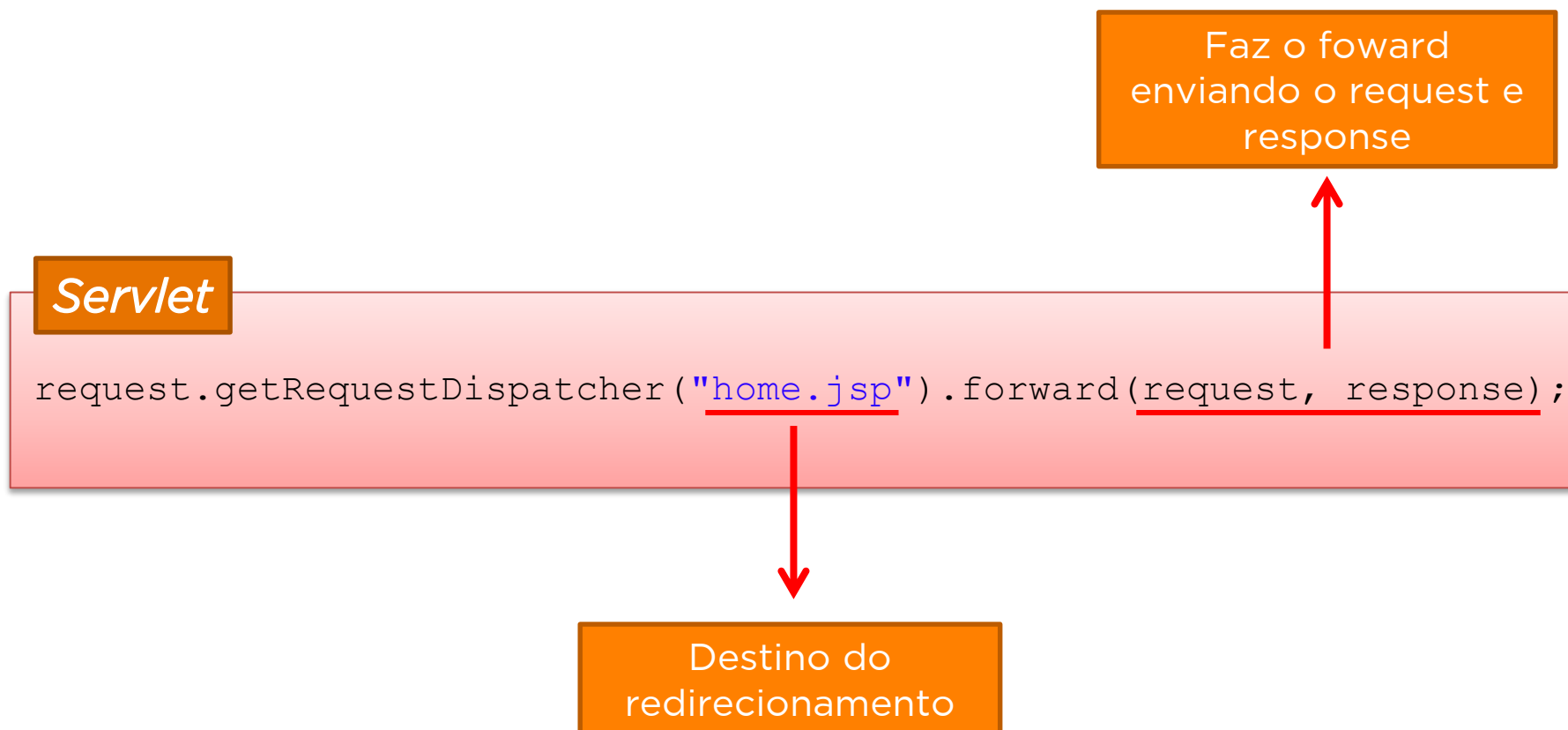
REDIRECCIONAMIENTO DE PÁGINAS

- Após um servlet processar a requisição do usuário, ele pode redirecionar a requisição para um arquivo JSP, que será exibido para o usuário
- Existem duas formas de redirecionar:
 - Forward
 - Redirect



- É executado internamente pelo servlet (Controller);
- O browser não sabe o que está ocorrendo durante o processamento no servidor, ou seja, não sabe por quais servlets ou páginas a requisição esta passando;
- No final do processamento da requisição a URL da barra de endereços do browser não muda;
- O reload da página resultante irá executar a requisição original.

Redirecionamento interno feito pelo servidor



- É um processo de dois passos, ao receber uma requisição a aplicação web “solicita” ao browser para acessar uma segunda URL, por isso a URL muda;
- O reload de página não repetirá a requisição original, mas sim a nova URL (2ª Requisição);
- É um processo muito mais lento que um forward, pois são necessárias duas requisições, e não apenas uma;
- Objetos colocados no escopo do request original são perdidos durante o segundo request.

Redirecionamento feito pelo cliente (outra URL)

- É possível ver na URL o redirecionamento
- É possível redirecionar o usuário para uma URL externa.

Servlet

```
response.sendRedirect("home.jsp");
```



Destino do
redirecionamento

Copyright © 2013 - 2016 Prof. Me. Thiago T. I. Yamamoto

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor (autor).

*“Sempre que você vir uma pessoa de sucesso, você sempre verá as glórias,
nunca os sacrifícios que os levaram até ali” – Vaibhav Shah*