JavaServer Pages (JSP) e Servlets
Parte 2: elementos JSP dinâmicos, MVC, e integração JSP+Servlet

Fernando dos Santos

UDESC
Ibirama

JSP: elementos dinâmicos

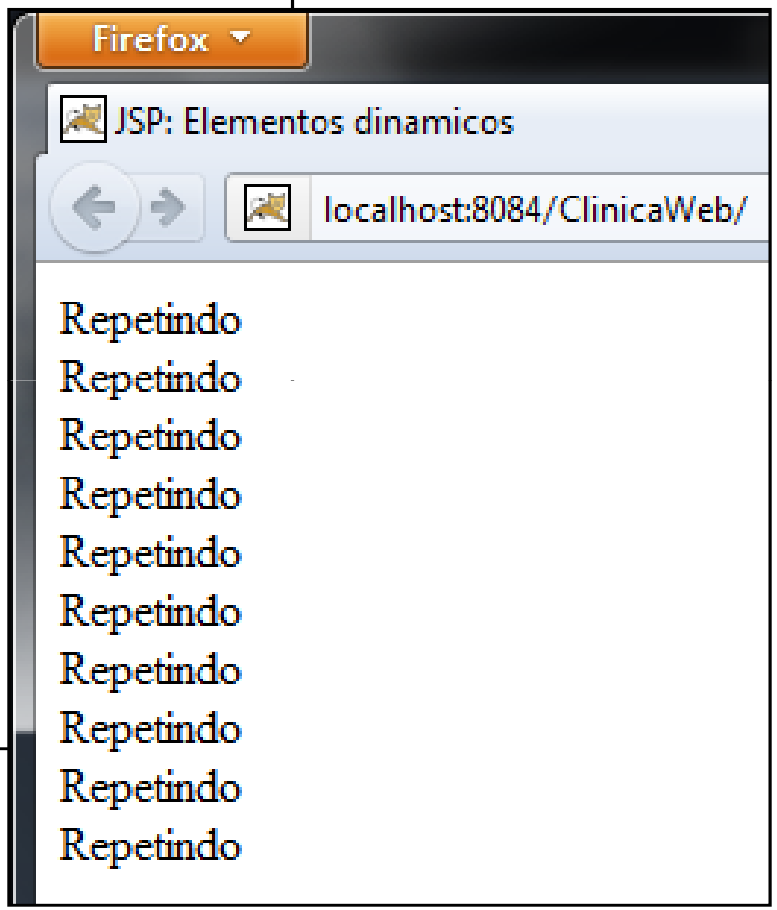
- Em um arquivo JSP, os elementos dinâmicos dinâmicos são trechos de código Java, que constróem o conteúdo HTML dinamicamente.
  - scriptlets
  - expressões
  - diretivas

JavaServer Pages: scriptlets

- Um scriptlet é um bloco de código Java que será executado pelo servidor ao receber a requisição para mostrar uma página JSP.
- Fica entre **<% .... %>**

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <title>JSP: Elementos dinamicos</title>
    </head>
    <body>
        <% for(int i = 0; i < 10; i++) { %>
            Repetindo <br>
        <% } %>
    </body>

</html>
```

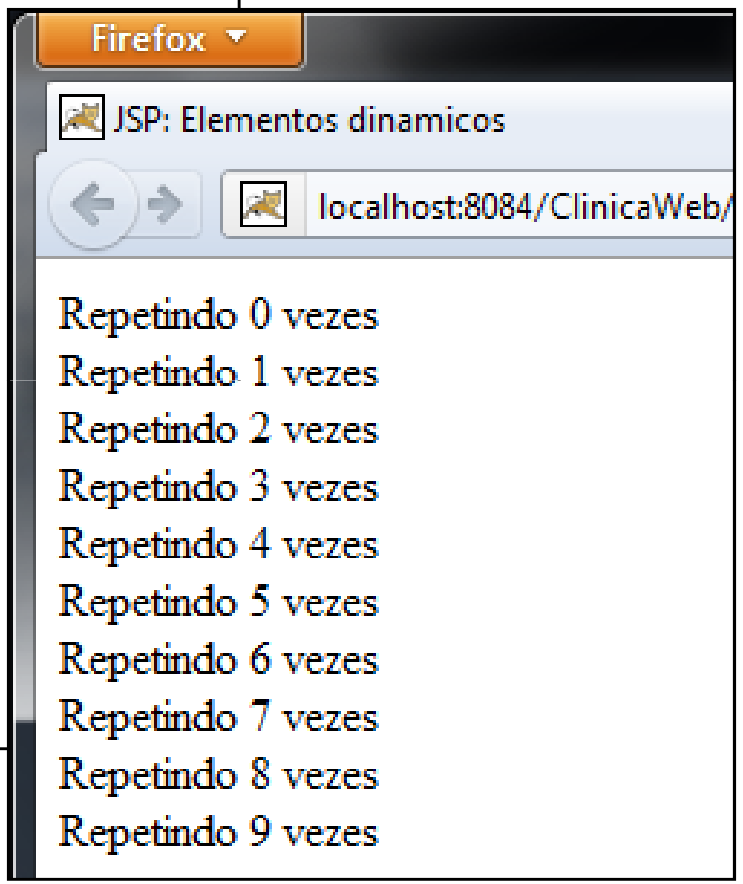


JavaServer Pages: expressões

- Permitem escrever o resultado de uma expressão na página HTML
  - A expressão pode ser uma variável, uma equação, etc.
- Fica entre **<%= .... %>**

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <title>JSP: Elementos dinamicos</title>
    </head>
    <body>
        <% for(int i = 0; i < 10; i++) { %>
            Repetindo <%= i %> vezes <br>
        <% } %>
    </body>

</html>
```



JavaServer Pages: diretivas

- Configurações do JSP:
  - importação de classes, inclusão de arquivos, etc.
- Fica entre <%@ ... %>
- Tipos principais:
  - <%@ page atrib=“valor” %>

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

  - <%@ include file=“nomearquivo.jsp” %>

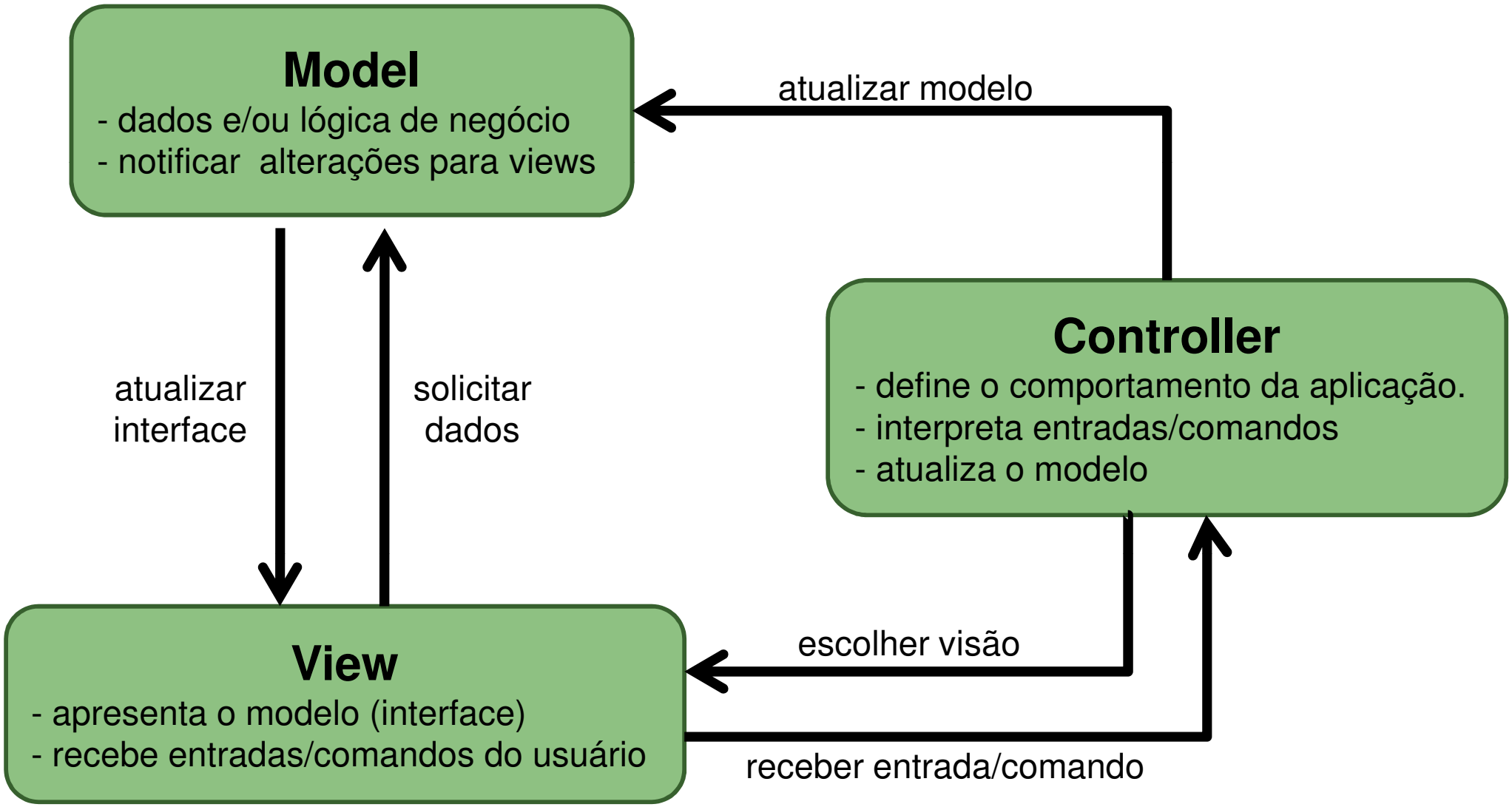
```
<%@include file=“cabecalho.jsp"%>
... corpo da pagina ...
<%@include ifile=“rodape.jsp"%>
```

Padrão Model-View-Controller (MVC)

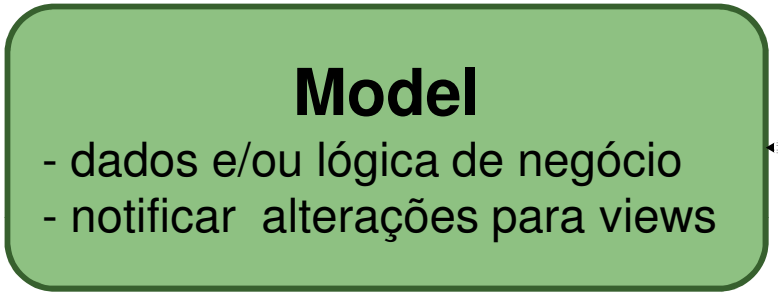
- Em aplicações mal projetadas, um único código fonte concentra a lógica do negócio, a interface e o acesso aos dados.
- Deficiências desta abordagem?
  - precisa alterar o visual da interface... onde mexer no código?
  - precisa alterar uma lógica do negócio... onde mexer no código?

- Separar diferentes responsabilidades:
  - apresentação (telas) → visão
  - dados (entidades) → modelo
  - negócio (regras, rotinas) → controle

Padrão Model-View-Controller (MVC)



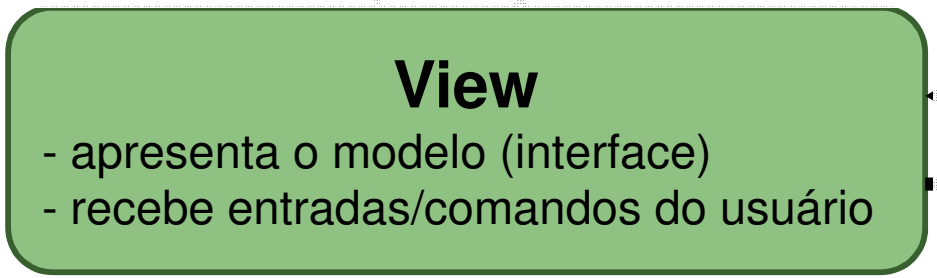
Padrão Model-View-Controller (MVC)



- **Model**
  - Representa dados e/ou lógica de negócio;
  - Seus elementos são acessados pelo controller, para atualizar dados e/ou executar lógica de negócio;
  - Notifica as views quando ocorrem alterações nos dados;
  - Atende as solicitações de dados provenientes das views.

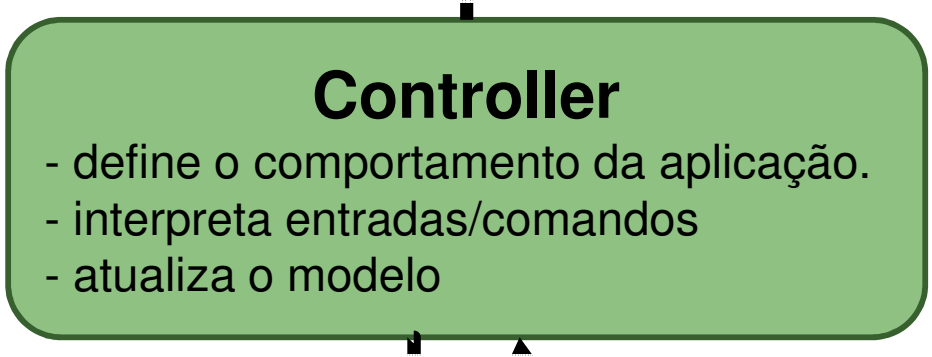
Padrão Model-View-Controller (MVC)

- **View**
  - Interface gráfica para apresentar dados do modelo;
  - Solicita dados do modelo para exibí-los, se necessário.
  - Recebe as entradas e comandos do usuário e as encaminha para o controller tratar.



Padrão Model-View-Controller (MVC)

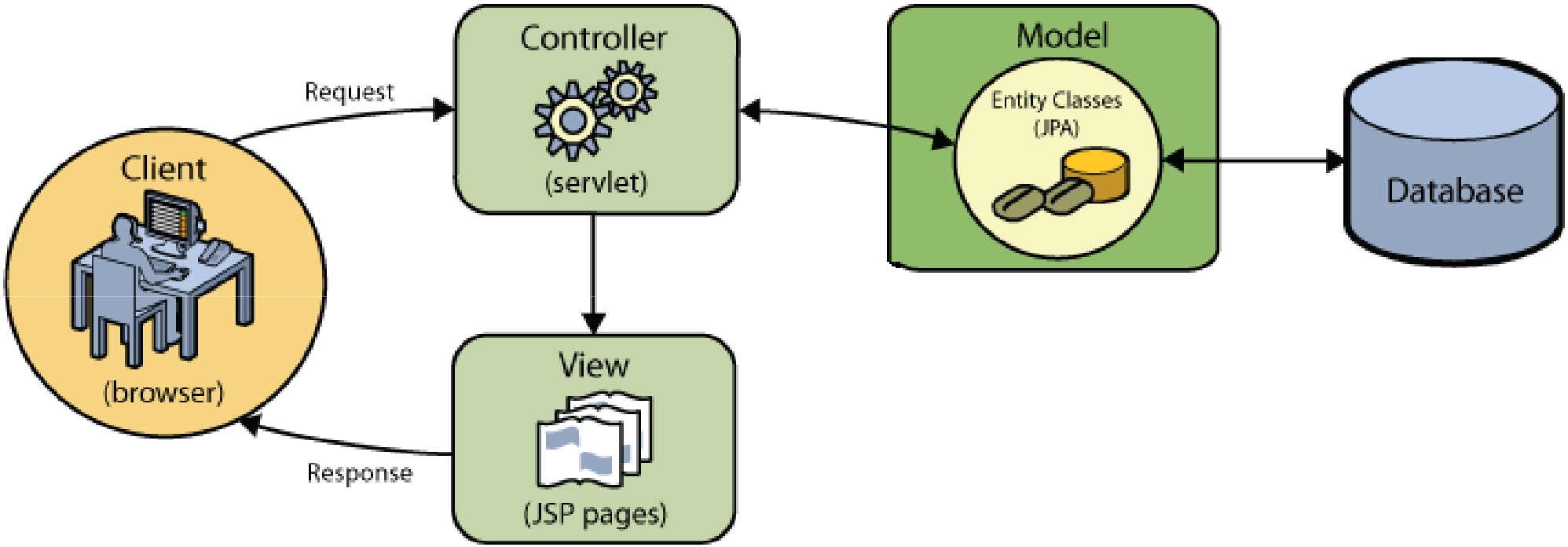
- **Controller**
  - Define o comportamento da aplicação.



  - Processa as entradas e comandos, atualizando os dados do model e gerenciando qual view deve ser apresentada em resposta a cada comando;

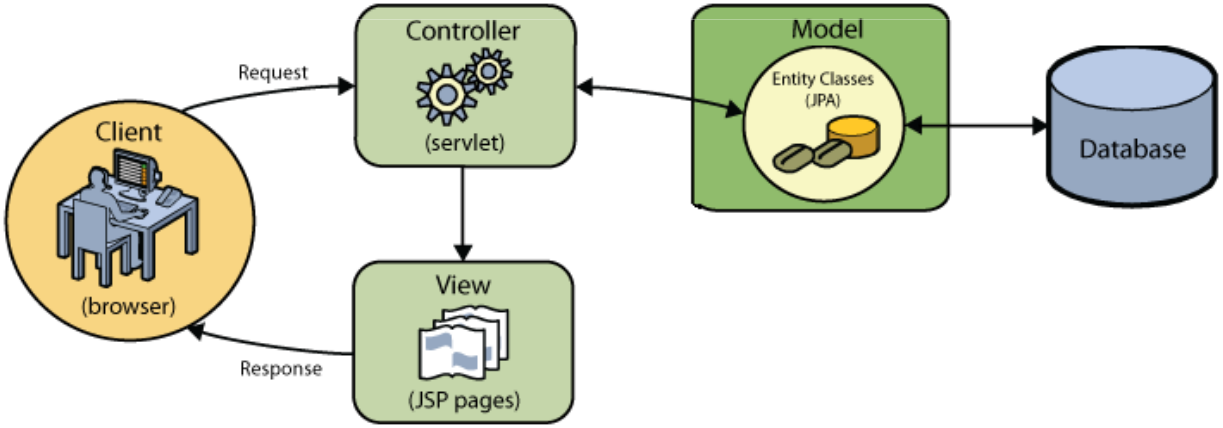
MVC em aplicações WEB Java

- **Controller**: é um Servlet
- **View**: conjunto de páginas JSP
- **Model**: classes de entidades



MVC com JSP e Servlet (1)

- Um dos papéis do servlet (controlador) é gerenciar qual JSP (visão) será apresentada em cada situação.



- Após determinar a próxima visão, o Servlet deve **encaminhar** a navegação para ela. Este encaminhamento leva consigo os objetos request e response, para uso no JSP:

```
RequestDispatcher encaminhador =
                request.getRequestDispatcher("pagina.jsp");
 encaminhador.forward(request, response);
```

MVC com JSP e Servlet (2)

- Antes de encaminhar a navegação, o Servlet pode adicionar parâmetros (**atributos**) no objeto **request**
- Estes parâmetros poderão ser acessados pelo JSP **encaminhado**.

- Para adicionar atributos no **request** pelo Servlet:
  - `request.setAttribute("id", valor);`
    - **id**: String que identifica o atributo
    - **valor**: valor do atributo – pode ser qualquer objeto

- Para recuperar atributos do **request** pelo JSP:
  - `<% Object valor = request.getAttribute("id"); %>`
  - `<%= request.getAttribute("id"); %>`

Exercício (1)

1. ClinicaServlet, após cadastrar o paciente, deve adicionar o objeto de paciente no request, e encaminhar a navegação para a página "confirmacaoCadastroPaciente.jsp".

2. Esta página deve exibir os dados do paciente cadastrado, com uma mensagem "Cadastro realizado com sucesso".

- Implante e teste o projeto
  - cadastre um paciente e verifique a confirmação.

Exercício (2)

- Criar uma página para mostrar os pacientes cadastrados
  - listarPacientes.jsp

- Problema: como fazer a requisição passar pelo controlador, para que possa acessar o banco de dados e recuperar a lista de pacientes?

- Solução:
  - adicionar um padrão de url: **/ClinicaServlet/listarPacientes**
  - no Servlet, verificar qual é o padrão de url recebido:
    - **String action = request.getServletPath();**
    - **if (action.equals("/ClinicaServlet/listarPacientes") { .... }**

Bibliografia

- TODD, N. **JavaServer pages :o guia do desenvolvedor.** Rio de Janeiro: Elsevier: 2003.

- BASHAM, Brian; SIERRA, Kathy; BATES, Bert. **Use a cabeça!**: Servlets & JSP. Rio de Janeiro : Alta Books, c2005. 534 p, il.

- BERGSTEN, Hans. **JavaServer Pages**. Beijing : OReilly, 2001. xviii, 552p, il. (The Java series).