

**FIAP GRADUAÇÃO**

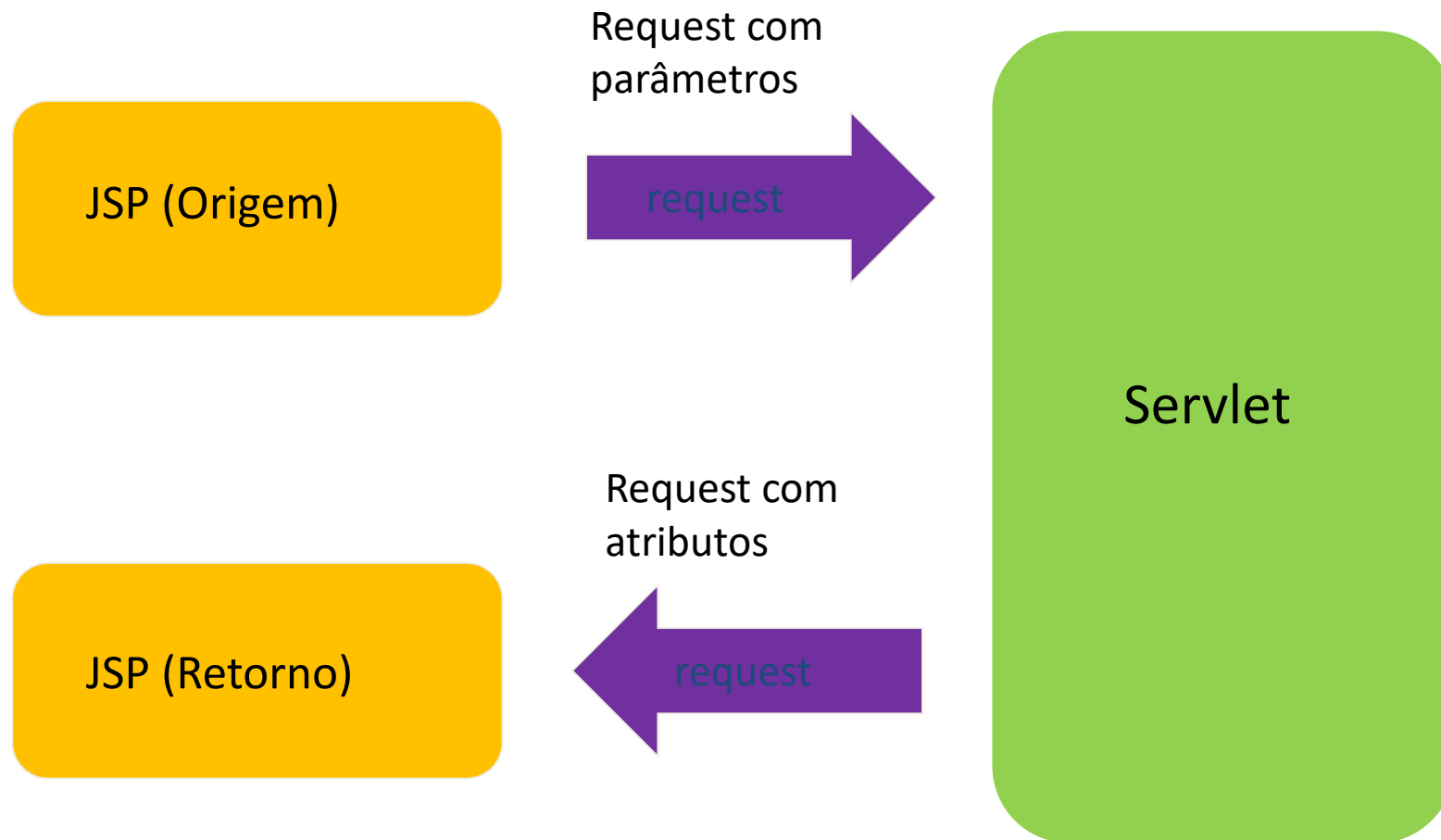
# RESPONSIVE WEB DEVELOPMENT

PROF. RAFAEL MATSUYAMA  
profrafael.matsuyama@fiap.com.br

INTEGRAÇÃO SERVLET E JSP

- Integração entre os componentes: Servlets e JSPs
- Parâmetros
- Atributos
- Escopo dos atributos:
  - Escopo de página
  - Escopo de request
  - Escopo de context (aplicação)
  - Escopo de sessão
- Redirecionamento:
  - Forward
  - Redirect

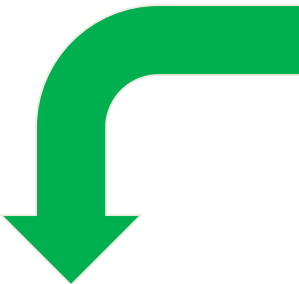
# I INTEGRAÇÃO ENTRE OS COMPONENTES FIAP



# I PARÂMETROS

- 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

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

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

```
}
```

# ATRIBUTOS

- 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 = "Luís";  
    req.setAttribute("usuario", nome);  
}
```

JSP

```
<% String nome = (String) request.getAttribute("usuario"); %>  
  
<h1>Olá <%=nome %></h1>
```

# ESCOPO DOS ATRIBUTOS

- 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

# I ESCOPO DE PÁGINA

- 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>
```



# I ESCOPO DE REQUEST

- Os atributos estarão presentes no request enquanto durar a requisição

## Servlet

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

## JSP

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



# I ESCOPO CONTEXT (APPLICATION)

- Os atributos são compartilhados por todos os componentes

## Servlet

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

## JSP

```
<% application.setAttribute("login","Luis"); %>  
  
<%= 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", "Luis");
//Recupera o atributo usuario da sessão
String usuario = (String) session.getAttribute("usuario");
```

## JSP


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

# CONFIGURAÇÃO TIMEOUT (WEB.XML)

- 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

- 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

# I FORWARD

- É um redirecionamento interno feito pelo servidor
  - Não é possível ver na URL para onde foi feito o redirecionamento.

Servlet

```
request.getRequestDispatcher("home.jsp").forward(request, response);
```

Faz o foward enviando o request e response

Destino do redirecionamento

# I REDIRECT

- É um 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 © 2018 Prof. Rafael Matsuyama / Prof. Luís Carlos de S Silva

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