DR3

Disciplina Regular 3

Desenvolvimento Web com Java EE

Graduação em Engenharia de Software - 2020

DR3

Etapa 2 Aula 1

JavaServer Pages

Para acompanhar pelo Moodle você deve estudar a etapa 3

DR3

Etapa 2 Aula 1 - Competências e Checklist

- **Competências** Trabalhadas Nesta Etapa
  - **Construir aplicativos Java utilizando JSP.**
  - Construir aplicativos Java utilizando Servlets.
  - Escrever interfaces simples com HTML e CSS.
- **Checklist** Desta Aula:
  - Já ter o ambiente de desenvolvimento funcionando.
  - Já ter revisado os slides e feito os exercícios da etapa 1.
  - Já ter lido os capítulos 1 e 2 do livro "Head First Servlets and JSP, 2nd Edition" disponível na O'Reilly Safari.

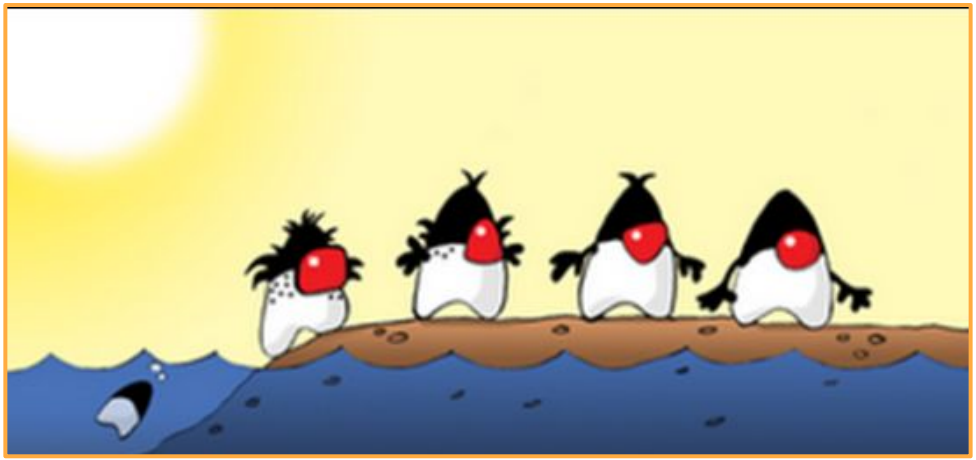
DR3

Elementos Principais

DR3

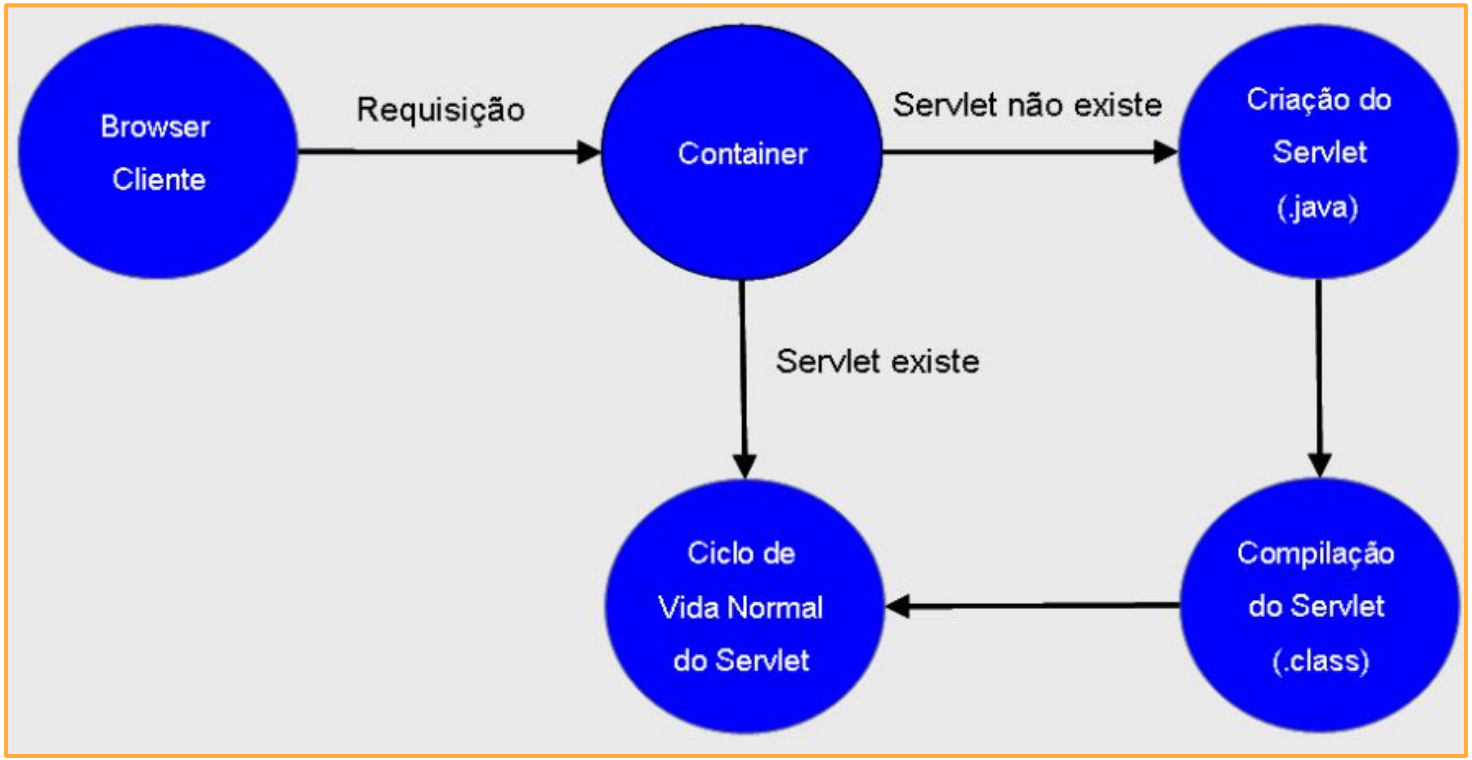
O que é JSP?

- Alternativa aos servlets na geração de conteúdo.
- Tecnologia de scripts para geração de conteúdo dinâmico.
- Permite a separação da lógica de aplicação da lógica de conteúdo.



DR3

Processamento do JSP



Páginas JSP se transformam em servlets por questões de otimização.

Deployment das Páginas

DR3

- Com relação à localização, as páginas JSP são tratadas como arquivos comuns, não tendo um lugar específico.

- O container possui um mapeamento interno da extensão ***.jsp**.

- Sempre que uma requisição com a extensão **.jsp** chega, o container inicia o seu processamento.

Scripting

DR3

- Páginas JSP podem conter trechos de código Java = Scripting.

- **O conceito de separação entre lógica de apresentação e negócio fica prejudicado**.

- Scripting deve ser usado em casos bem específicos.

- A forma correta é utilizar JSTL+EL como veremos adiante.

Expressões de Saída

DR3

- Usa-se: `<%= expressão %>` que será o equivalente a fazer `out.println(expressão);`
- A expressão tem que retornar um resultado que possa ser convertido em String.
- Erros de sintaxe só serão emitidos quando o JSP for convertido e compilado.
- Expressões são muito úteis na atribuição de valores para atributos html:

```
<font size = '<%=Math.random()*5%>' >
```

Exemplo

- Uma página JSP mostrando a data atual:

```
<h2>Banco Money</h2>

<b>Data:</b> <%= new java.util.Date() %><br>

<form action='ValidarLoginServlet' method='POST'>
...
```

Scriptlet

DR3

- Usados para incluir trechos de código dentro do JSP.
- São incluídos nos métodos do servlet derivado no mesmo local onde estão no JSP.
- Scriptlets são incluídos no JSP com a sintaxe:

```
<%
código Java
%>
```

Banco Money

DR3

- Mudança na formatação da data mostrada na página JSP:

```
<%
String dataFormatada = null;
DateFormat df = DateFormat.getDateInstance(
  DateFormat.FULL);
dataFormatada = df.format(new Date());
%>
<h2>Banco Money</h2>
<b>Data:</b> <%= dataFormatada %><br />
```

Comentários

- Usa-se a tag abaixo para comentários JSP:

```
<%-- comentários --%>
```

- Outros tipos podem ser usados:
  - Comentários Java:

```
<% //comentários   %>
<% /* comentários */   %>
```

  - Comentários HTML:

```
<!-- comentários -->
```

Diretivas

- São executadas em tempo de tradução e alteram características do servlet gerado.
- Existem três diretivas:
  - **page**: permite a execução de diversas tarefas.
  - **include**: inclusão do conteúdo de outra página.
  - **taglib**: uso de bibliotecas de tags customizadas.
- Sintaxe: <%@ ***diretiva*** *{atributo="valor"}* %>
- Xml: *<jsp:directive.**diretiva**{atributo="valor"}/>*

Diretiva Page

DR3

- Importação de pacotes:

```
<%@ page import="java.util.*"%>
```

  - Não precisamos importar **java.lang.***, **javax.servlet.***, **javax.servlet.http.*** e **javax.servlet.jsp.***

- Tipo do conteúdo:

```
<%@ page contentType="text/plain"%>
```

  - O valor default desse atributo é text/html.

Exemplo

DR3

- Inclusão do atributo import para facilitar a programação:

```
<%@ page import="java.util.*,java.text.*" %>
<b>Data:</b> <%= formatarData(new Date()) %><br>
...
<%!
private String formatarData(Date data)
{
    DateFormat df = DateFormat.getDateInstance(
  DateFormat.FULL);

    return df.format(data);
}
%>
```

Objetos predefinidos

DR3

- Existem vários objetos predefinidos que podem ser utilizados dentro de um scriplet JSP:
  - **request** → requisição que iniciou o serviço.
  - **response** → resposta à requisição.
  - **out** → objeto que escreve no fluxo de saída.
  - **page** → sinônimo de *this* (se a linguagem é Java).
  - **session** → objeto session criado para o cliente.
  - **application** → contexto do servlet obtido da configuração.
  - **config** → objeto *ServletConfig* para o JSP.
  - **pageContext** → contexto da página para o JSP.
  - **exception** → objeto de erro.

DR3

Biblioteca de Tags JSP

Criando e Acessando POJO

DR3

- Para carregar um POJO, usa-se:

```
<jsp:useBean id="nomeBean"
             scope="escopoBean"
             class="classeBean" />
```

- Essa tag cria um objeto de nome “id” da classe “class” com o escopo “scope”.
- A criação ocorre apenas se o objeto não existir como atributo do escopo definido por “scope”.
- O escopo pode assumir os valores: page (página atual), **request** (requisição), session (sessão do usuário) e application (contexto da aplicação).

Recuperando Propriedades

DR3

- Para recuperar uma propriedade do Bean:

```
<jsp:getProperty name="nomeBean"
                 property="nomeProp" />
```

- Essa tag retorna a propriedade “property” do bean cujo nome é “name”.
- A tag é convertida para o seguinte código Java:

```
nomeBean.getNomeProp();
```

Exemplo

DR3

- Um servlet pode ter colocado um objeto da conta-corrente na requisição antes de chegar a este JSP:

```
<jsp:useBean id="cc"
             scope="request"
             class="br.edu.infnet.contas.ContaCorrente"
  />

Nome : <jsp:getProperty name="cc"
  property="titular"/>
Saldo: <jsp:getProperty name="cc" property="saldo" />
```

Alterando Propriedades

DR3

- Para definir uma propriedade do Bean, usamos:

```
<jsp:setProperty name="nomeBean"
                 property="nomeProp"
                    value="valor" | param = "parâmetro" />
```

- Essa tag altera o valor da propriedade usando o método set do Bean:

```
nomeBean.setNomeProp(valor);
```

ou

```
nomeBean.setNomeProp(request.getParameter(parâmetro));
```

- O bean alterado continua disponível no mesmo escopo onde fora definido.

Inicializando o JavaBean

DR3

- Se a tag **setProperty** for usada dentro do corpo da tag **useBean**, só será executada se o bean não existir no escopo e tiver que ser criado.

- Exemplo:

```
<jsp:useBean id="cc" scope="request"
             class="br.edu.infnet.contas.ContaCorrente">
   <jsp:setProperty name="cc“ property="titular“
                                        param=“nome”/>
   <jsp:setProperty name="cc" property="saldo"
                                        value=“100”/>
<jsp:useBean />
```

Redirecionamento

DR3

- Para encaminhar requisições, usamos a ação:

```
<jsp:forward page="url" />
```

- Essa ação é equivalente ao redirecionamento com o forward de **RequestDispatcher.**
- Para passar parâmetros, coloque dentro do forward a seguinte tag:

```
<jsp:param name='nome" value="valor"/>
```

Inclusão de Resposta (Dinâmica)

DR3

- Para incluir a resposta de um servlet ou jsp, deve ser usada a ação:

```
<jsp:include page="nomeArquivo" />
```

- Essa ação é equivalente à inclusão com o include de **RequestDispatcher.**
- A inclusão ocorre em **tempo de execução.**
- Parâmetros podem ser passados com a ação:

```
<jsp:param/>
```

Inclusão de Conteúdo (Estática)

- Para incluir o conteúdo de um arquivo estático (html, txt, xml etc.) deve ser usada a diretiva include:

```
<%@ include file="nomeArq" %>
```

- Ela deve ser colocada no lugar exato onde o conteúdo do arquivo deve ser incluído.
- O conteúdo do arquivo é inserido no jsp original **antes da tradução**.

Exemplo

DR3

- Criação de um cabeçalho padrão para todas as páginas.
- Arquivo topo.jsp:

```
<div align=center>
    <h2>Banco Money</h2>
    <hr width=40%>
    <i>O melhor lugar para o seu dinheiro.</i>
</div>
```

- Qualquer arquivo que use o cabeçalho:

```
<%@ include file="topo.jspf" %>
```

DR3

Exercício

- Construir uma aplicação simples com um formulário da entidade Contato, com nome, email e fone.

- Construir um servlet que obtenha os dados passados pelo formulário, valide os dados conforme as regras abaixo e redirecione de volta para o formulário:
  - nome → obrigatório.
  - email → obrigatório.
  - fone → obrigatório, numérico.

- Quando redirecionar, exibir uma mensagem de “Sucesso” ou uma lista de mensagens de erro.