

Desenvolvendo com Java para Web

Fernando Boaglio



Grupo de Usuários Java



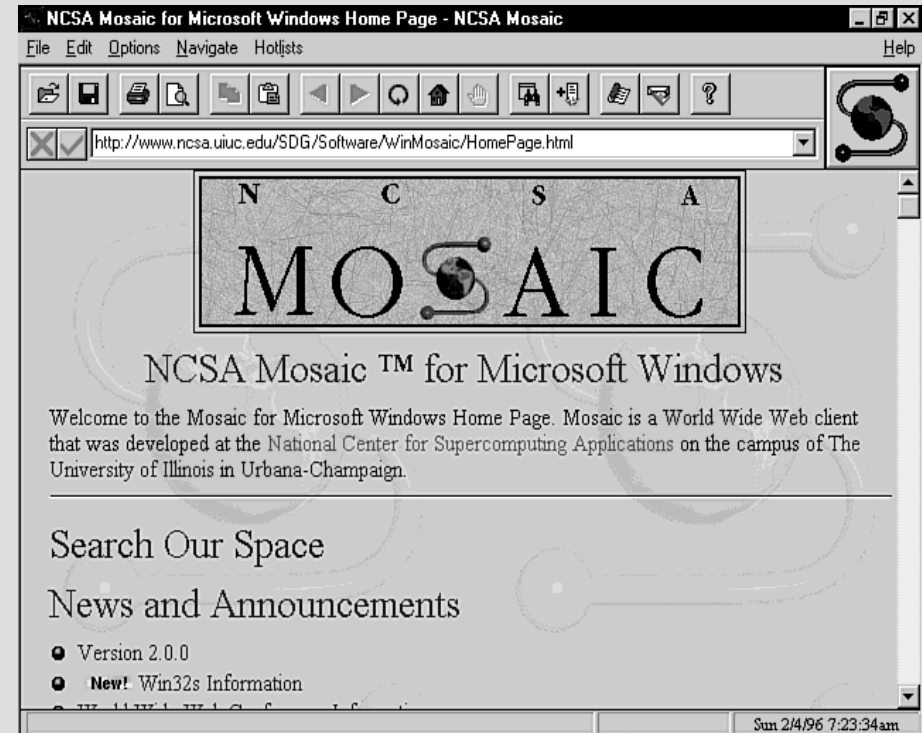
Agenda

- A evolução das aplicações web
- Java: do desktop para a Web
- O que saber para desenvolver ?
- Nossa primeira aplicação
- Como usar os conceitos de web
- Nossa segunda aplicação
- Funcionalidades
- Produtividade na programação
- Comunidade virtual



A evolução das aplicações

- Internet estática:
início dos anos 90
- páginas apenas de
divulgação de documentos
- 1992: padrão HTML 1.0
- Internet ainda não era
popular



A evolução das aplicações

Internet dinâmica:

- páginas para sites comerciais
- Internet tornando-se popular
- surgem diversas linguagens de geração de conteúdo dinâmico
- padrões HTML:
 - 1994: 2.0 (formulários)
 - 1996: 3.2 (applets Java/JavaScript)
 - 1997: 4.0 (páginas de estilo-CSS)
 - 2000: XHTML 1.0 (HTML4 + XML)
 - 2004: em desenvolvimento: XHTML 2.0



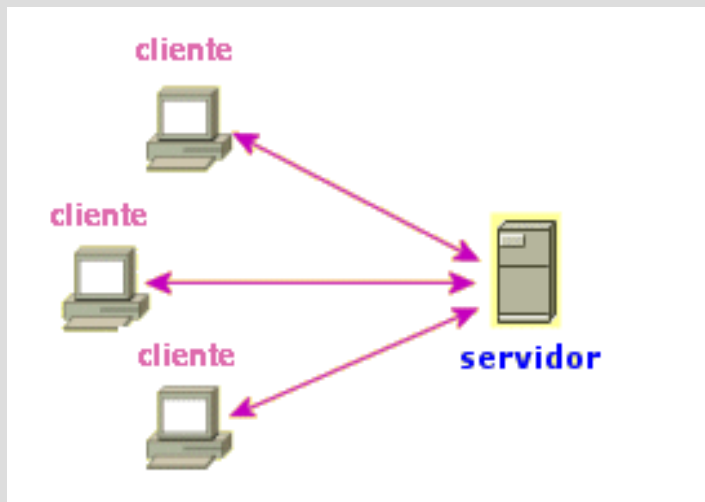
Agenda

- A evolução das aplicações web
- **Java: do desktop para a Web**
- O que saber para desenvolver ?
- Nossa primeira aplicação
- Como usar os conceitos de web
- Nossa segunda aplicação
- Funcionalidades
- Produtividade na programação
- Comunidade virtual



Java: do desktop para a Web

- 1995 - criada a tecnologia Java
- 1996 - lançado o JDK 1.0
- 1997 - lançado o JDK 1.1(JDBC,JavaBeans)

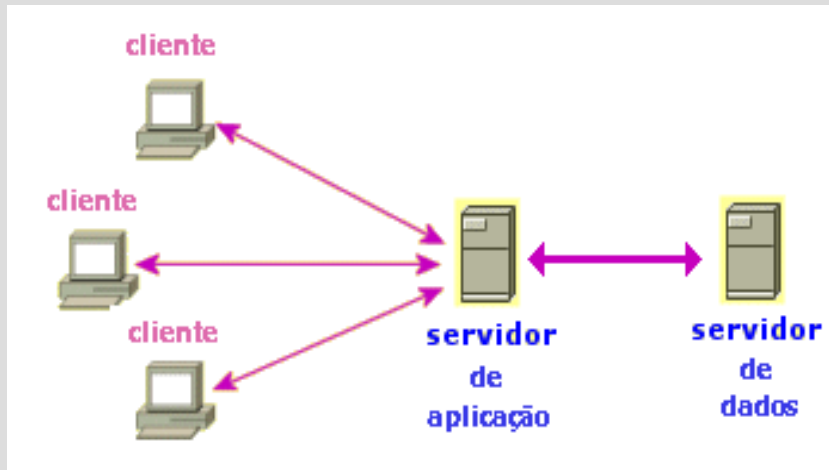


- Aplicações desenvolvidas em Java acessam bancos de dados via JDBC
- Web browsers acessam sites com applets Java



Java: do desktop para a Web

- 1997 – criado o Servlet Dev. Kit
- 1998 - lançado o Java 2 (Java 1.2)
- 1999 – Sun divide sua tecnologia em J2SE, J2ME e J2EE



J2ME – Java 2 Micro Edition

J2SE – Java 2 Standard Edition

J2EE – Java 2 Enterprise Edition



Java: do desktop para a Web

- 1999 - lançado o JSP

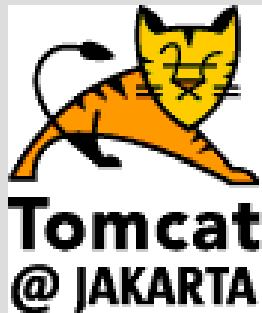
principais motivos:

- facilitar o desenvolvimento web (Servlet é considerado complicado para novatos)
- trazer desenvolvedores de outras linguagens de script (PHP, ASP)



Java: do desktop para a Web

- 2000 – lançado o Java 1.3
- 2002 – lançado o Java 1.4
- 2003 – lançado o JSP 2.0
- 2004 – lançada versão Beta do Java 1.5



Agenda

- A evolução das aplicações web
- Java: do desktop para a Web
- O que saber para desenvolver ?
- Nossa primeira aplicação
- Como usar os conceitos de web
- Nossa segunda aplicação
- Funcionalidades
- Produtividade na programação
- Comunidade virtual



O que saber para desenvolver?

Fundamentos do HTML

- Como funcionam as tags
- Como trabalhar com formulários
- Como trabalhar com imagens
- Protocolos: HTTP, FTP



O que saber para desenvolver?

Fundamentos do Java

Não desenvolva em JSP sem conhecer Java!

- Conheça todos os tipos de dados
- Como trabalhar com vetores
- Como trabalhar com classes, métodos, herança e referência
- Conheça as APIs e não reinvente a roda!



O que saber para desenvolver?

Como o JSP funciona?

1. **Cliente** faz a primeira requisição via *browser*
2. **Servidor web** encaminha para a **Engine JSP**
3. **Engine JSP** compila a página e transforma em servlet
4. O servlet é executado
5. O resultado é enviado de volta ao *browser* do cliente



O que saber para desenvolver?

Instalação do Ambiente de Desenvolvimento

- Instalação do **J2SE**
- Instalação do servidor web **Apache Tomcat**
- Uso de uma ferramenta: **Eclipse**



Agenda

- A evolução das aplicações web
- Java: do desktop para a Web
- O que saber para desenvolver ?
- **Nossa primeira aplicação**
- Como usar os conceitos de web
- Nossa segunda aplicação
- Funcionalidades
- Produtividade na programação
- Comunidade virtual



Nossa primeira aplicação

Uma página inicial com um contador



Nossa primeira aplicação

Conceitos importantes

diretiva - especifica instruções da página para a JSP Engine (taglib, include, page) durante a compilação

declaração – declara e define variáveis e métodos que podem ser usados na página JSP

scriptlet – são trechos de código Java imbutidos na página JSP

expressão – faz um cálculo e exibe um resultado

action – especifica instruções da página para a JSP Engine (forward, useBean, plugin) na durante a requisição



Nossa primeira aplicação

Conceitos importantes

```
<%- - Agenda Pessoal - -%> ← comentário  
<html>  
<center>  
<h1> Minha Agenda Pessoal </h1>  
</center>  
<BR>  
<%@ page language = "java" %> ← diretiva  
<%! int contador = 0; %> ← declaração  
<% contador++; %> ← scriptlet  
<hr>  
Acessos: <%= contador %> ← expressão  
</html>
```



Nossa primeira aplicação

Dividindo a aplicação

cabeçalho

conteúdo principal

rodapé



Nossa primeira aplicação

Minha nova agenda:

`cabecalho.jsp`

`principal.jsp`

`rodape.jsp`



Nossa primeira aplicação

cabecalho.jsp

```
<%-- Agenda Pessoal --%>  
<html>  
<body bgcolor="Silver">  
<center>  
<h1>Minha Agenda Pessoal</h1>  
</center>  
<hr>
```



Nossa primeira aplicação

rodape.jsp

```
<BR>  
<hr>  
<center>  
  <a href="principal.jsp">Inicio</a>  
</center>  
</body>  
</html>
```



Nossa primeira aplicação

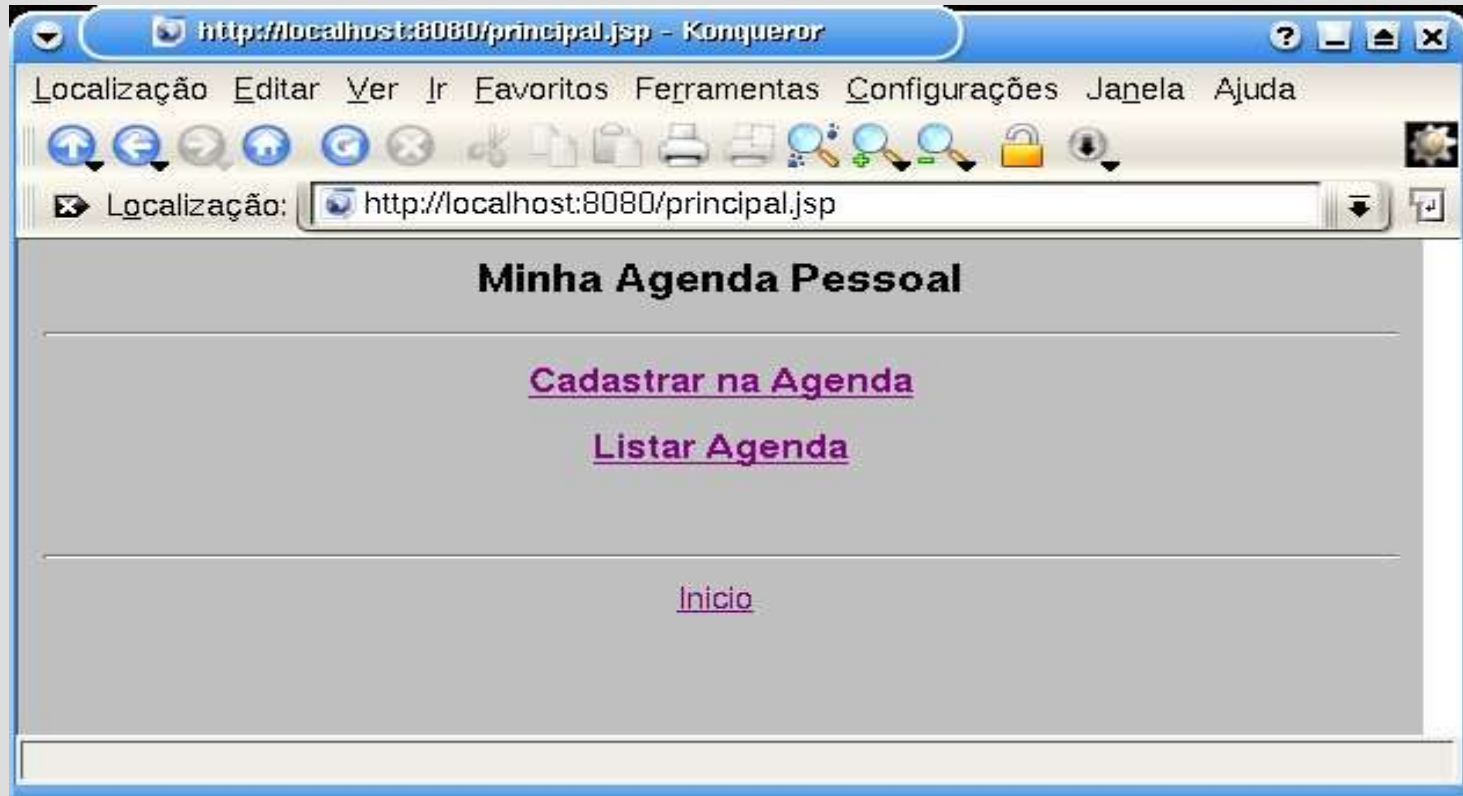
principal.jsp

```
<%@ include file="cabecalho.jsp" %>
<center>
  <h2>
    <a href="novo.jsp?opcao=incluir">Cadastrar na Agenda</a>
  </h2>
  <h2>
    <a href="novo.jsp?opcao=listar">Listar Agenda</a>
  </h2>
</center>
<%@ include file="rodape.jsp" %>
```



Nossa primeira aplicação

Resultado:



Agenda

- A evolução das aplicações web
- Java: do desktop para a Web
- O que saber para desenvolver ?
- Nossa primeira aplicação
- Como usar os conceitos de web
- Nossa segunda aplicação
- Funcionalidades
- Produtividade na programação
- Comunidade virtual



Como usar os conceitos de web

- Operações de busca e cadastro na internet são tratadas com formulários
- As variáveis de requisição são capturadas usando o método *request.getParameter*
- O request é uma variável do tipo *javax.servlet.http.HttpServletRequest*

Seu IP é `<%= request.getRemoteAddr() %>`



Agenda

- A evolução das aplicações web
- Java: do desktop para a Web
- O que saber para desenvolver ?
- Nossa primeira aplicação
- Como usar os conceitos de web
- **Nossa segunda aplicação**
- Funcionalidades
- Produtividade na programação
- Comunidade virtual



Nossa segunda aplicação

- Uma agenda que contém uma lista de contatos num HashMap
- A listagem e cadastro de contatos é feita na mesma página: [novo.jsp](#)

```
<%@ include file="cabecalho.jsp" %>
```

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

```
<%! Map contatos = new HashMap(); %>
```



Nossa segunda aplicação

```
<% if (request.getParameter("opcao").equals("listar"))
{ %>
<center><BR><h2>Contatos
cadastrados</h2><BR><table>
  <% Iterator it = contatos.keySet().iterator();
    while (it.hasNext()) {
      Object chave = it.next(); %>
      <TR><TD><B>Nome:</B> <%= chave %>
      <B>Email:</B><%= contatos.get(chave) %></TD></TR>
    } %>
</table> <BR>
  Total: <%=contatos.size() %>
</center>
```

<HR>



Nossa segunda aplicação



Nossa segunda aplicação

```
<%  
}  
else  
{  
%>  
<center><h2>Novo Contato</h2>  
<BR>  
<form action="novo.jsp" method="POST">  
  Nome:<input type="text" name="nome"><BR>  
  E-mail:<input type="text" name="email"><BR>  
  <input type="submit" value="Cadastrar">  
  <input type="hidden" name="opcao" value="">  
</form>
```

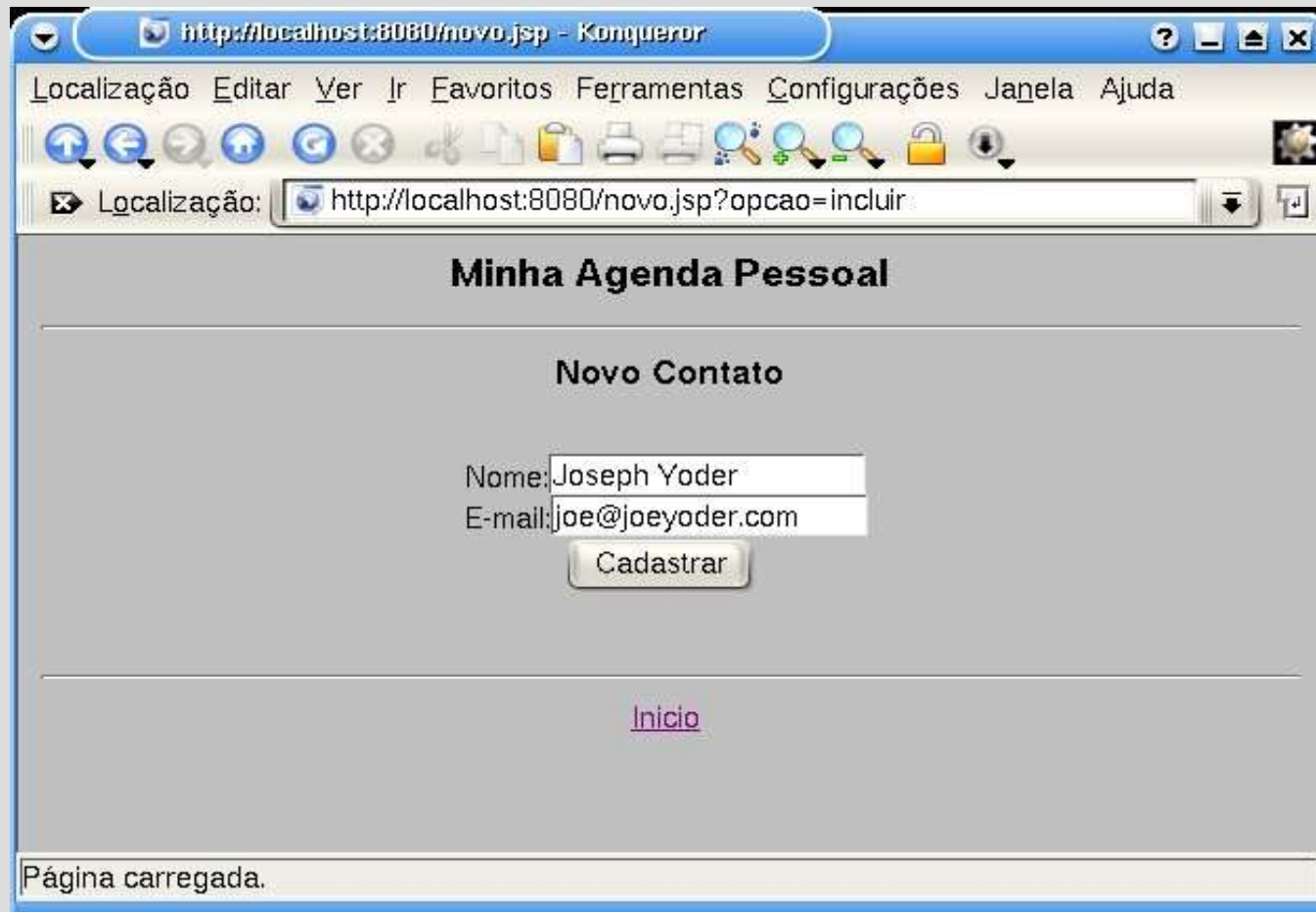


Nossa segunda aplicação

```
<%  
String nome=request.getParameter("nome");  
String email=request.getParameter("email");  
// cadastra o valor  
if (nome!=null) {  
    contatos.put(nome,email);  
    out.println("<BR><b>Contato cadastrado!</b><BR>");  
}  
%>  
</center><%  
} %>  
<%@ include file="rodape.jsp" %>
```



Nossa segunda aplicação



Agenda

- A evolução das aplicações web
- Java: do desktop para a Web
- O que saber para desenvolver ?
- Nossa primeira aplicação
- Como usar os conceitos de web
- Nossa segunda aplicação
- **Funcionalidades**
- Produtividade na programação
- Comunidade virtual



Funcionalidades

Use JavaBeans!

- `<jsp:useBean>` - declara o uso de JavaBean na página JSP
- `<jsp:setProperty>` - atribui novos valores ao componente
- `<jsp:getProperty>` - retorna os valores do componente

Os componentes são objetos que carregam as informações que serão usadas na suas regras de negócio.

Facilita a manipulação de valores usados nos parâmetros dos formulários e a persistência de valores na aplicação



Funcionalidades

Como implementar essas funcionalidades?

- SQL ? E ler um BLOB ?
- Gerar PDF?
- Mandar e-mail ?
- Gerar telas para WAP?
- Fazer log da sua aplicação?
- Gerar gráficos dinâmicos?
- Usar funções de manipular datas ou Strings?



Funcionalidades

Use taglibs!!!!

- É preferível na sua aplicação sempre usar taglib e nunca Scriptlet
- Taglibs como JSTL padronizam seu código!

```
<%@ taglib uri="/WEB-INF/tld/c.tld" prefix="c" %>  
<%-- exemplo de um IF --%>  
<c:if test="$ {param.usuario == 'convidado'}">  
    <B>Não perca o minicurso de JSTL!</B>  
</c:if>
```



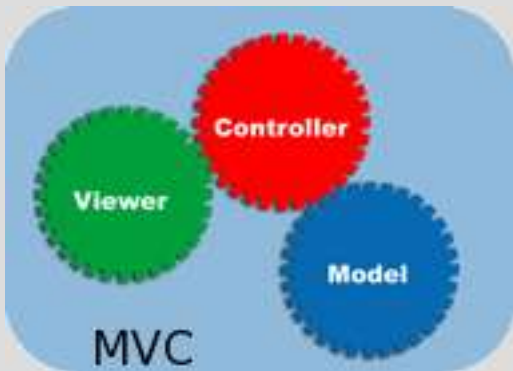
Agenda

- A evolução das aplicações web
- Java: do desktop para a Web
- O que saber para desenvolver ?
- Nossa primeira aplicação
- Como usar os conceitos de web
- Nossa segunda aplicação
- Funcionalidades
- **Produtividade na programação**
- Comunidade virtual



Produtividade na programação

Diversos frameworks geram suas soluções usando a arquitetura MVC, utilizando a camada de visão o JSP: *Struts*, *WebWork*, *Spring*



Model – faz as regras de negócio e busca os dados

Viewer – lógica de apresentação

Controller – controla o fluxo dos dados entre o Model e o Viewer



Agenda

- A evolução das aplicações web
- Java: do desktop para a Web
- O que saber para desenvolver ?
- Nossa primeira aplicação
- Como usar os conceitos de web
- Nossa segunda aplicação
- Funcionalidades
- Produtividade na programação
- **Comunidade virtual**



Comunidade virtual

- **Diversos Grupos de Usuários :**
GUJ, SouJava, etc.

- **Sites de JSP:**

<http://www.jspbrasil.com.br>

<http://www.jsptut.com>

<http://www.servletsuite.com/jsp.htm>

<http://jsptags.com>

- **Literatura:**

SCWCD Exam Study Kit — Hanumant Deshmukh e Jignesh Malavia
editora Manning (baixe os exercícios do livro de <http://www.manning.com>)



Alguma dúvida ???



Minicurso Java-Web

Obrigado!

Fernando Boaglio

fernando@boaglio.com



Grupo de Usuários Java

