JavaServer Pages (JSP)

- **Objetivo**: usar HTML de forma direta com uso de Java;
- **Comparação Javascript X Servlets X JSP:**
  1) **Javascript**: client-side, necessita suporte e configuração no browser. Cerca de 10% de toda Internet tem Javascript bloqueado no browser por razões de segurança;
  2) **Servlets e JSP**: server-side. Usuário não necessita nada a mais para executar, portanto, rodam em qualquer browser;
- **JSP**: arquivos com extensão jsp, baseado em HTML onde trechos Java são escritos entre <% e %> (esses trechos são chamados *scriplets)*. Arquivos JSP são compilados por um compilador JSP embutido no Tomcat que transformar o arquivo JSP em um servlet.

  Exemplo de um trecho de código Java em um arquivo jsp:

```
<%
	String msg = "Olá Mundo!";
	out.println(msg);
%>
```

- **Exercício 1:** Altere o projeto "aula5" feito na última aula. Modifique o arquivo "index.html" acrescentando o seguinte script:

```
<script type="application/javascript">
	// Exemplo de código em JavaScript
	document.write("<br>Este é um javascript<br>");
	for(i=1; i<=4; i++){
		document.write("Meu "+i+"o Código em JavaScript<br>");
	}
	alert("Acabou!");
</script>
```

Exemplo de validação de formulário usando Javascript:
http://www.w3schools.com/js/js_form_validation.asp

- **Exercício 2:** Altere o projeto "aula5" feito na última aula. Crie um arquivo jsp nomeado "primeirojsp.jsp" com o seguinte conteúdo:

```
<html>
	<head>
		<title>Primeiro JSP</title>
	</head>
	<body>
		<%
			String msg = "Olá Mundo!";
			out.println(msg);
		%>

		<%="Outra mensagem!" %>

		<!-- comentário em html -->
		<%-- comentário em jsp --%>
	</body>
</html>
```

Não esqueça: coloque um link em "admin.html" para o arquivo "primeirojsp.jsp".

- **Exercício 3:** Altere o projeto "aula5", criando um arquivo jsp para listar os alunos. Nomeie o arquivo como "listajsp.jsp". Compare o arquivo jsp com o servlet que você já fez para a mesma função.

```
<body>
	<%
		AlunoDAO dao = new AlunoDAO();
		List<Aluno> alunos = dao.getLista();

		for ( Aluno aluno : alunos){
			out.println("Nome: "+aluno.getNome()+" Email:
			"+aluno.getEmail()+" Endereço: "+
			aluno.getEndereco()+"<br>");
		}
	%>
</body>
```

Não esqueça de adicionar os *imports* e de reiniciar o Tomcat antes de checar as alterações:

```
<%@ page import="java.util.*, br.edu.ufabc.prograd.dao.*,
br.edu.ufabc.prograd.modelo.*"%>
```

- **Vantagem do uso de javabeans**: a vantagem de usarmos Javabeans é que, em jsp, podemos instanciar classes se o construtor da classe é público e sem argumentos. Além disso, a Expression Language entende automaticamente que ${aluno.nome} significa aluno.getNome() e, por isso, é importante obedecer à convenção de Javabeans (observe que o parâmetro "nome" em ${aluno.nome} está em letra minúscula). Diversas outras ferramentas também estão baseadas em Javabeans tais como Hibernate, Structs, VRaptor, JSF, EJB, etc. Assim, em nosso projeto, podemos instanciar dentro do HTML um objeto do tipo aluno como segue abaixo (coloque esse código dentro das tags <body></body> do HTML).

```
<jsp:useBean id="aluno" class="br.edu.ufabc.prograd.modelo.Aluno" />
Estou imprimindo: ${aluno.nome}
```

TAGLIBS

- **problema com jsp**: abuso de código Java dentro dos arquivos jsp;
- **solução proposta pela Sun**: conjunto de tags (*tag library* ou *taglib*) para substituir trechos de código;
- **Taglib sugerido pela Sun:** JSTL (JavaServer Pages Standard Tag Library);
- Baixe as bibliotecas do JSTL no site da disciplina (disponível originalmente em: http://jstl.java.net/) e copie para a pasta lib;
- Exercício 4: Altere o projeto "`aula5`", criando um arquivo `jsp` para listar os alunos. Nomeie o arquivo como "`listataglib.jsp`". Importe a taglib JSTL acrescentando no início do arquivo o seguinte:

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

- O prefixo substitui termos que escrever toda a `uri` cada vez que usarmos uma tag da biblioteca. Acrescente o código para listar os alunos:

```
<jsp:useBean id="dao" class="br.edu.ufabc.prograd.dao.AlunoDAO" />
<c:forEach var="aluno" items="${dao.lista}">
      Nome: ${aluno.nome}<br>
      Email: ${aluno.email}<br>
      Endereço: ${aluno.endereco}<br><br>
</c:forEach>
```

- **Exercício 5:** Altere o arquivo "`listataglib.jsp`", colocando as informações da listagem de alunos em uma tabela HTML com os títulos das colunas em negrito e com a cor das linhas alternando entre azul e amarelo.
- **Exercício 6**: No arquivo "`listataglib.jsp`", faça uma validação para o e-mail. Se o campo de e-mail for não vazio, coloque um link para abrir o software de e-mail padrão do computador para enviar um e-mail a esse aluno:

```
<c:if test="${not empty aluno.email}">
      <a href="mailto:${aluno.email}">${aluno.email}</a><br>
</c:if>
```