



Estácio

SERVLETS – 1ª Parte

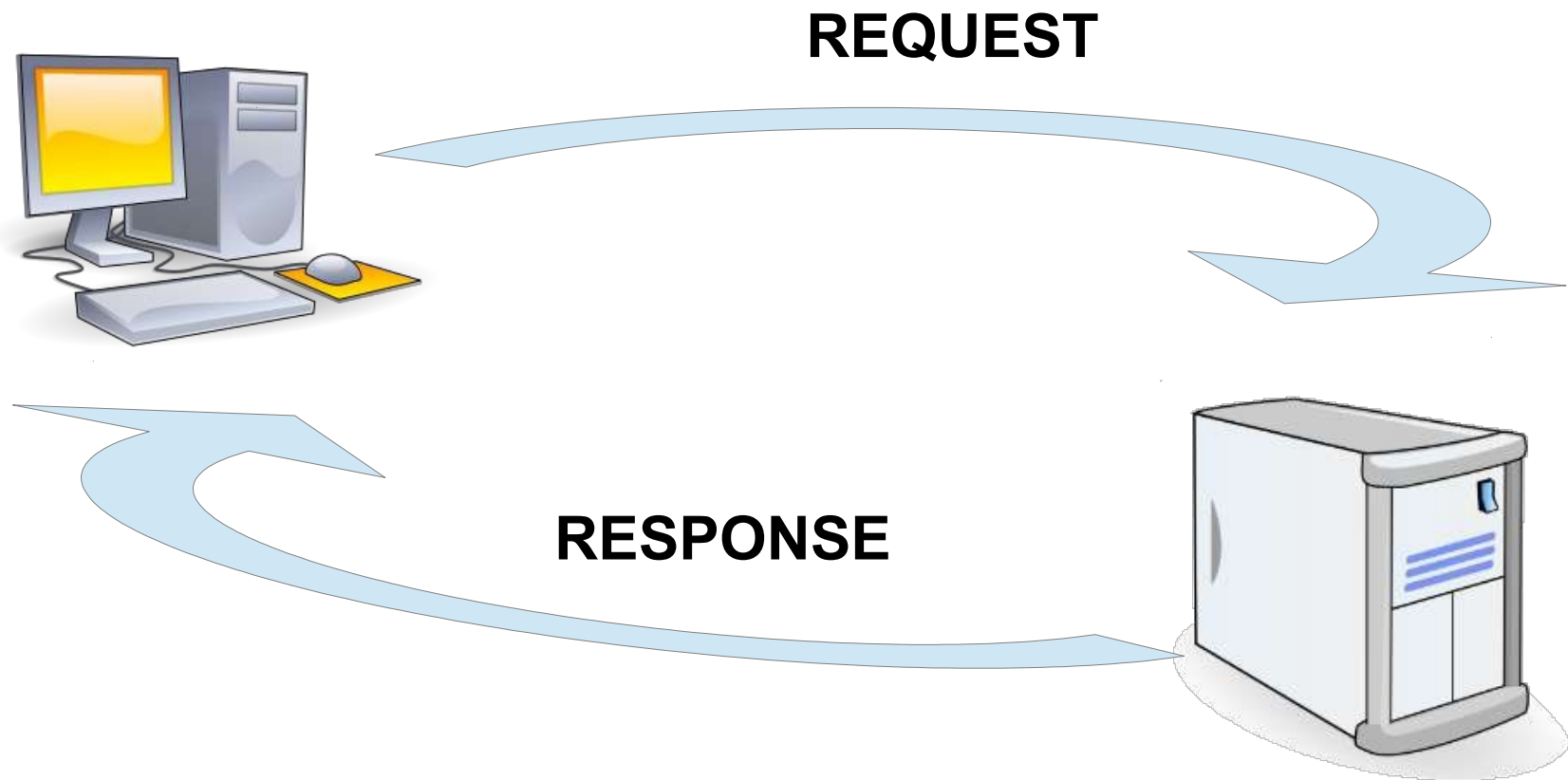


OBJETIVOS

- Compreender a Tecnologia de Servlets
- Criar aplicativos Web Baseados nessa Tecnologia
- Compreender as formas de recepção de dados, bem como o encaminhamento e redirecionamento no fluxo de páginas

ARQUITETURA WEB

- **Arquitetura Web**
 - Na aula passada ...



ARQUITETURA WEB

- Quem é responsável por gerar a request?
 - O computador cliente
 - Navegador
 - Outro programa qualquer que fale o mesmo protocolo
- Quem é responsável por gerar a response?
 - O computador servidor
 - Programa servidor
 - O servidor presta um serviço ao cliente

ARQUITETURA WEB

- O cliente: navegador
 - Requisições são geradas quando:
 - Digitamos uma nova URL
 - Clicamos em um link
 - Enviamos um formulário
- O servidor: aplicativo em computador 24/7
 - Ele recebe requests e devolve responses
 - Apenas quando uma requisição chega ele atua
 - É este tipo de aplicativo que iremos desenvolver!

CONTAINER WEB

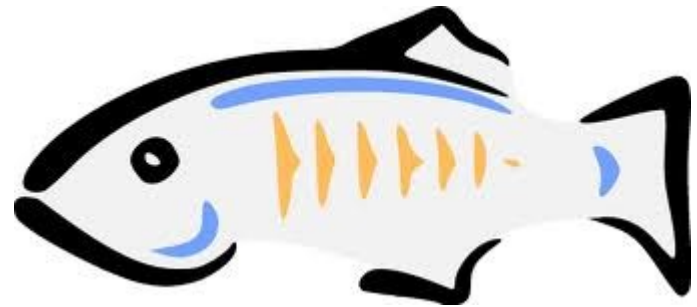
- **Servlets** são as soluções Java para aplicações Web (JEE)
 - Suportam os métodos de requisição HTTP padrão
 - Geram respostas compatíveis com HTTP
 - Interagem com Cookies
- Além dessas tarefas básicas, também
 - Suportam filtros, que podem ser chamados em cascata para tratamento de dados durante a requisição
 - Suportam controle de sessão transparentemente através de cookies ou rescrita de URLs (automática)
- **Para isso funcionar é preciso usar um servidor que suporte as especificações de Servlets e JSP: Web Container**

CONTAINER WEB

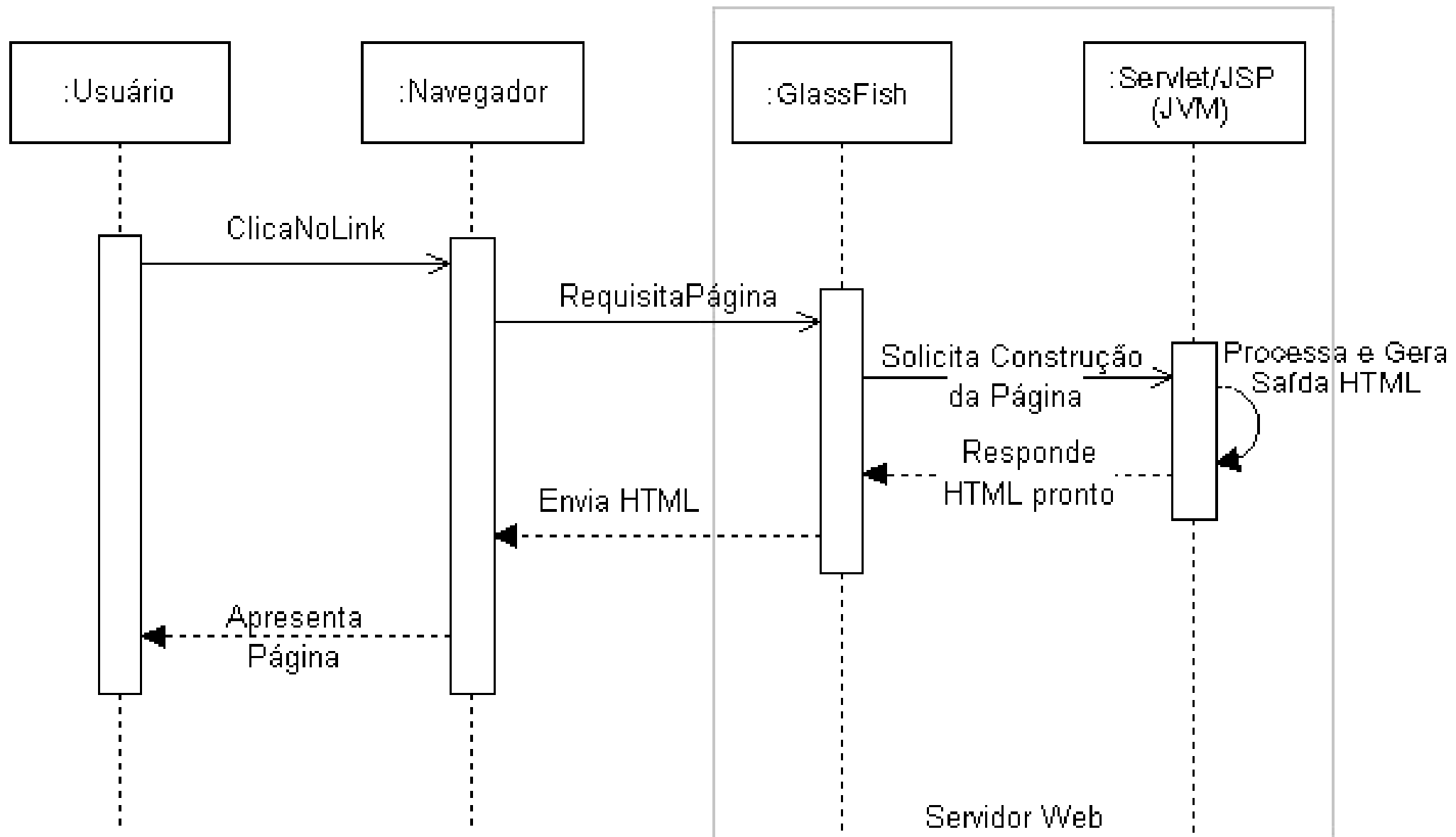
- Servidor de Aplicações - **Funcionamento básico:**
 - Aguarda pelas requests
 - Repassa a request para o servlet correto
 - Recebe a response do servlet
 - Devolve a response para o cliente que enviou a request
- Ou seja, ele é responsável por **gerenciar as conexões**, dentre outras coisas.

CONTAINER WEB

- Existem vários Contentores Java
 - GlassFish
 - TomCat
 - JBOSS
 - Gerônimo
 - Jetty
 - Dentre outros...
- Neste curso usaremos o **GlassFish**
 - É o mais completo
 - É o padrão sugerido pela Oracle



CONTAINER WEB

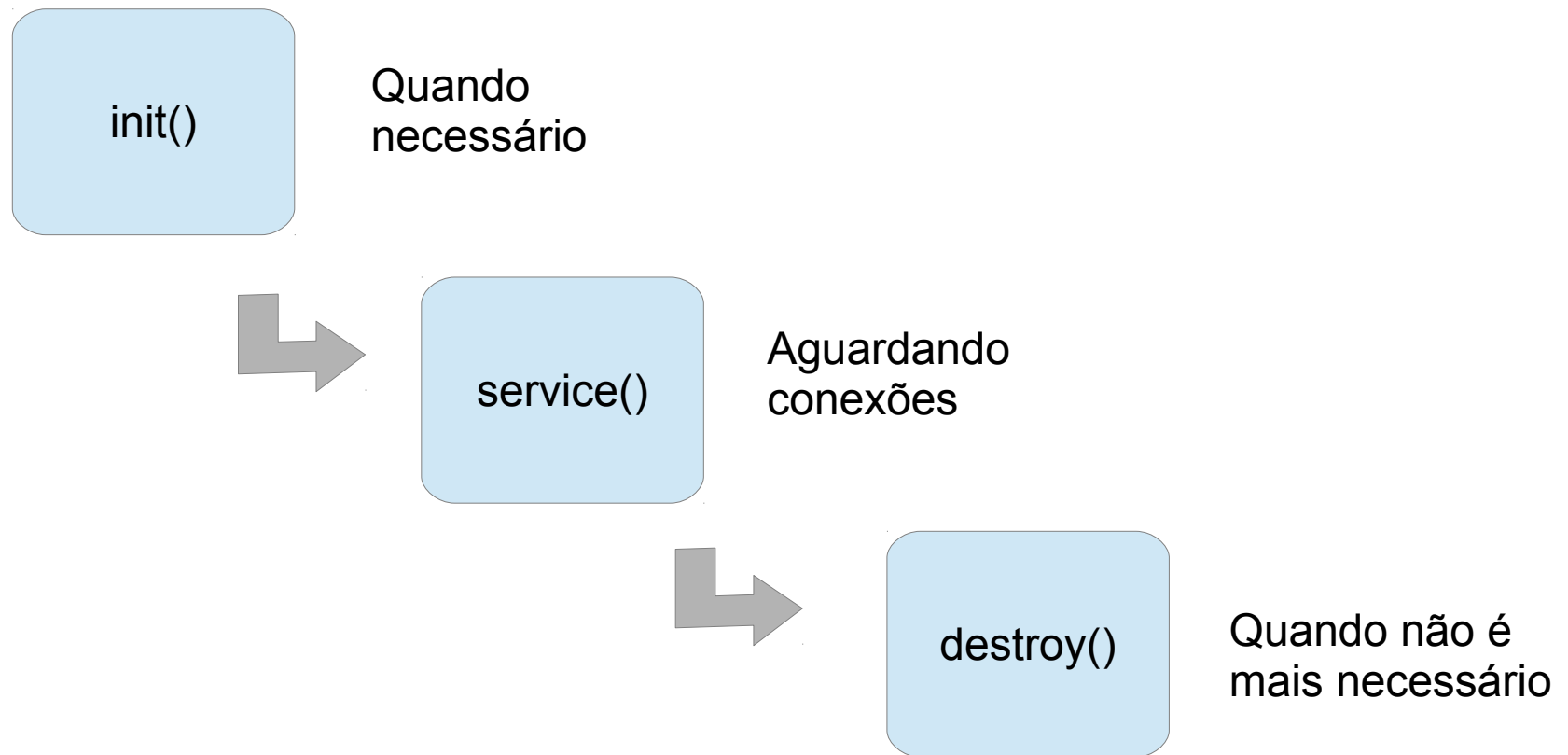


SERVLET

- Mas o servidor de aplicativos não faz apenas isso...
- Dentre outras coisas, ele também cuida do
ciclo de vida do Servlet:
 - Quando necessário, inicia o Servlet (**init**)
 - Coloca este Servlet ativo, aguardando conexões (**service**)
 - Finaliza o Servlet quando não mais necessário (**destroy**)

SERVLET – CICLO DE VIDA

- Resumidamente ...



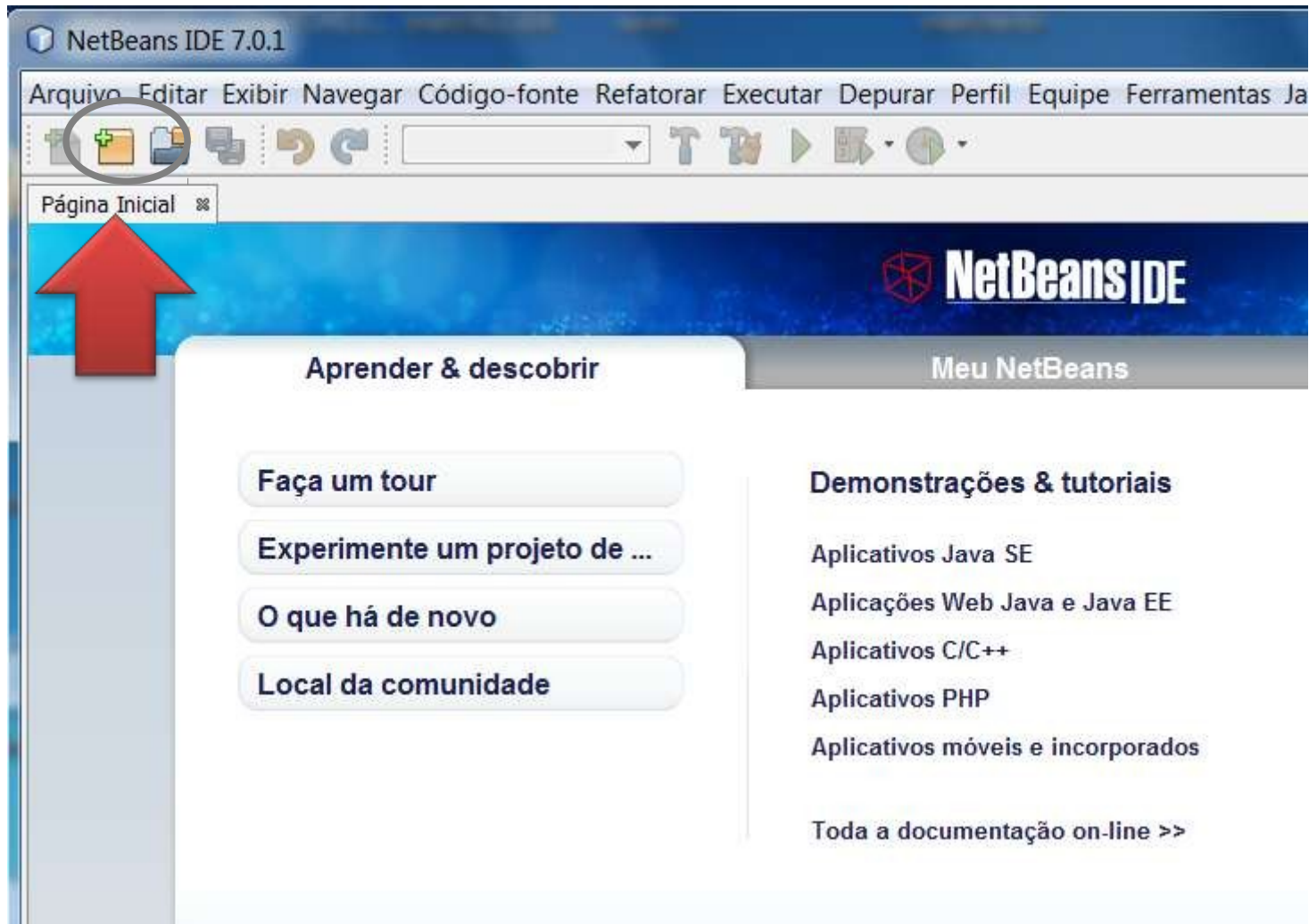
SERVLET

- Servlets são classes Java que têm a capacidade de receber requisições e produzir a resposta, quer seja em HTML ou outro formato de saída
- Os Servlets que iremos manipular estendem a classe `HttpServlet` do pacote `javax.servlet`
 - **HttpServlet** → Servlet específico para o HTTP
 - **javax.servlet** → pacote onde encontramos todas as classes/interfaces para trabalharmos com servlets.

SERVLETS NO NETBEANS

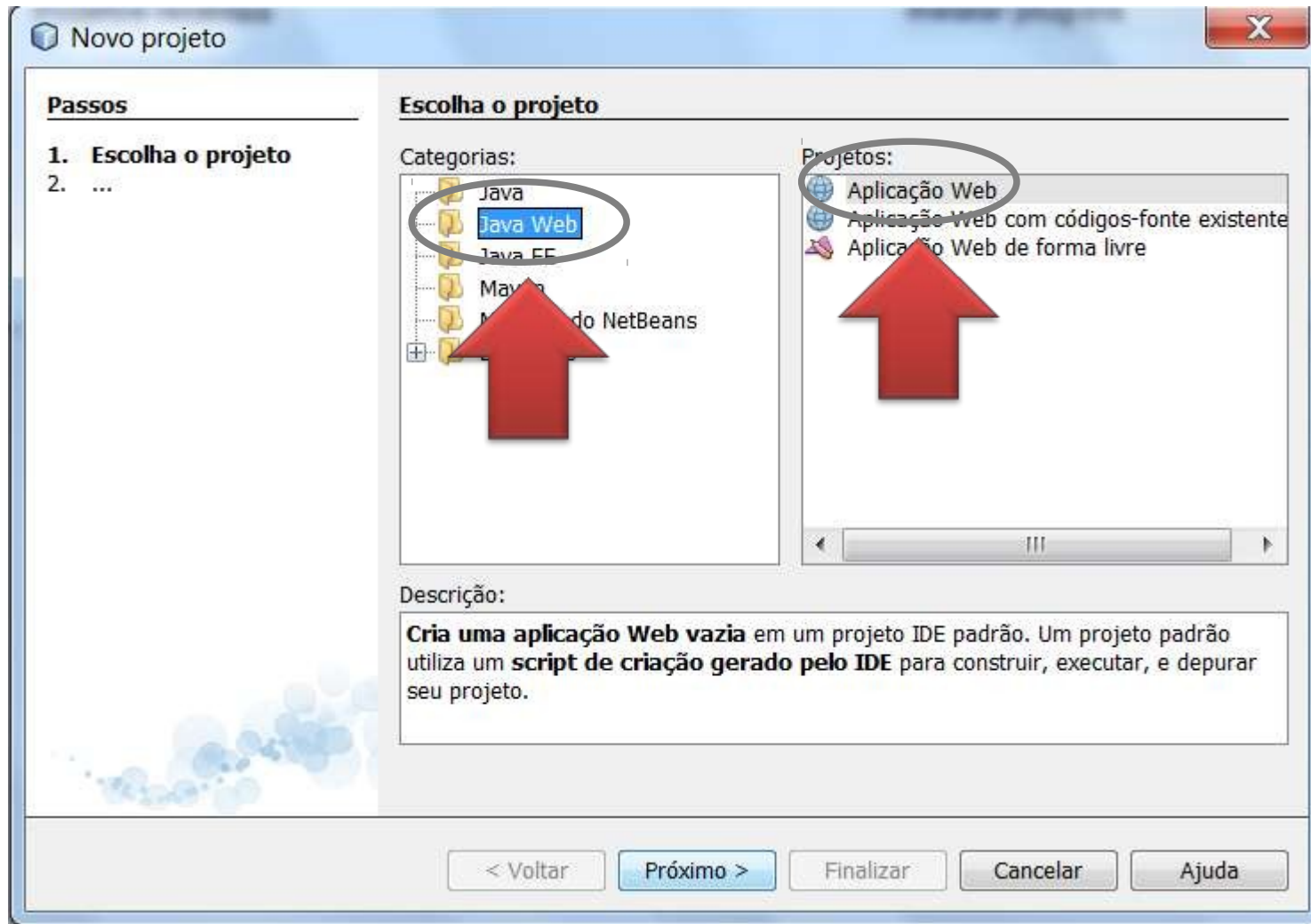
SERVLETS NO NETBEANS

- Iniciaremos criando um projeto



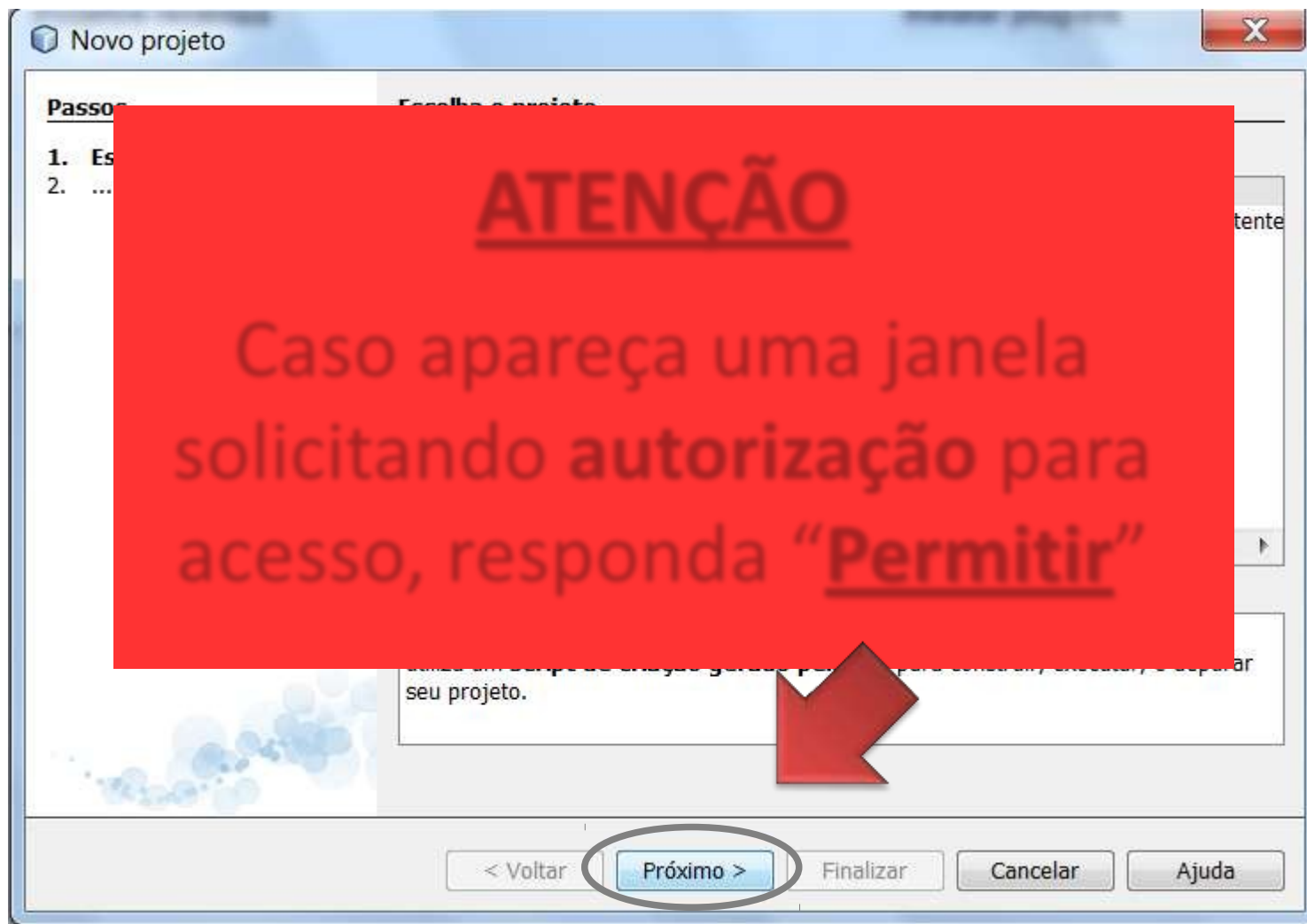
SERVLETS NO NETBEANS

- Escolha o tipo: **Java Web** e **Aplicação Web**



SERVLETS NO NETBEANS

- Escolha o tipo: **Java Web e Aplicação Web**



SERVLETS NO NETBEANS

- Dê um nome ao seu projeto: ex.: WebDaniel

Novo Aplicação Web

Etapas

1. Escolher Projeto
2. Nome e Localização
3. Servidor e Definições
4. Frameworks

Nome e Localização

Nome do Projeto: **WebDaniel**

Localização do Projeto: /home/daniel/NetBeansProjects **Procurar...**

Pasta do Projeto: /home/daniel/NetBeansProjects/WebDaniel

☐ Usar pasta dedicada para armazenar bibliotecas

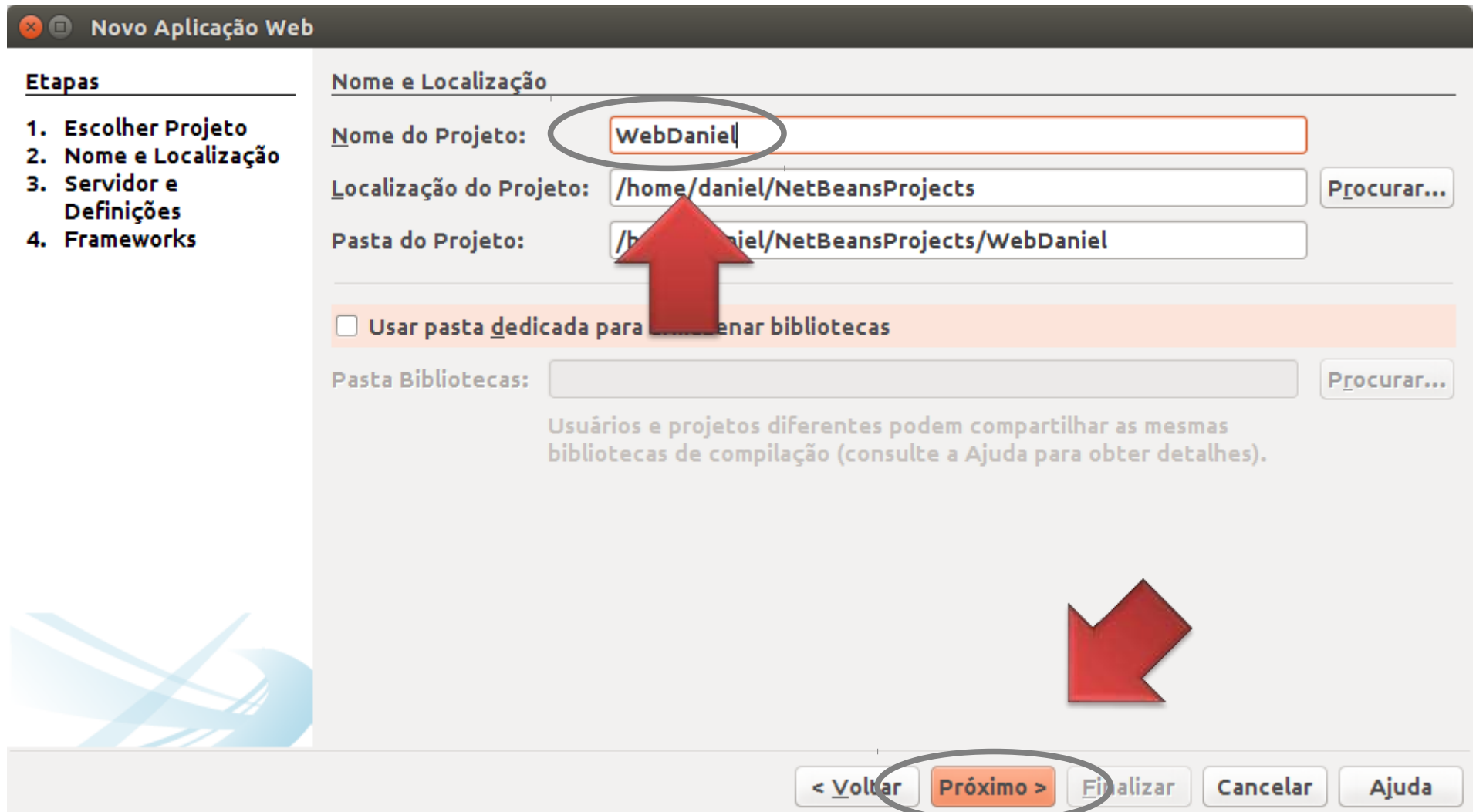
Pasta Bibliotecas: **Procurar...**

Usuários e projetos diferentes podem compartilhar as mesmas bibliotecas de compilação (consulte a Ajuda para obter detalhes).

< Voltar **Próximo >** Finalizar Cancelar Ajuda

SERVLETS NO NETBEANS

- Dê um nome ao seu projeto: ex.: WebDaniel



Novo Aplicação Web

Etapas

1. Escolher Projeto
2. Nome e Localização
3. Servidor e Definições
4. Frameworks

Nome e Localização

Nome do Projeto: WebDaniel

Localização do Projeto: /home/daniel/NetBeansProjects **Procurar...**

Pasta do Projeto: /home/daniel/NetBeansProjects/WebDaniel

☐ Usar pasta dedicada para armazenar bibliotecas

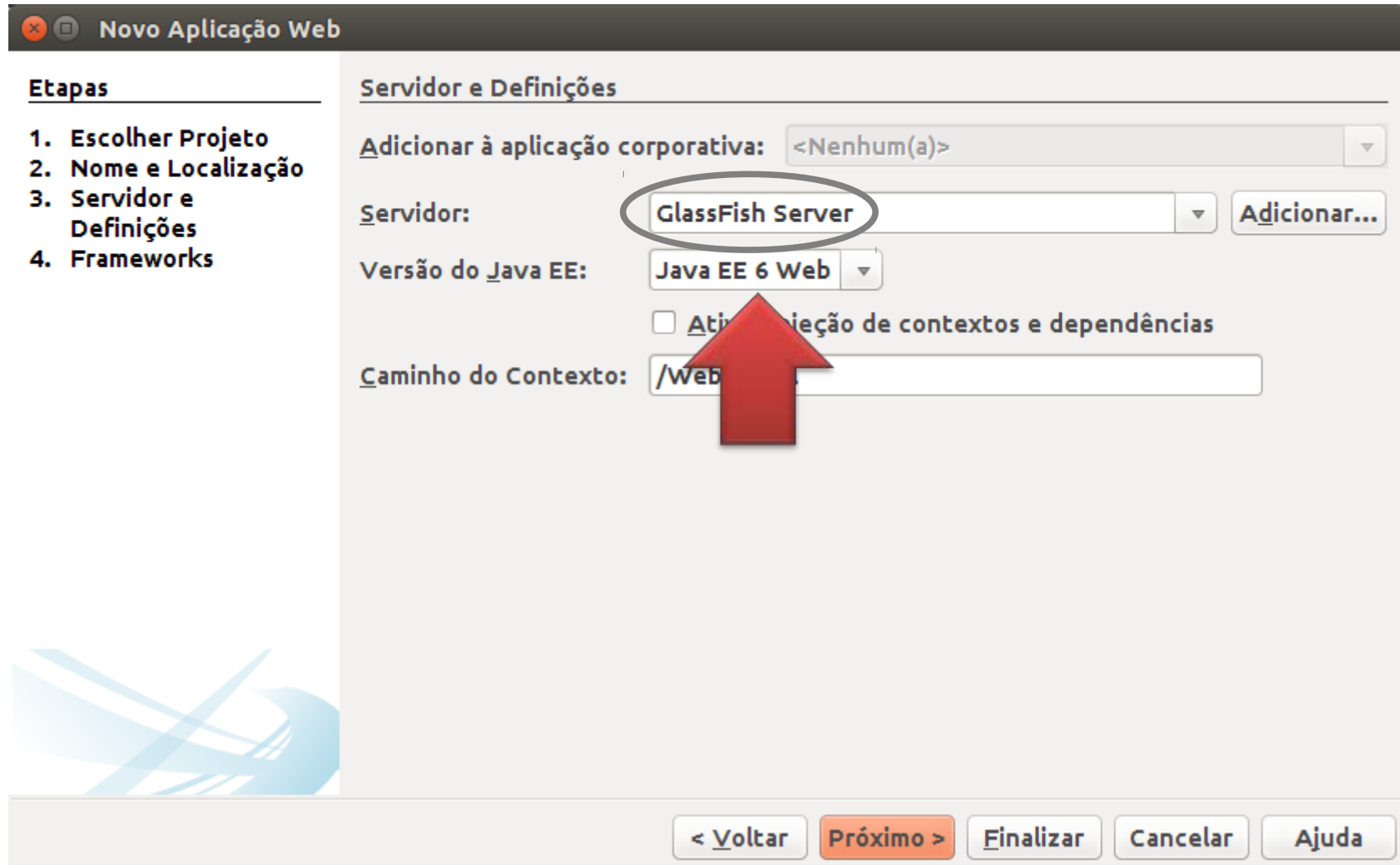
Pasta Bibliotecas: **Procurar...**

Usuários e projetos diferentes podem compartilhar as mesmas bibliotecas de compilação (consulte a Ajuda para obter detalhes).

< Voltar **Próximo >** **Finalizar** **Cancelar** **Ajuda**

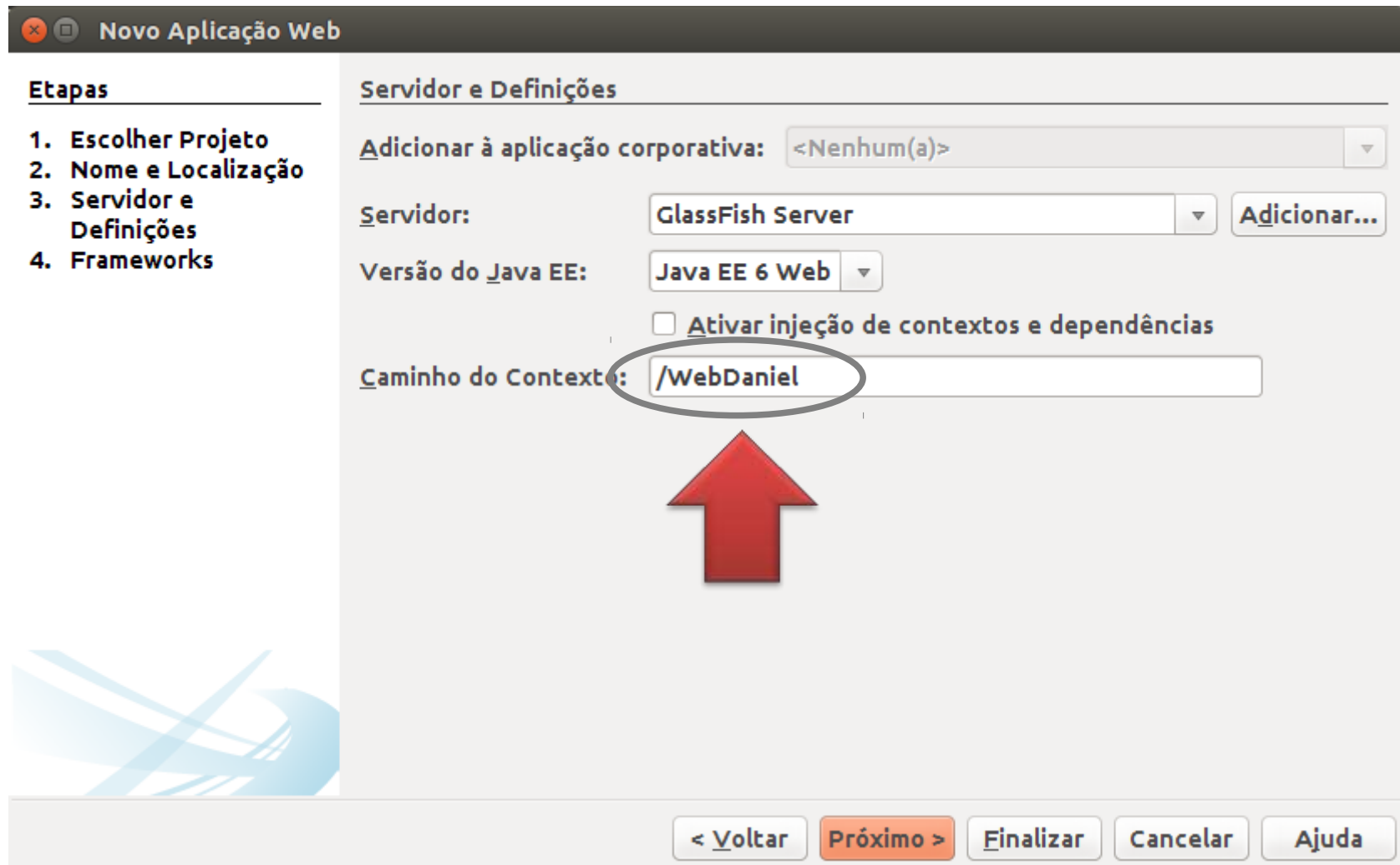
SERVLETS NO NETBEANS

- Selecione o contentor: Glassfish



SERVLETS NO NETBEANS

- Este é o endereço da aplicação: **/WebDaniel**



Novo Aplicação Web

Etapas

1. Escolher Projeto
2. Nome e Localização
3. Servidor e Definições
4. Frameworks

Servidor e Definições

Adicionar à aplicação corporativa: <Nenhum(a)>

Servidor: GlassFish Server [Adicionar...]

Versão do Java EE: Java EE 6 Web

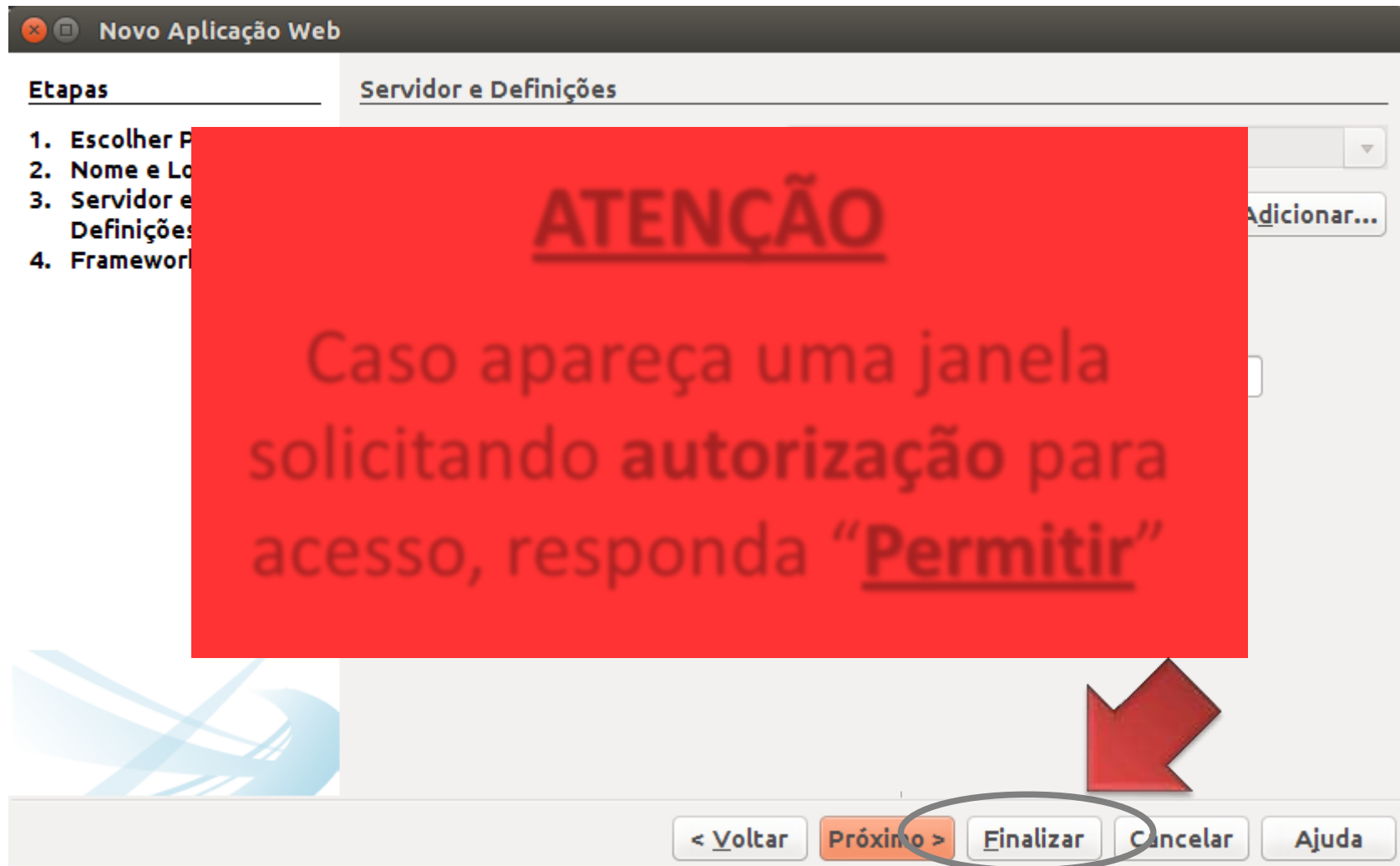
☐ Ativar injeção de contextos e dependências

Caminho do Contexto: **/WebDaniel**

< Voltar Próximo > Finalizar Cancelar Ajuda

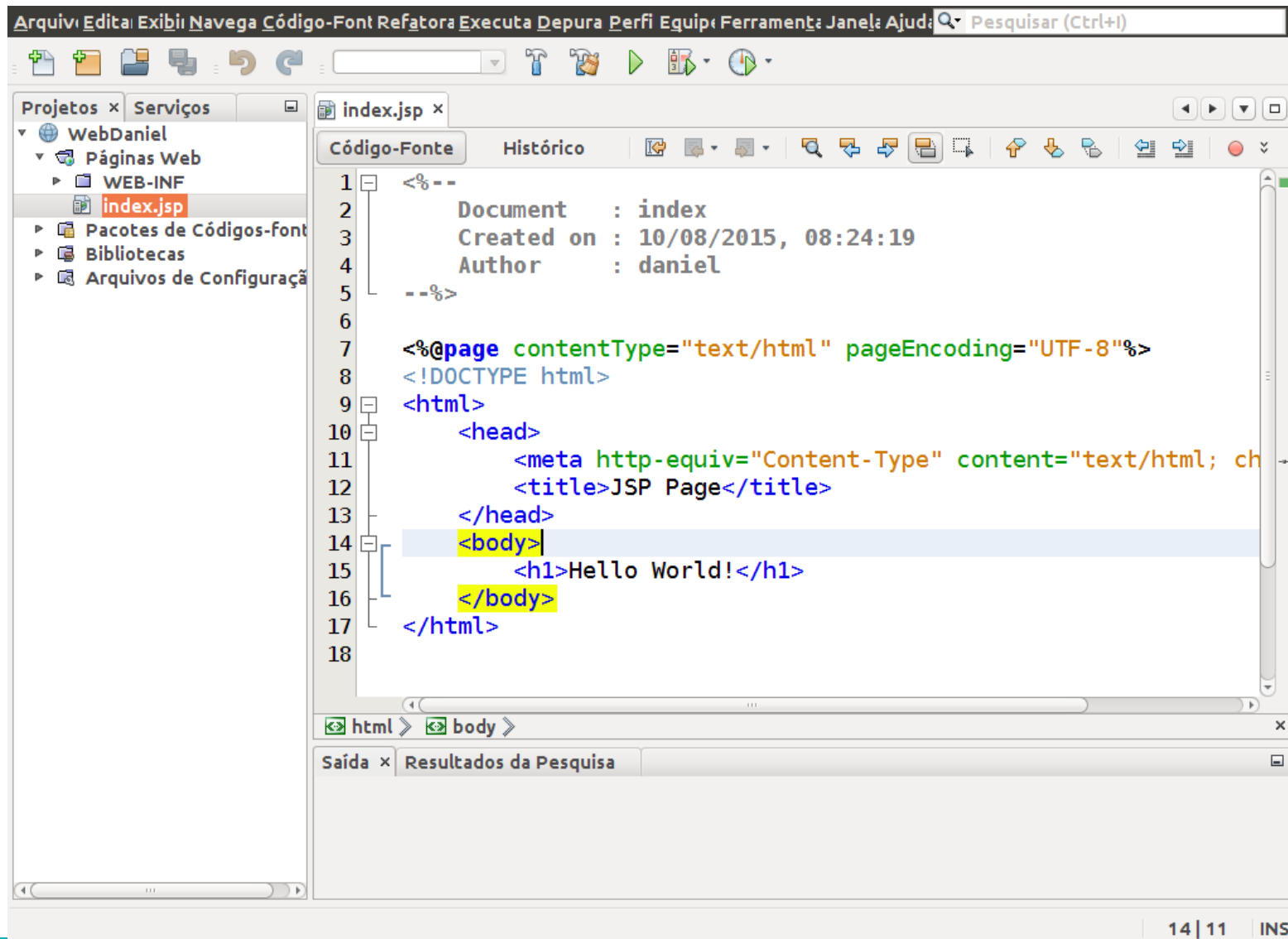
SERVLETS NO NETBEANS

- Este é o endereço da aplicação: **/WebDaniel**



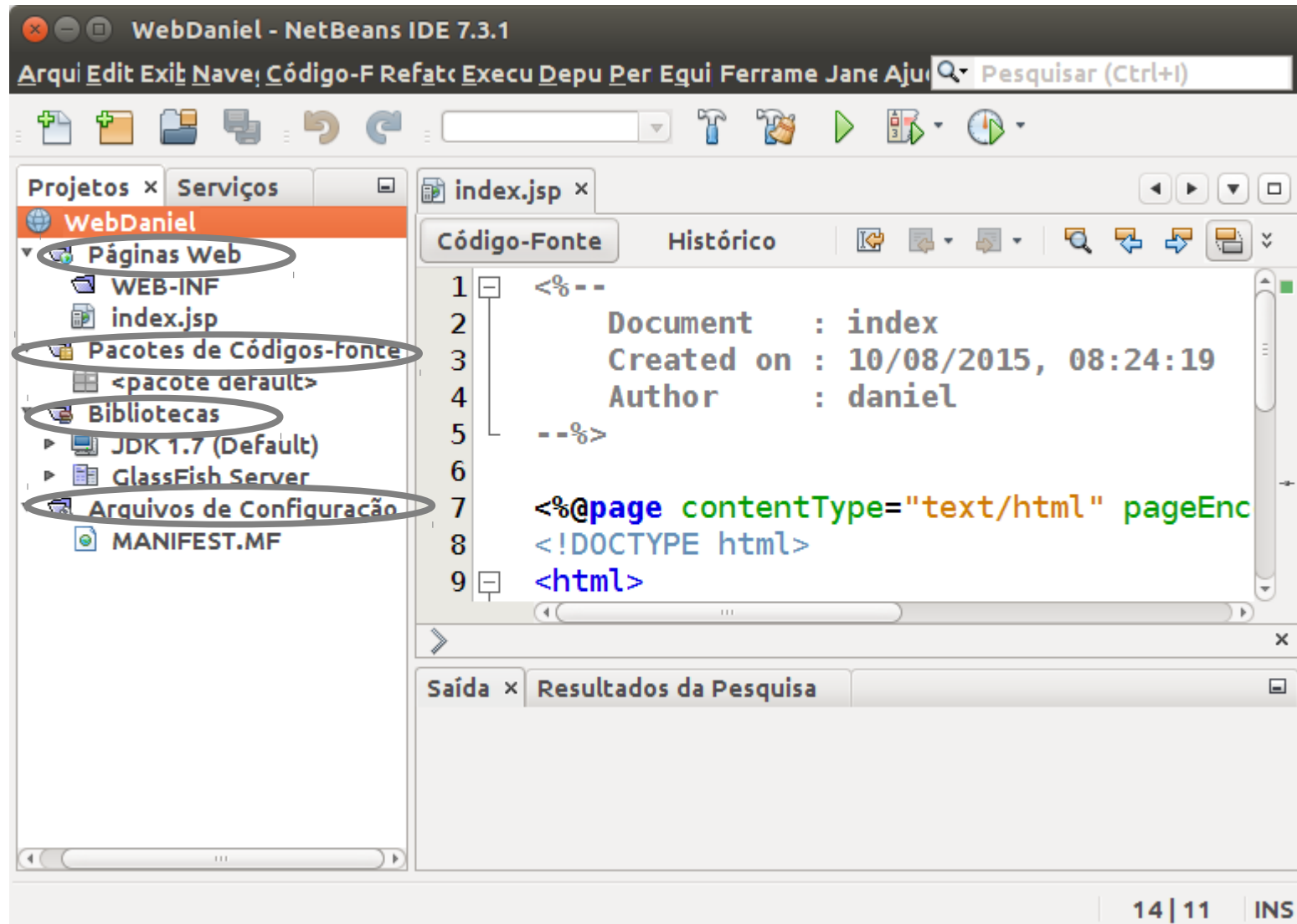
SERVLETS NO NETBEANS

- Esta tela vai aparecer ...



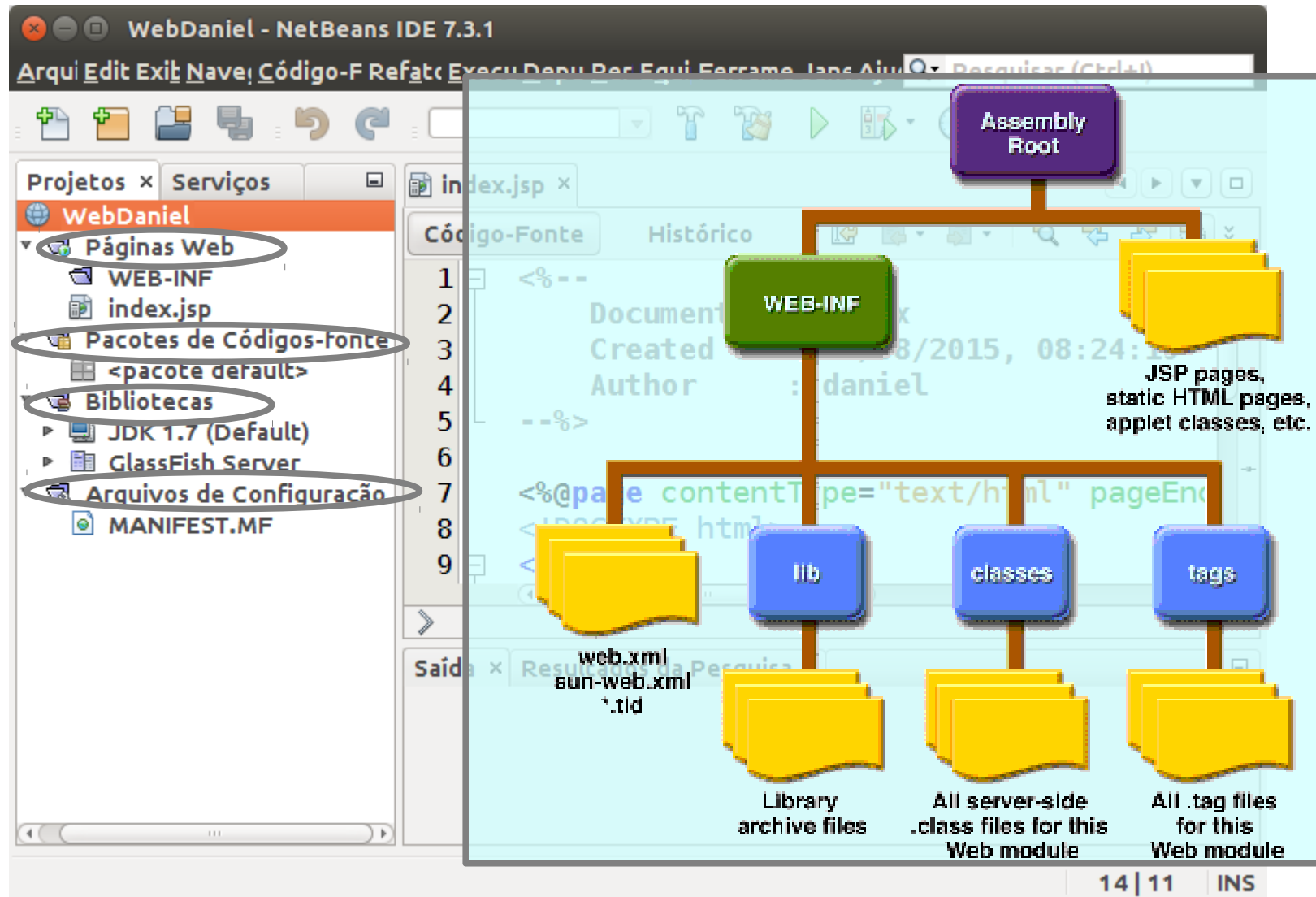
SERVLETS NO NETBEANS

- Expandindo as pastas do Projeto ...



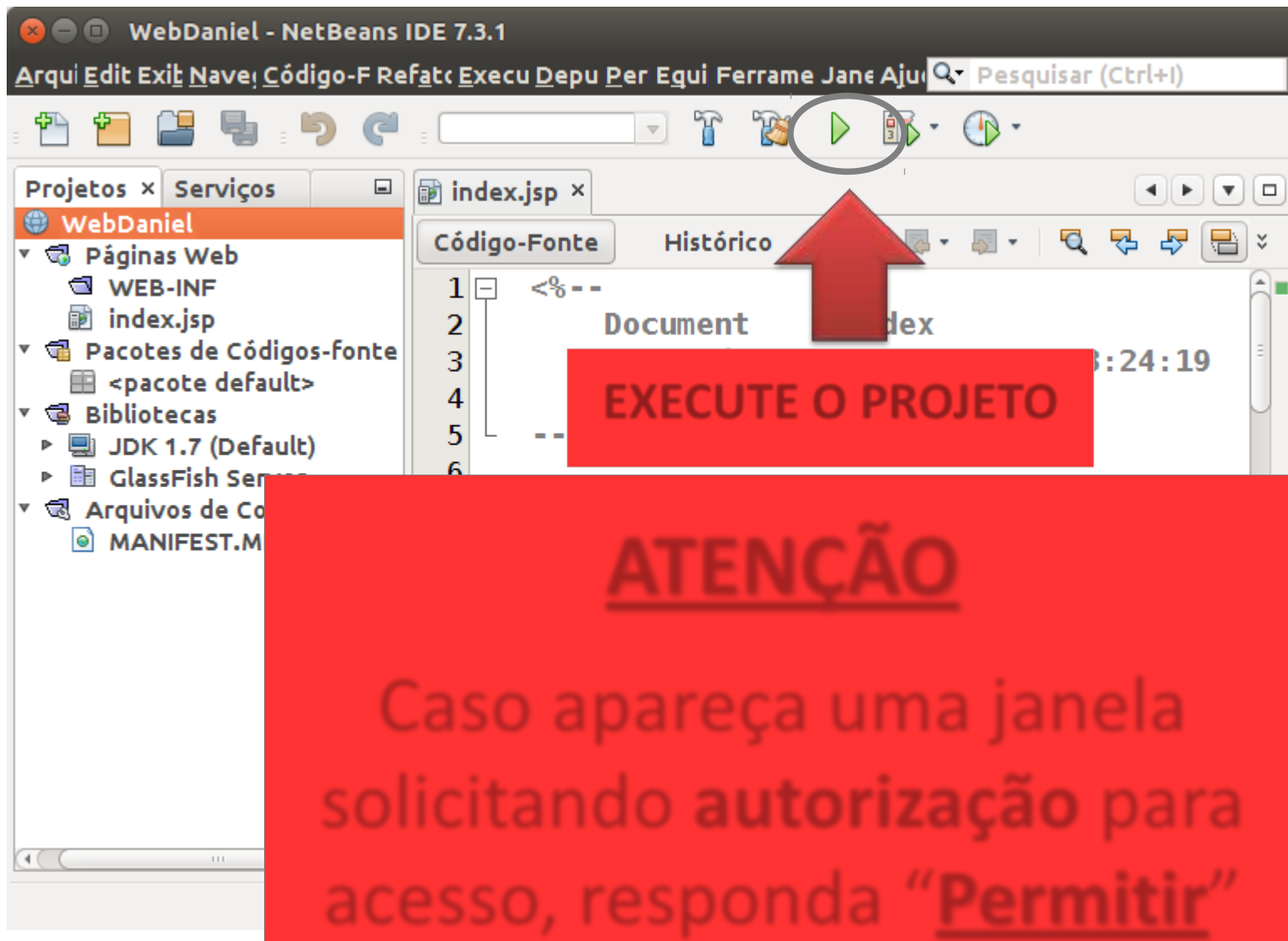
SERVLETS NO NETBEANS

- Expandindo as pastas do Projeto ...



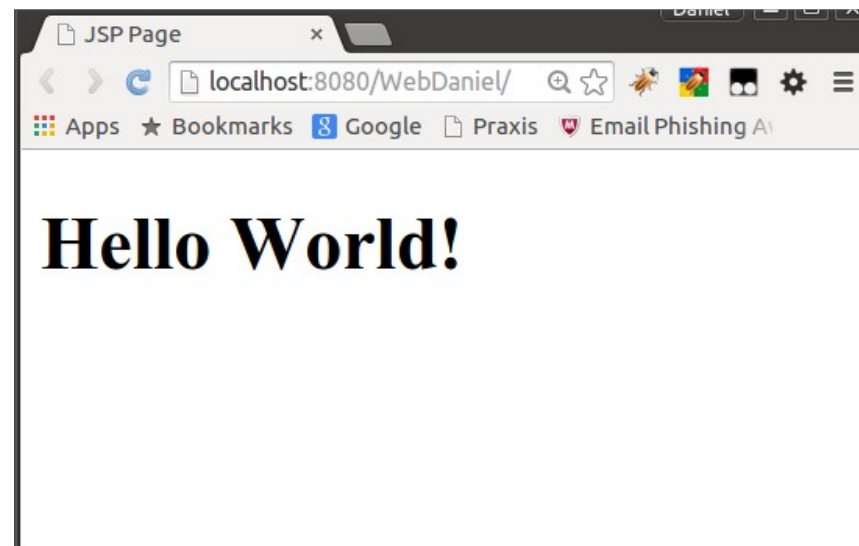
SERVLETS NO NETBEANS

- Execute o projeto ...



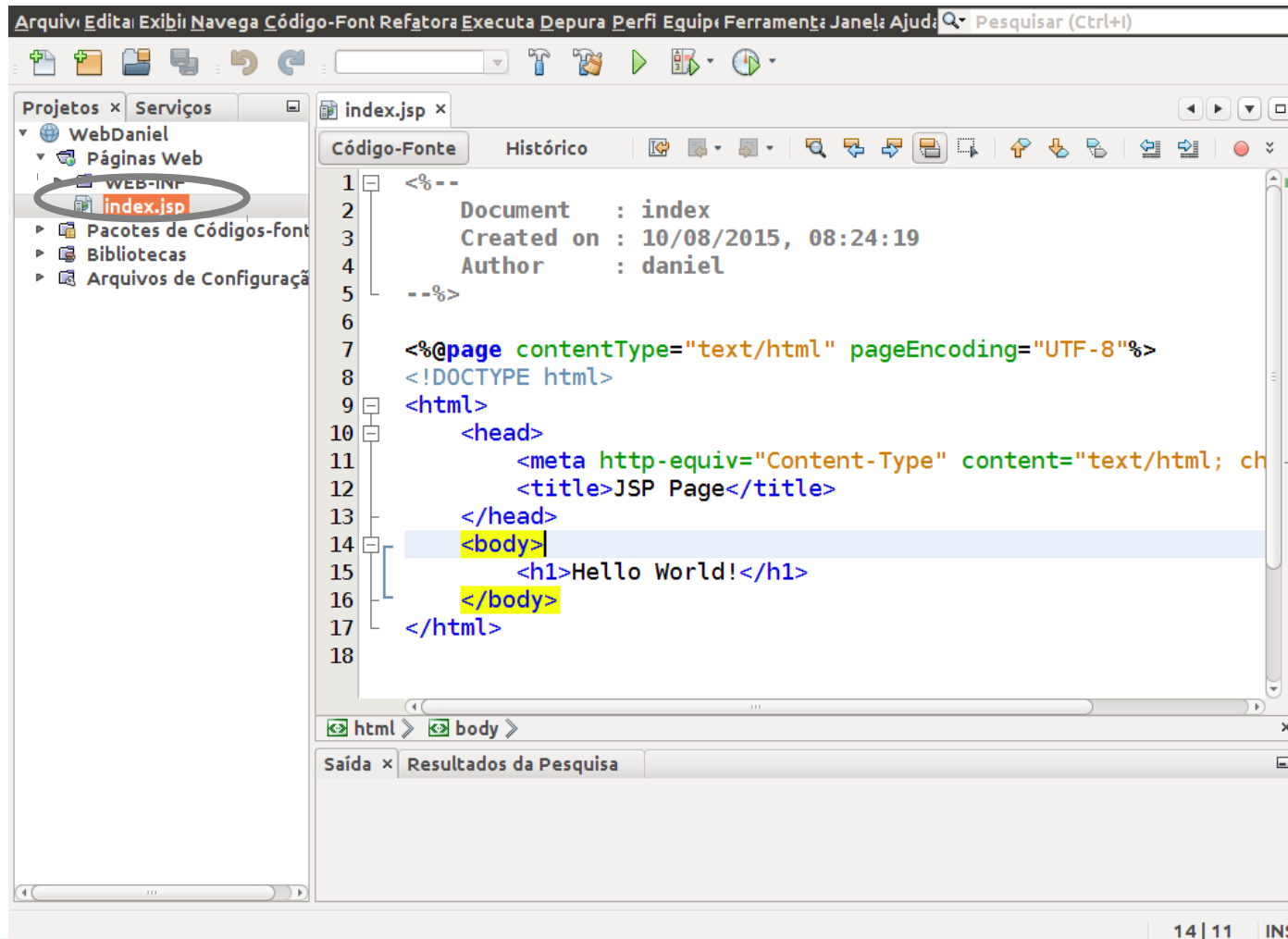
SERVLETS NO NETBEANS

- Durante a execução do projeto:
 - O Glassfish será iniciado
 - Será realizado deploy da aplicação para o Container
 - O navegador será aberto
 - Na raiz de sua aplicação: `http://localhost:8080/WebDaniel`



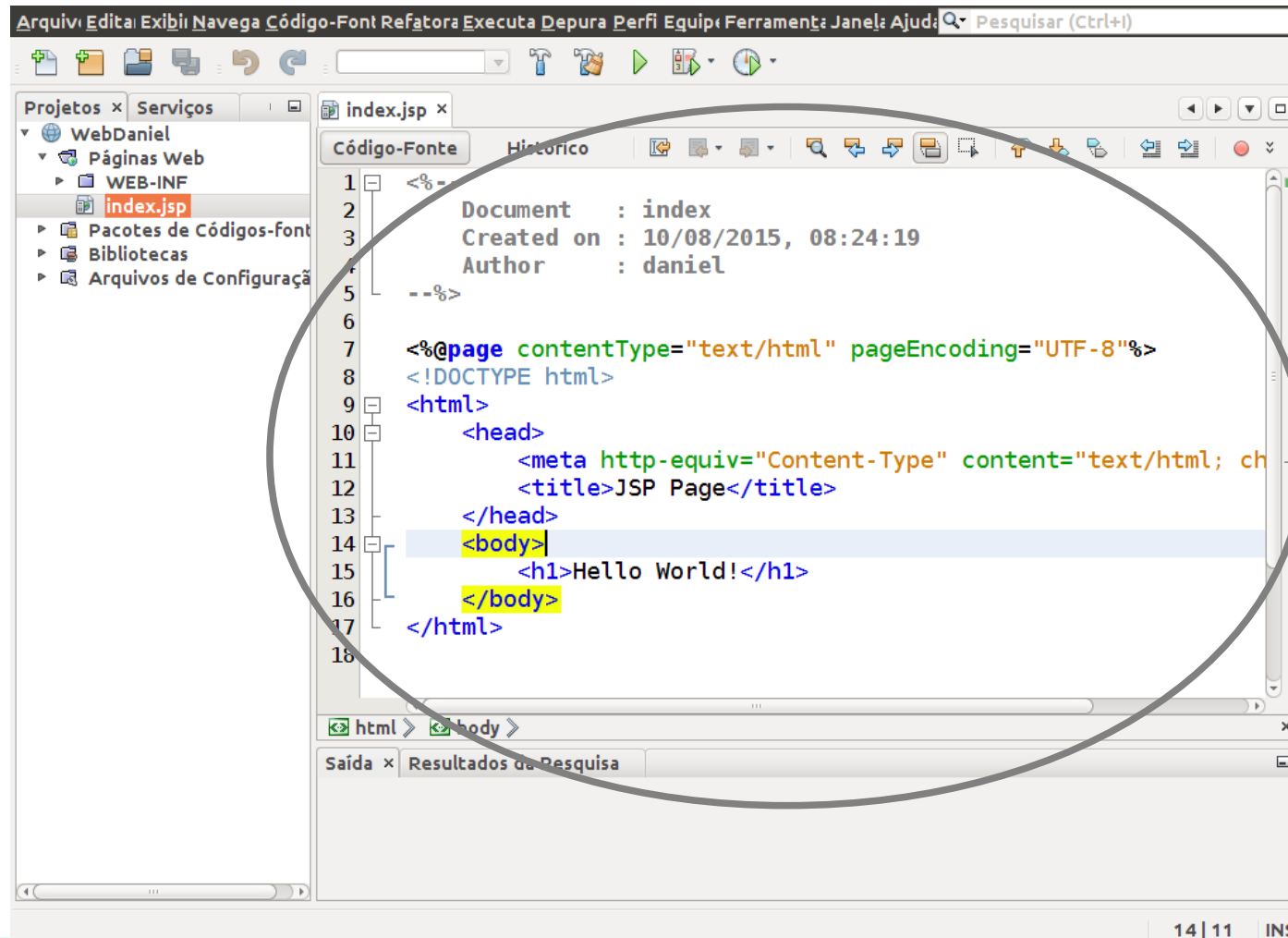
SERVLETS NO NETBEANS

- Por padrão o arquivo index.jsp será aberto
(depende do template de projeto Web de seu Netbeans)



SERVLETS NO NETBEANS

- A página JSP inicial
 - Vamos alterá-la



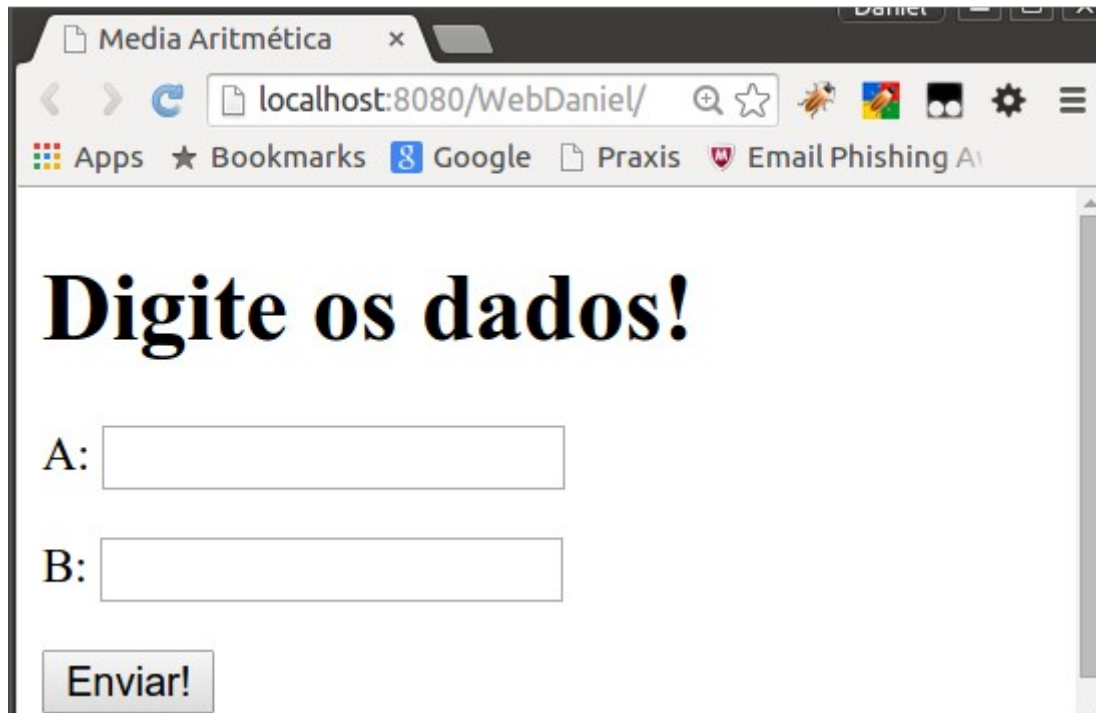
SERVLETS NO NETBEANS

- A página JSP inicial
 - Vamos alterá-la

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"
    <title>Media Aritmética</title>
  </head>
  <body>
    <h1>Digite os dados!</h1>
    <form method="post" action="Calcula">
      <p>A: <input type="text" name="valorA"/></p>
      <p>B: <input type="text" name="valorB"/></p>
      <p><input type="submit" value="Enviar!"/></p>
    </form>
  </body>
</html>
```

SERVLETS NO NETBEANS

- A página JSP alterada!



SERVLETS NO NETBEANS

- A página JSP inicial
 - Vamos alterá-la

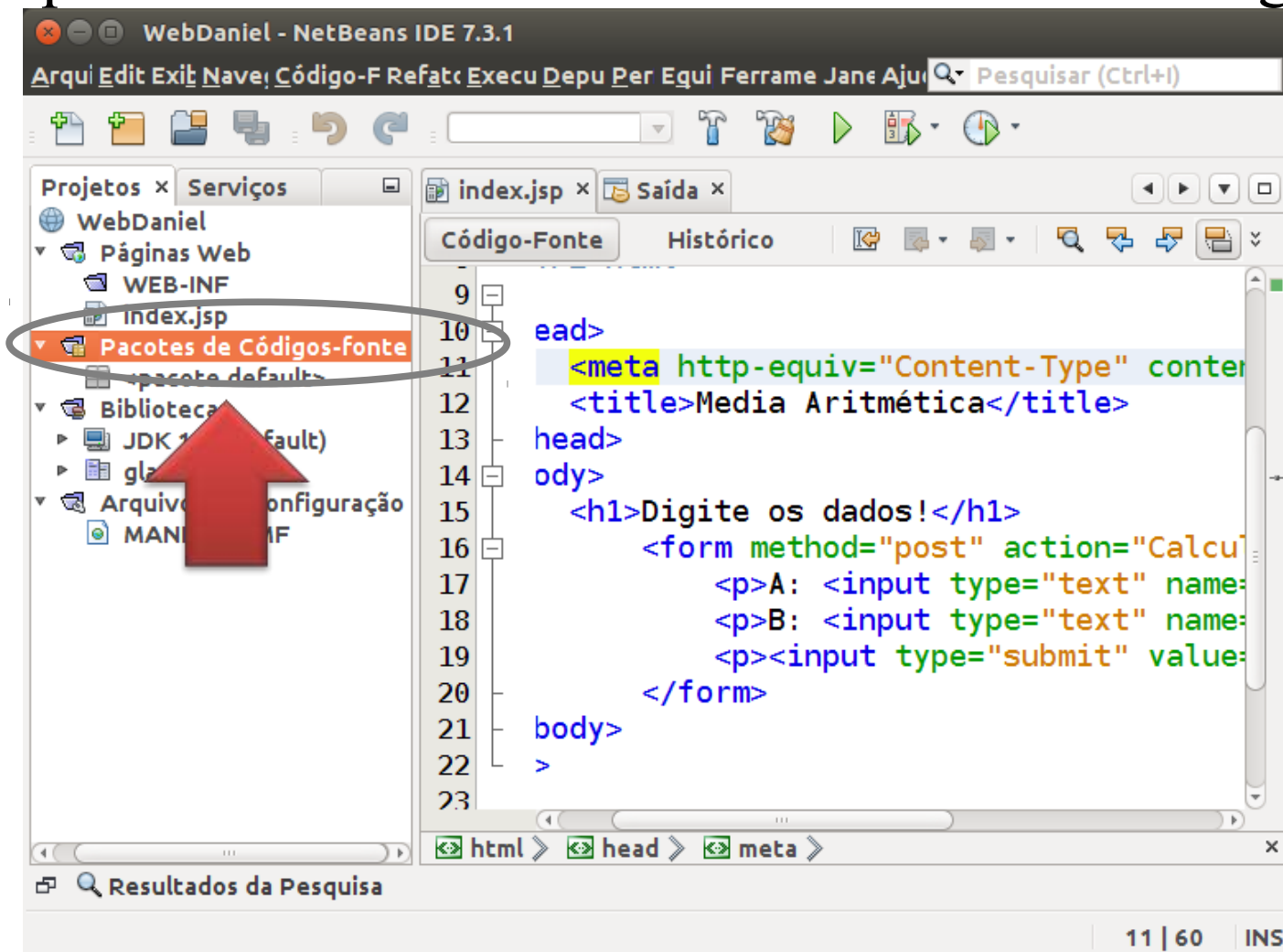
```
<%@page con  
<!DOCTYPE H  
<html>  
  <head>  
    <me  
    <ti  
  </head>  
  <body>  
    <h1>Digite os dados!</h1>  
    <form method="post" action="Calcula">  
      <p>A: <input type="text" name="valorA"/></p>  
      <p>B: <input type="text" name="valorB"/></p>  
      <p><input type="submit" value="Enviar!"/></p>  
    </form>  
  </body>  
</html>
```

Isso irá gerar uma request para
uma URL chamada Calcula.
Precisamos criá-la!

t=UTF-8"

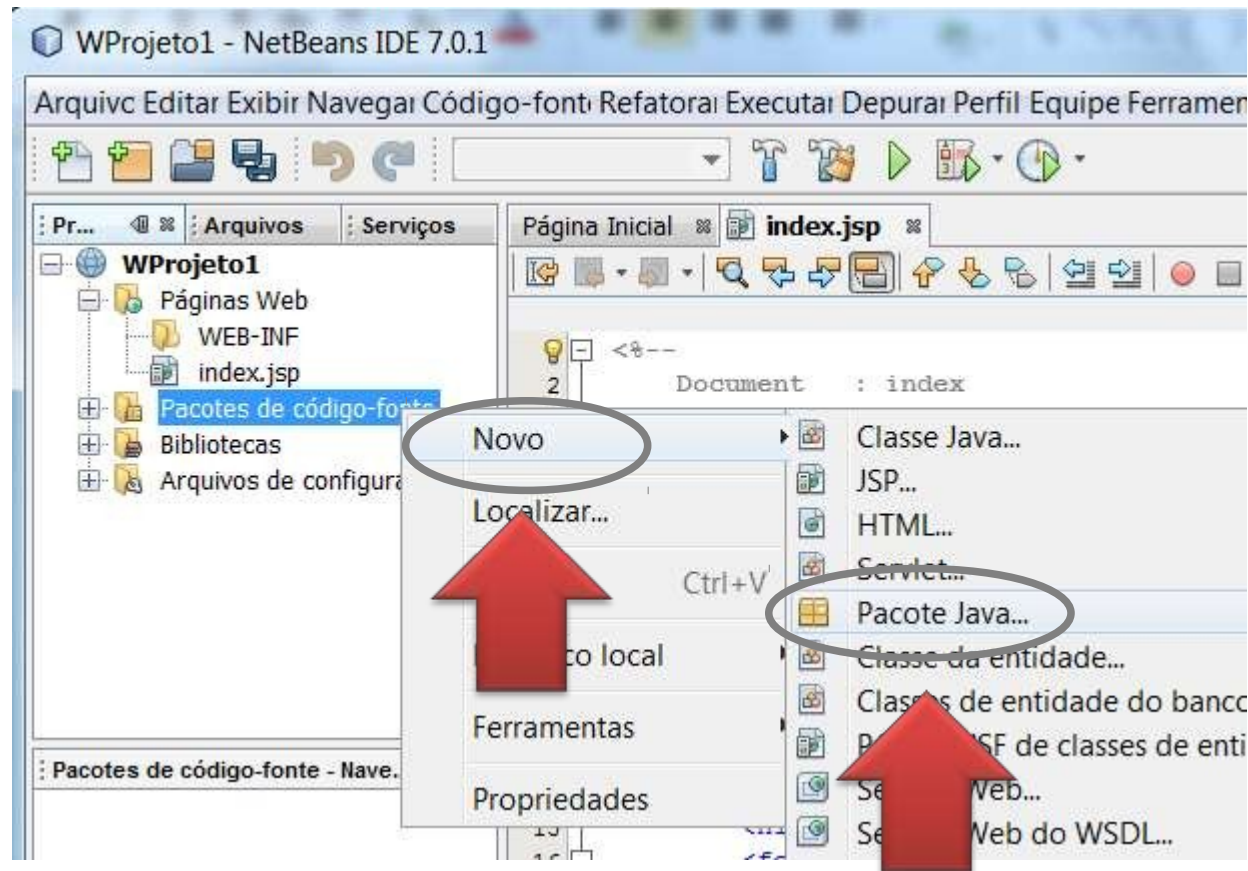
SERVLETS NO NETBEANS

- Criando um Servlet
 - Clique com o botão direito em “Pacotes de Códigos-Fonte



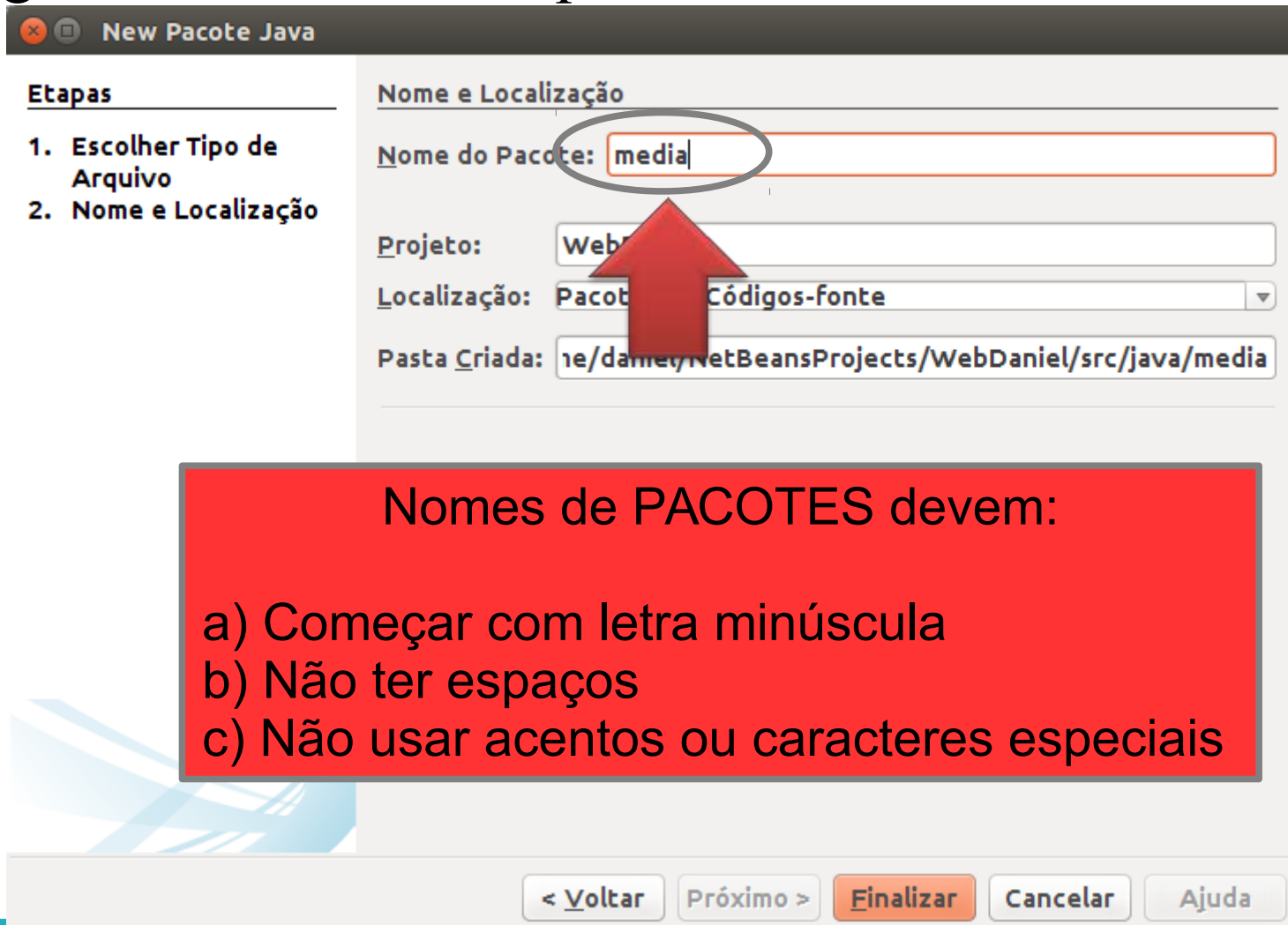
SERVLETS NO NETBEANS

- Criando um Servlet
 - Selecione: Novo → Pacote Java



SERVLETS NO NETBEANS

- Criando um Servlet
 - Agora dê um nome ao pacote



Novo Pacote Java

Etapas

1. Escolher Tipo de Arquivo
2. Nome e Localização

Nome e Localização

Nome do Pacote: media

Projeto: Web

Localização: Pacote Códigos-fonte

Pasta Criada: ne/daniel/NetBeansProjects/WebDaniel/src/java/media

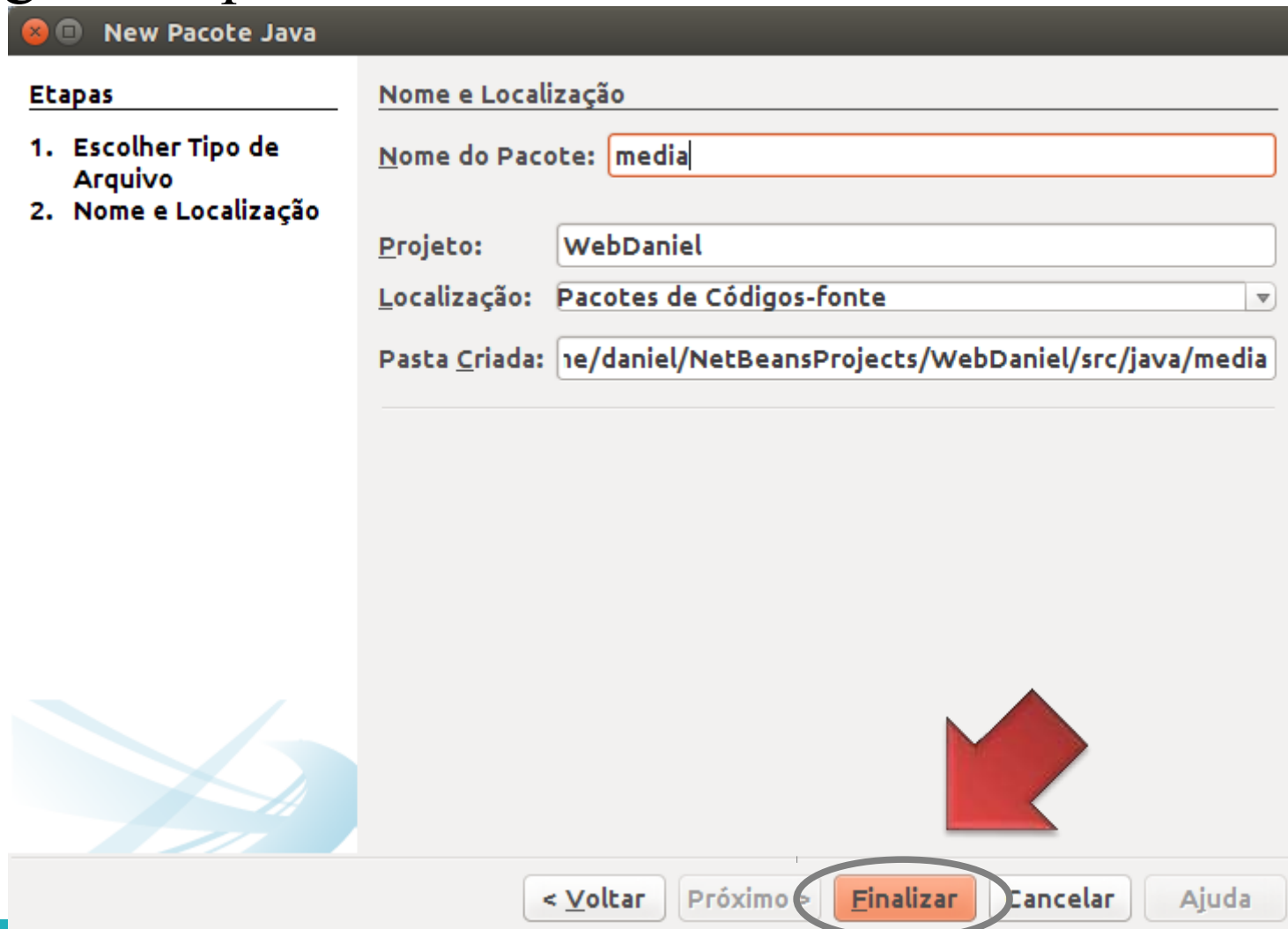
Nomes de PACOTES devem:

- a) Começar com letra minúscula
- b) Não ter espaços
- c) Não usar acentos ou caracteres especiais

< Voltar Próximo > Finalizar Cancelar Ajuda

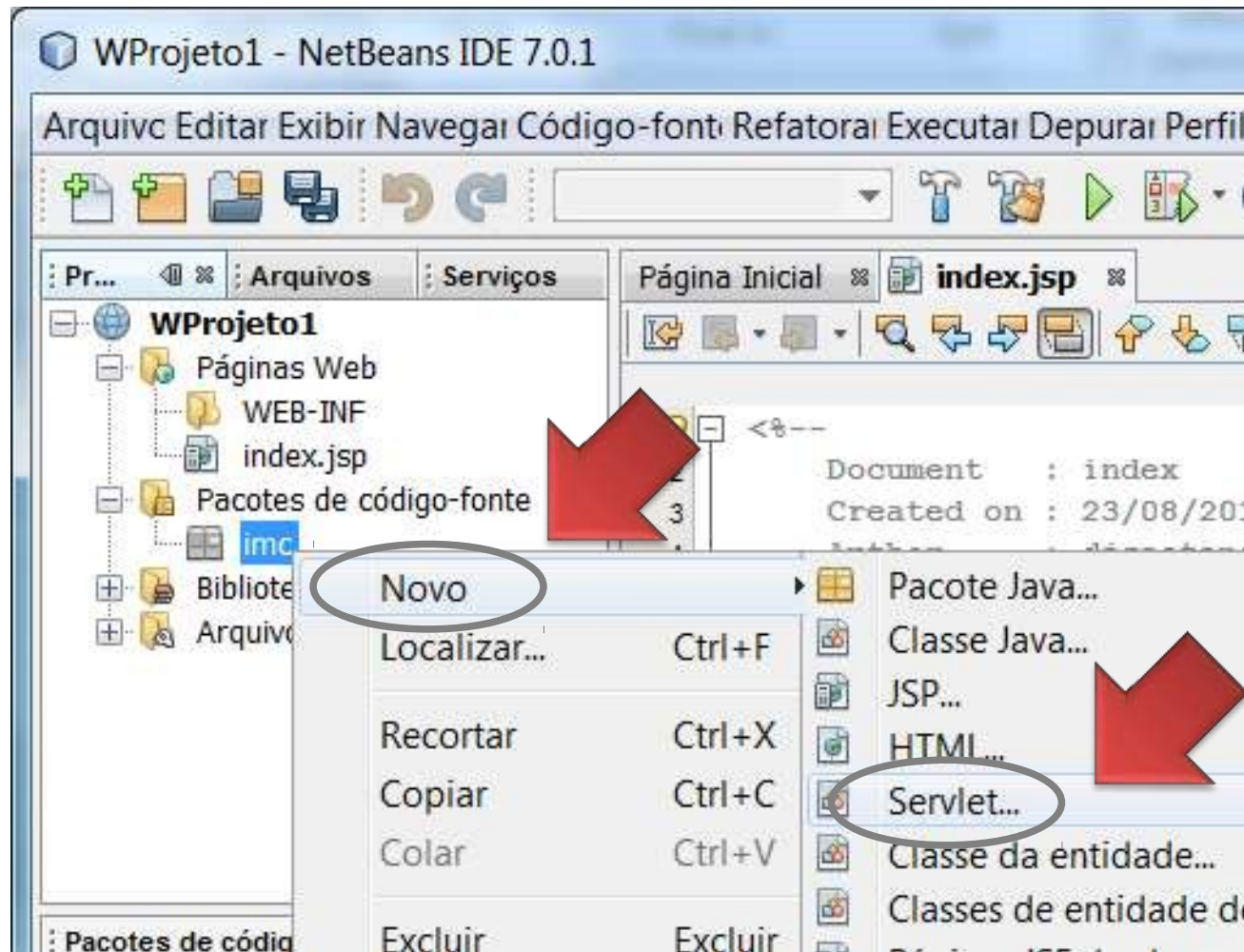
SERVLETS NO NETBEANS

- Criando um Servlet
 - Agora clique em Finalizar



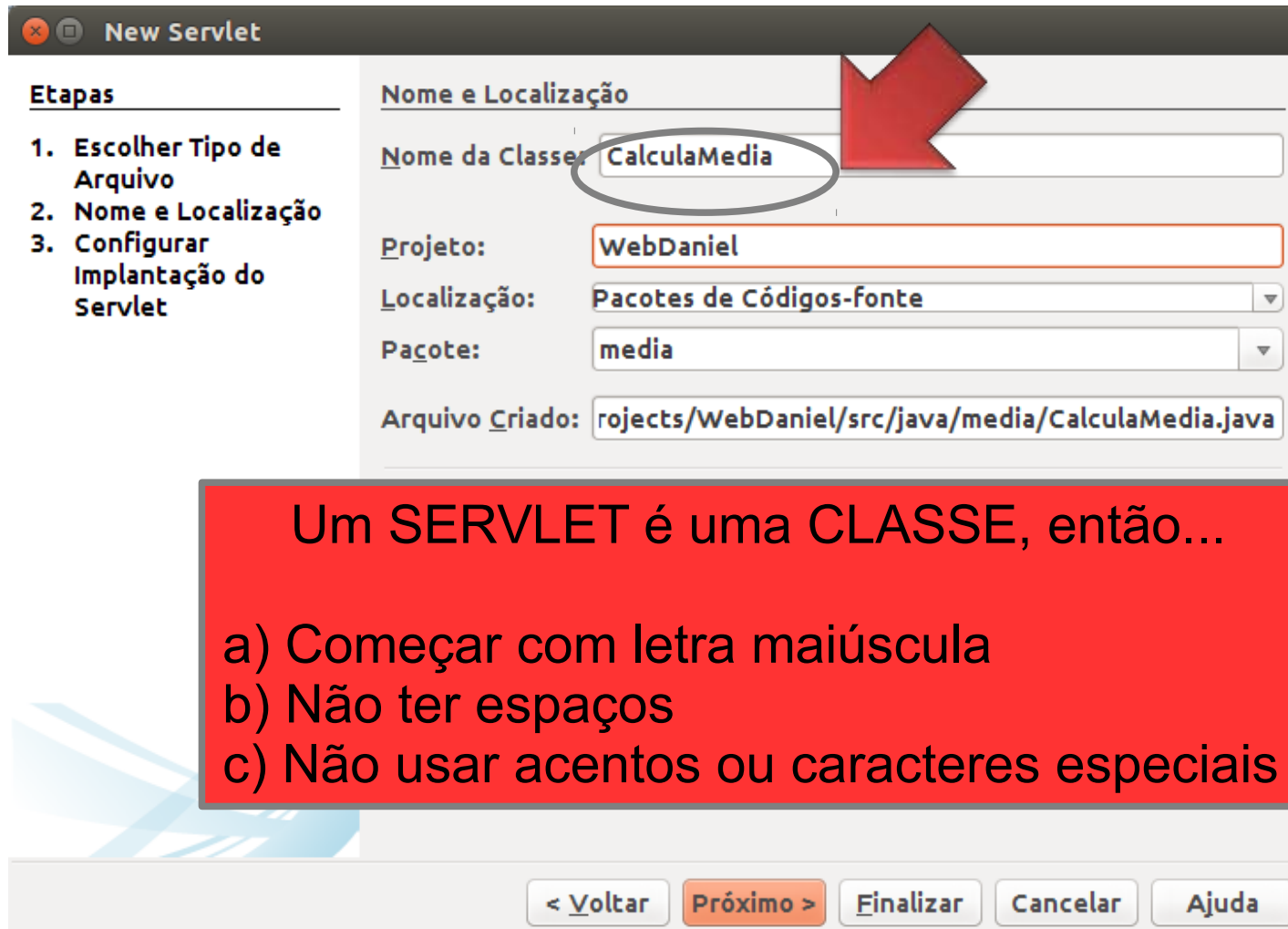
SERVLETS NO NETBEANS

- Criando um Servlet
 - Botão direito no pacote media e selecione Novo → Servlet



SERVLETS NO NETBEANS

- Criando um Servlet
 - Agora dê um nome ao servlet: CalculaMedia



Novo Servlet

Etapas

1. Escolher Tipo de Arquivo
2. Nome e Localização
3. Configurar Implantação do Servlet

Nome e Localização

Nome da Classe: CalculaMedia

Projeto: WebDaniel

Localização: Pacotes de Códigos-Fonte

Paquete: media

Arquivo Criado: projects/WebDaniel/src/java/media/CalculaMedia.java

Um SERVLET é uma CLASSE, então...

- a) Começar com letra maiúscula
- b) Não ter espaços
- c) Não usar acentos ou caracteres especiais

< Voltar Próximo > Finalizar Cancelar Ajuda

SERVLETS NO NETBEANS

- Criando um Servlet
 - Agora clique em Próximo

New Servlet

Etapas

1. Escolher Tipo de Arquivo
2. Nome e Localização
3. Configurar Implantação do Servlet

Nome e Localização

Nome da Classe:

Projeto:

Localização:

Paquete:

Arquivo Criado:

Buttons: < Voltar, **Próximo >**, Finalizar, Cancelar, Ajuda

SERVLETS NO NETBEANS

- Criando um Servlet
 - MARQUE a opção Adiciona informação ao descritor de implementação

New Servlet

Etapas

1. Escolher Tipo de Arquivo
2. Nome e Localização
3. Configurar Implantação do Servlet

Configurar Implantação do Servlet

Registre o Servlet na aplicação atribuindo ao Servlet um nome interno (Nome do Servlet). Depois especifique os padrões que identificam as URLs que invocam o Servlet. Separar múltiplos padrões com vírgulas.

☒ Adicionar informações ao descritor de implementação (web.xml)

Nome da Classe:

Nome do Servlet:

Padrão(ões) de URL:

Parâmetros de Inicialização:

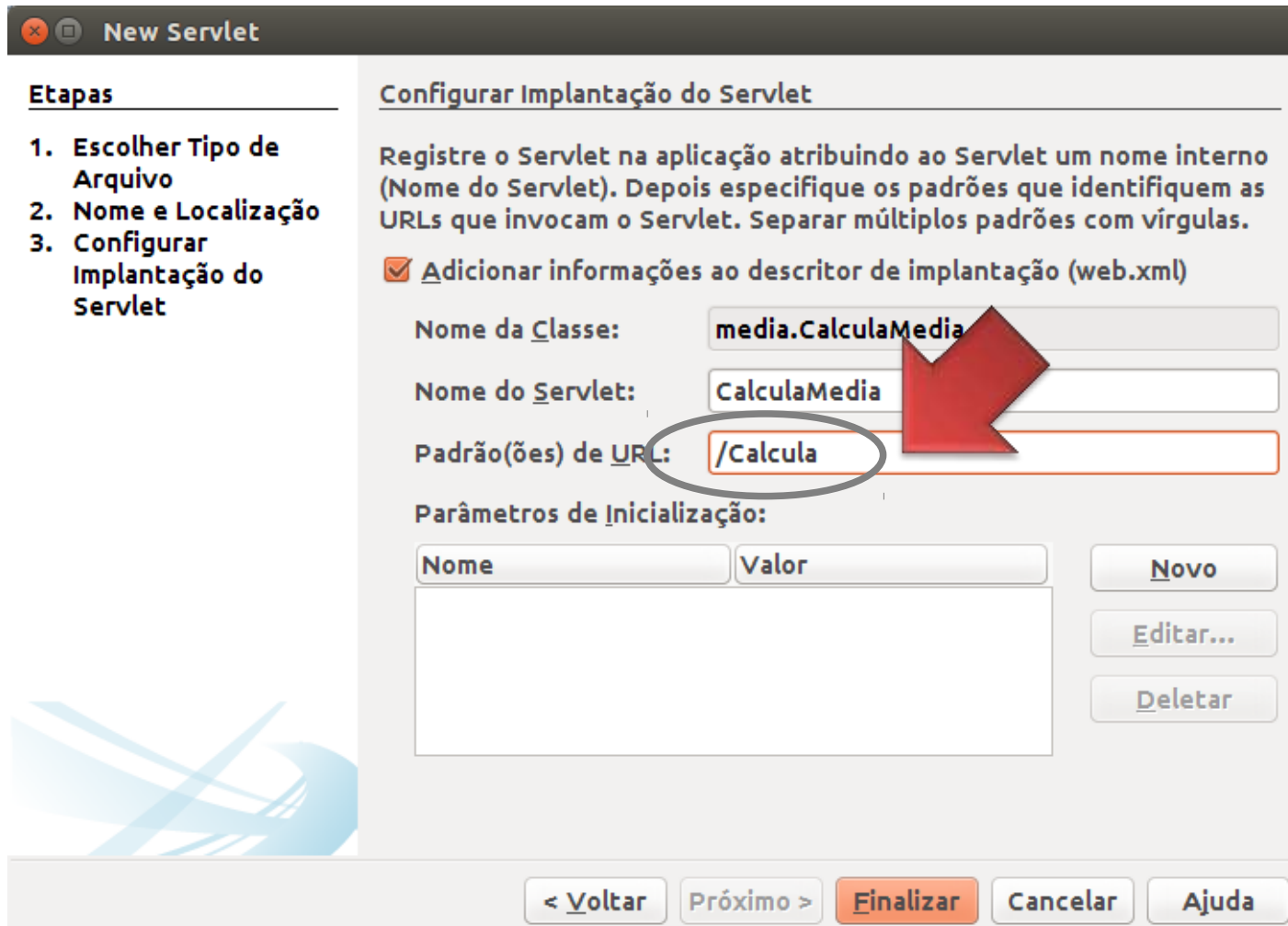
Nome	Valor
------	-------

Novo
Editar...
Deletar

< Voltar **Próximo >** **Finalizar** **Cancelar** **Ajuda**

SERVLETS NO NETBEANS

- Criando um Servlet
 - Defina o padrão de URL



New Servlet

Etapas

1. Escolher Tipo de Arquivo
2. Nome e Localização
3. Configurar Implantação do Servlet

Configurar Implantação do Servlet

Registre o Servlet na aplicação atribuindo ao Servlet um nome interno (Nome do Servlet). Depois especifique os padrões que identifiquem as URLs que invocam o Servlet. Separar múltiplos padrões com vírgulas.

☒ Adicionar informações ao descritor de implantação (web.xml)

Nome da Classe:

Nome do Servlet:

Padrão(ões) de URL:

Parâmetros de Inicialização:

Nome	Valor

SERVLETS NO NETBEANS

- Criando um Servlet
 - Vamos finalizar a criação de nosso 1º Servlet

The screenshot shows the 'New Servlet' dialog box in NetBeans IDE. The title bar reads 'New Servlet'. On the left, under 'Etapas', the steps are: 1. Escolher Tipo de Arquivo, 2. Nome e Localização, and 3. Configurar Implantação do Servlet. The main area is titled 'Configurar Implantação do Servlet' and contains the following fields and options:

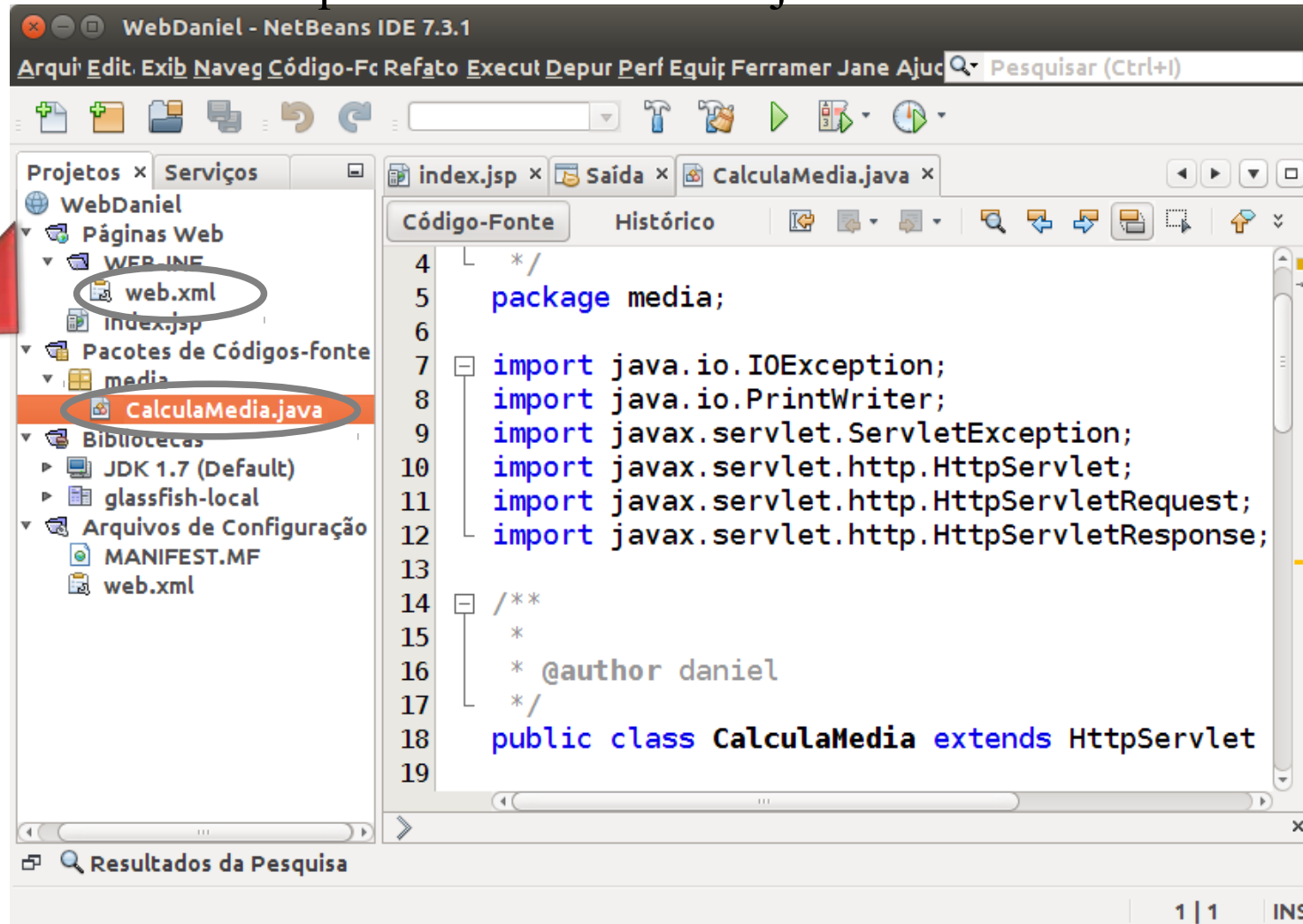
- Register the Servlet in the application by assigning an internal name (Servlet Name). Then specify the patterns that identify the URLs that invoke the Servlet. Separate multiple patterns with commas.
- ☒ Adicionar informações ao descritor de implantação (web.xml)
- Nome da Classe:
- Nome do Servlet:
- Padrão(ões) de URL:
- Parâmetros de Inicialização:

Nome	Valor

Buttons on the right: Novo, Editar..., Deletar. At the bottom, navigation buttons: < Voltar, Próximo >, Finalizar (highlighted with a red circle and a large red arrow), Cancelar, and Ajuda.

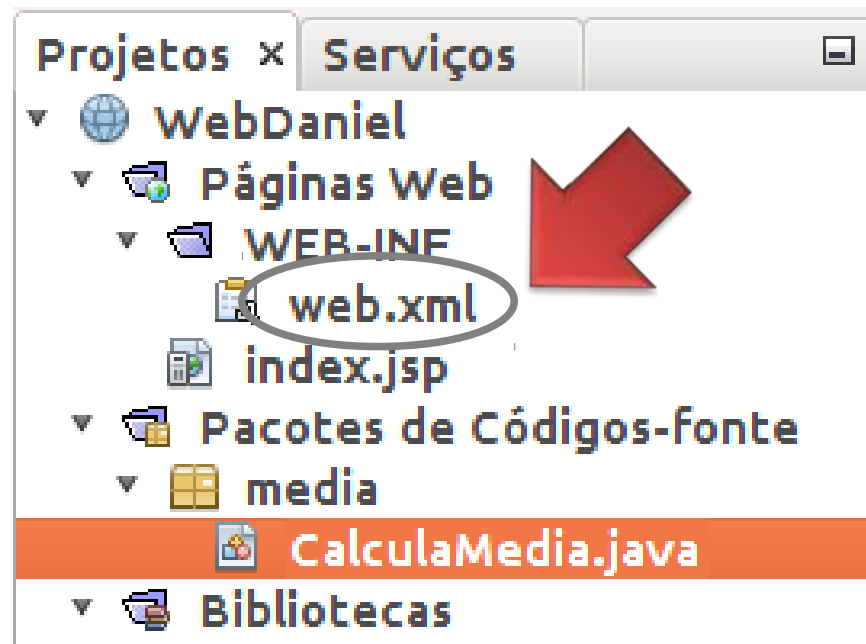
SERVLETS NO NETBEANS

- Criando um Servlet
 - Criamos 2 arquivos: CalculaMedia.java e web.xml



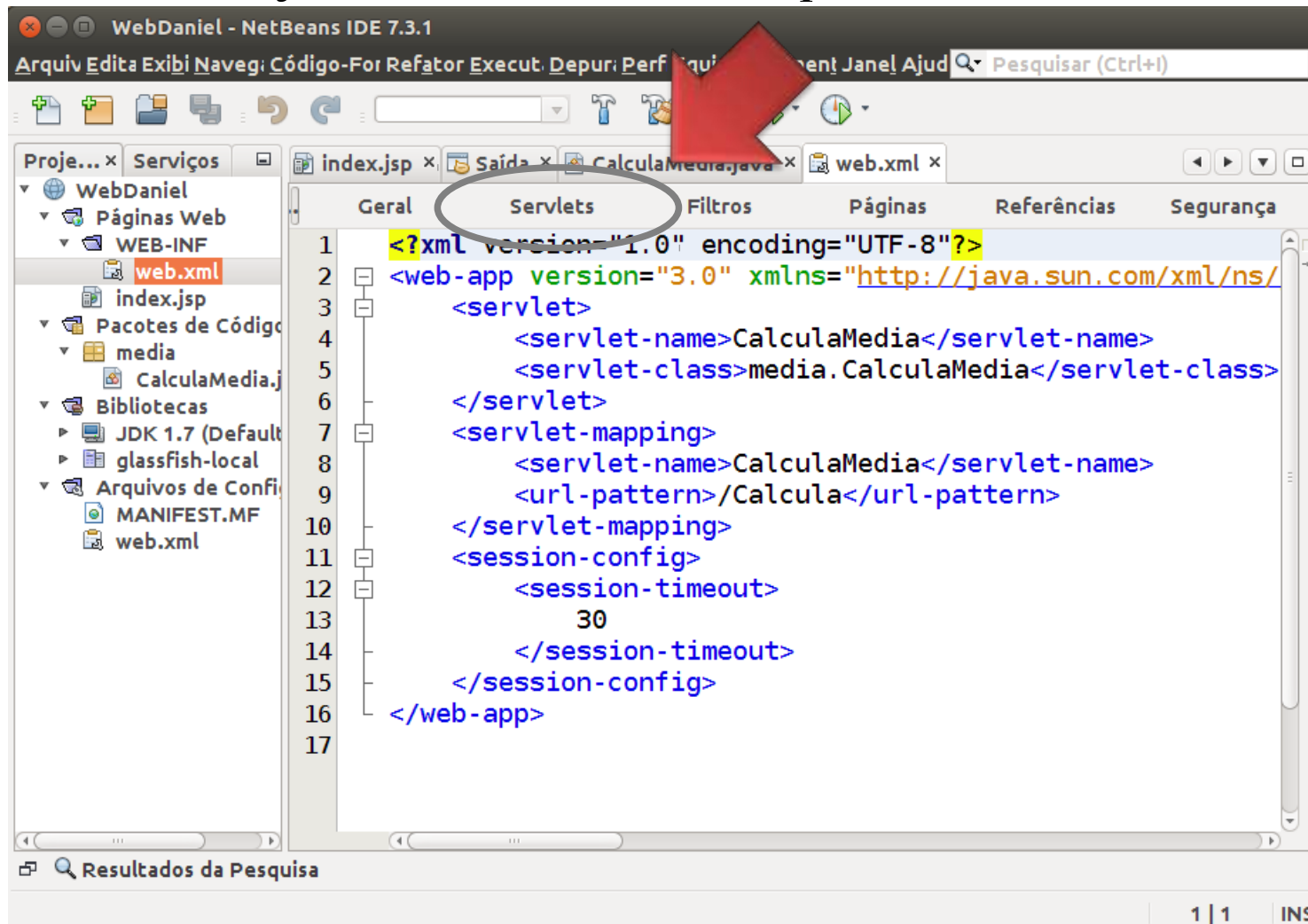
SERVLETS NO NETBEANS

- Configurando o Servlet
 - O arquivo de configuração principal é chamado: web.xml
 - Algumas vezes existem vários arquivos de configuração
 - Ele está na pasta WEB-INF
 - Clique duas vezes no arquivo ...



SERVLETS NO NETBEANS

- Configurando o Servlet
 - Caso não esteja na aba Servlet, vamos para ela ...



SERVLETS NO NETBEANS

- Configurando o Servlet
 - Na aba Servlet algumas informações importantes

The screenshot shows the 'CalculaMedia -> /Calcula' configuration window in NetBeans. Several fields are circled in grey to highlight important information:

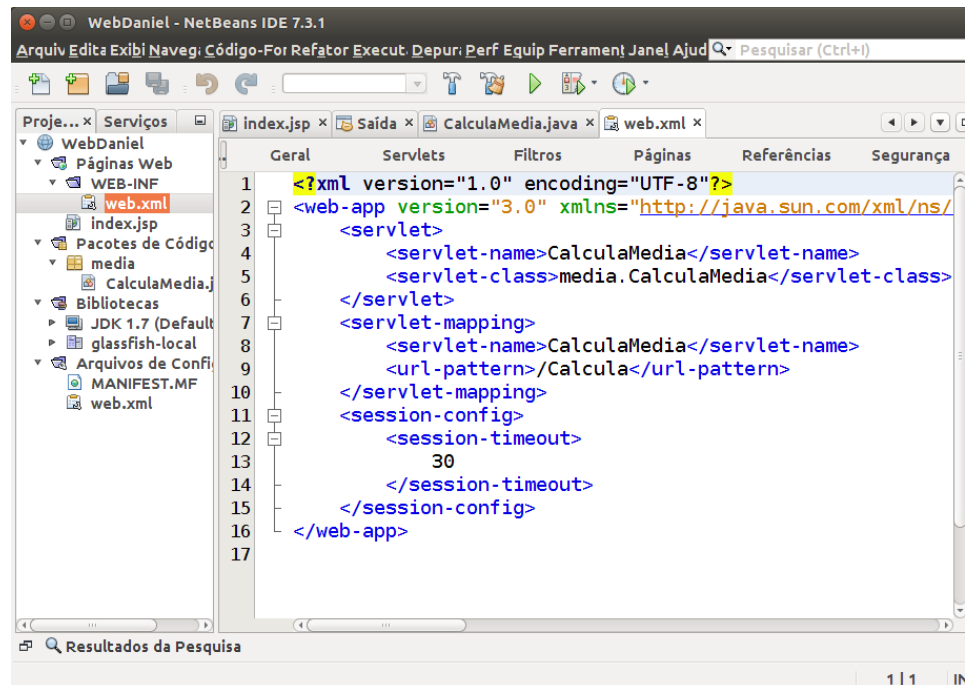
- Nome do servlet:** CalculaMedia
- Ordem de inicia** (partially visible)
- Classe do servlet:** media.CalculaMedia
- Padrão de URL:** /Calcula

Other visible fields and buttons include:

- Descrição:** (empty)
- Arquivo JSP:** (empty)
- Procurar...** (button next to Classe do servlet)
- Ir para Código** (link next to Classe do servlet)
- Procurar...** (button next to Arquivo JSP)
- Ir para o código** (link next to Arquivo JSP)
- Utilize vírgula (,) para separar múltiplos padrões.** (note below Padrão de URL)
- Parâmetros de inicialização:**
 - Table with columns: Nome do Parâmetro, Valor do Parâmetro, Descrição
 - Buttons: Adicionar..., Editar..., Remover
- Referência de atribuição de segurança:**
 - Table with columns: Nome Ref do papel, Link Ref Atribuição, Descrição
 - Buttons: Adicionar..., Editar..., Remover
- Executar como:** (empty field)

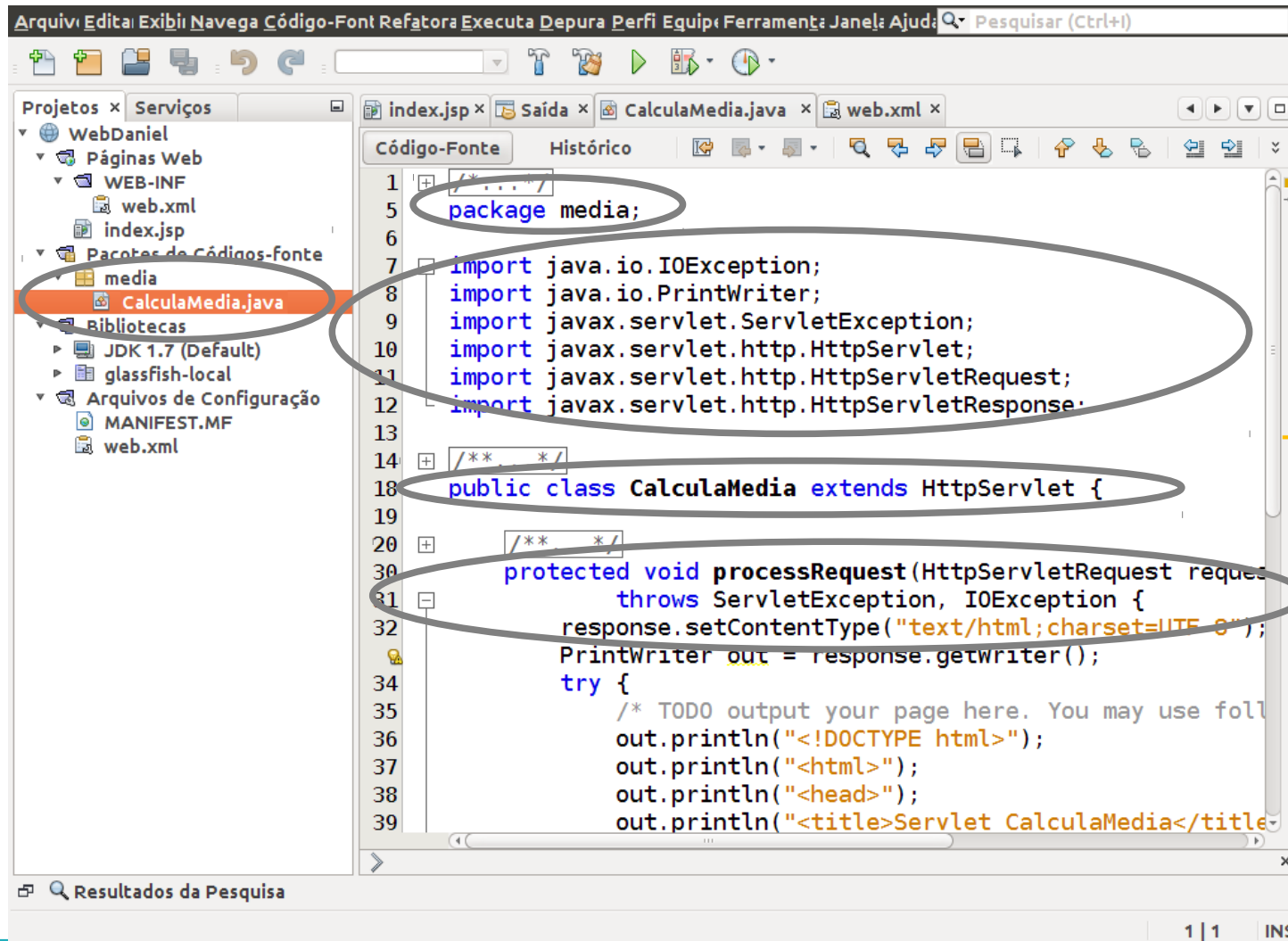
SERVLETS NO NETBEANS

- Configurando o Servlet
 - Pode ser feito também pela aba Código, que apresenta a visão em XML
 - No entanto, pelo Netbeans é mais conveniente
 - Caso não tivéssemos marcado para adicionar no descritor de implementação
 - teríamos que criar o arquivo na mão



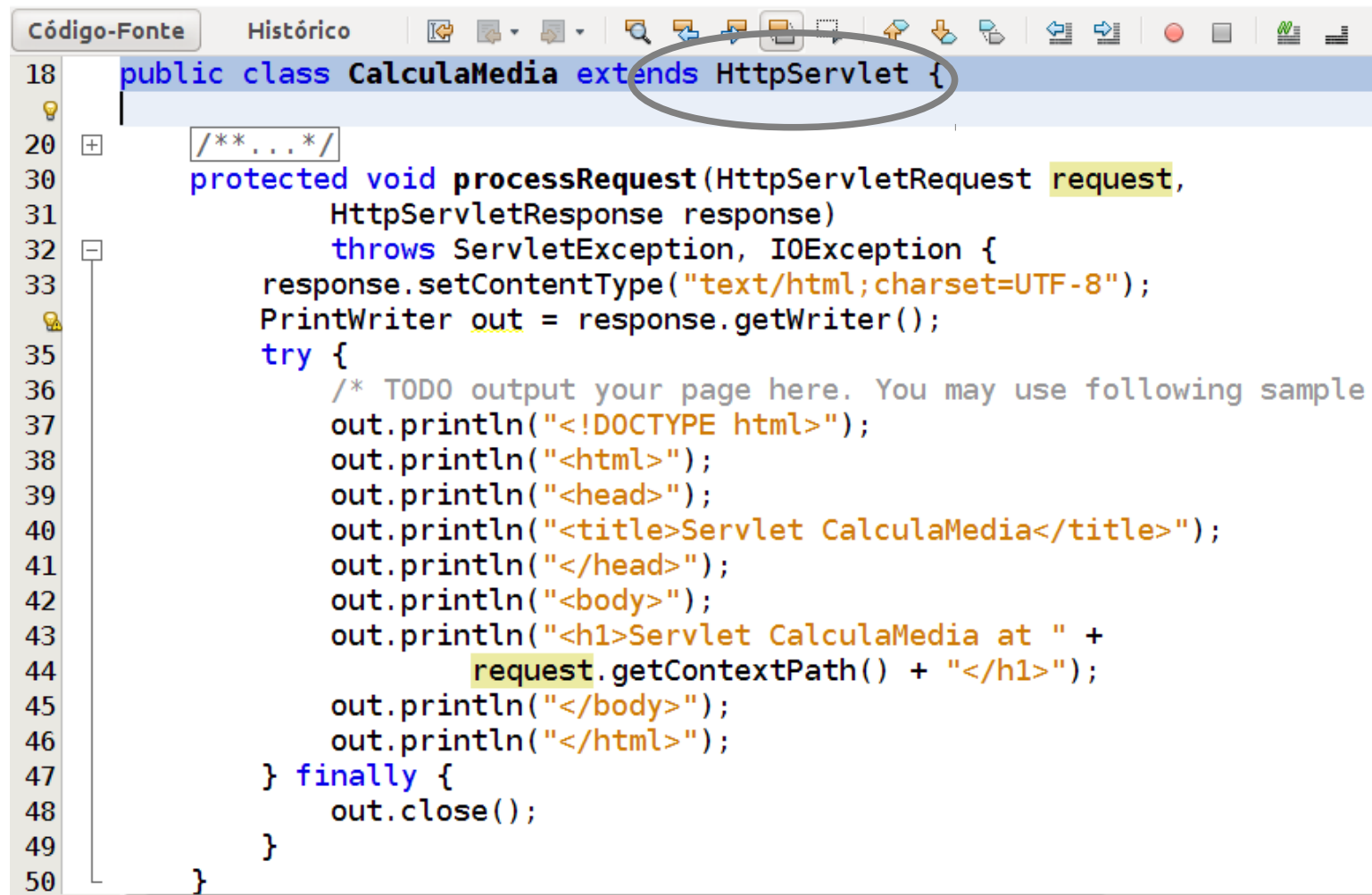
SERVLETS NO NETBEANS

- Editando o Servlet
 - Agora duplo clique no arquivo CalculaMedia.java



SERVLETS NO NETBEANS

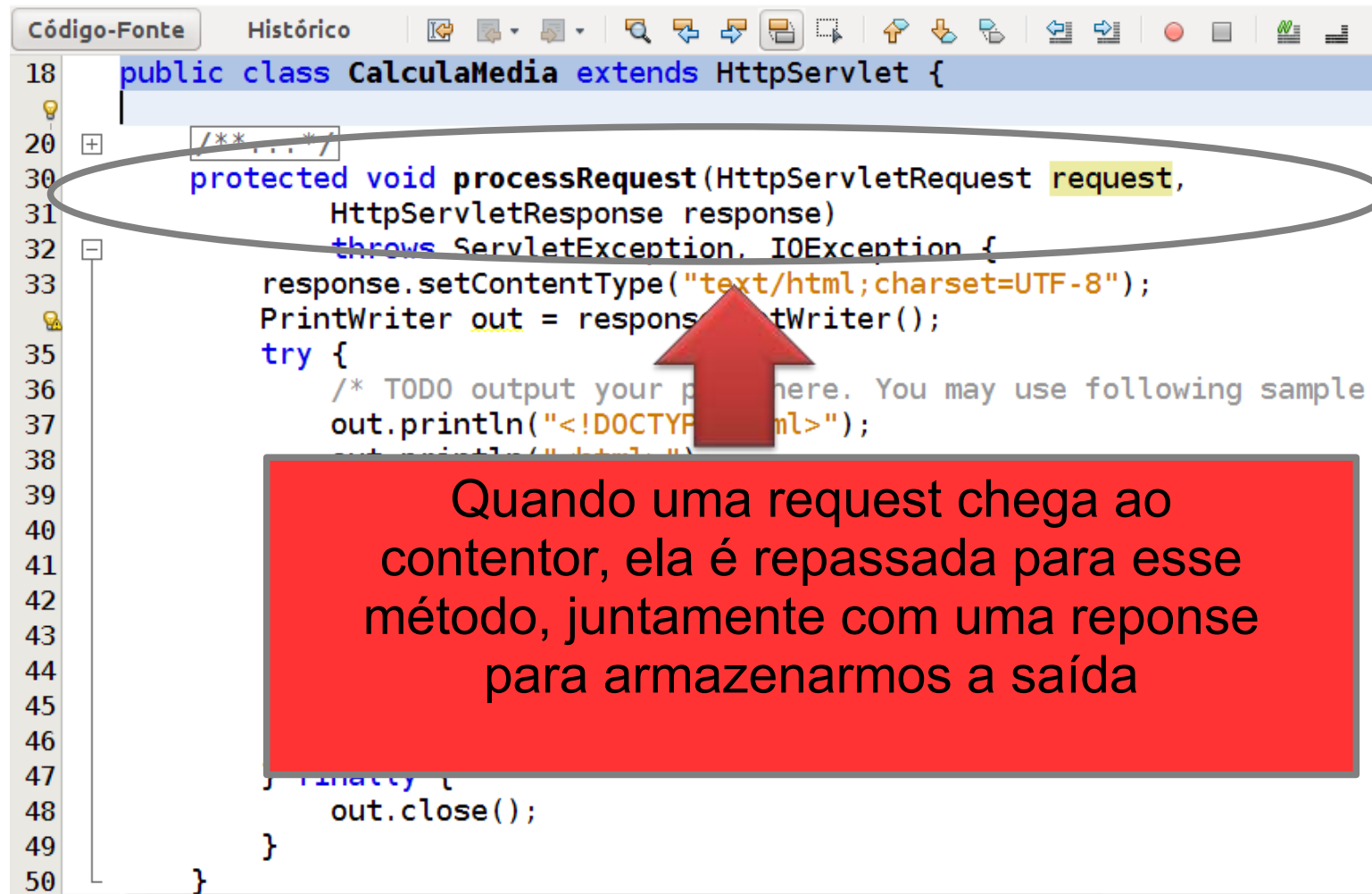
- Editando o Servlet
 - Vamos olhar o Servlet Mais de perto



```
18 public class CalculaMedia extends HttpServlet {
19
20     /** ... */
21
22     protected void processRequest(HttpServletRequest request,
23                                 HttpServletResponse response)
24         throws ServletException, IOException {
25         response.setContentType("text/html; charset=UTF-8");
26         PrintWriter out = response.getWriter();
27         try {
28             /* TODO output your page here. You may use following sample
29             out.println("<!DOCTYPE html>");
30             out.println("<html>");
31             out.println("<head>");
32             out.println("<title>Servlet CalculaMedia</title>");
33             out.println("</head>");
34             out.println("<body>");
35             out.println("<h1>Servlet CalculaMedia at " +
36                 request.getContextPath() + "</h1>");
37             out.println("</body>");
38             out.println("</html>");
39         } finally {
40             out.close();
41         }
42     }
43 }
```


SERVLETS NO NETBEANS

- Editando o Servlet
 - Vamos olhar o Servlet Mais de perto

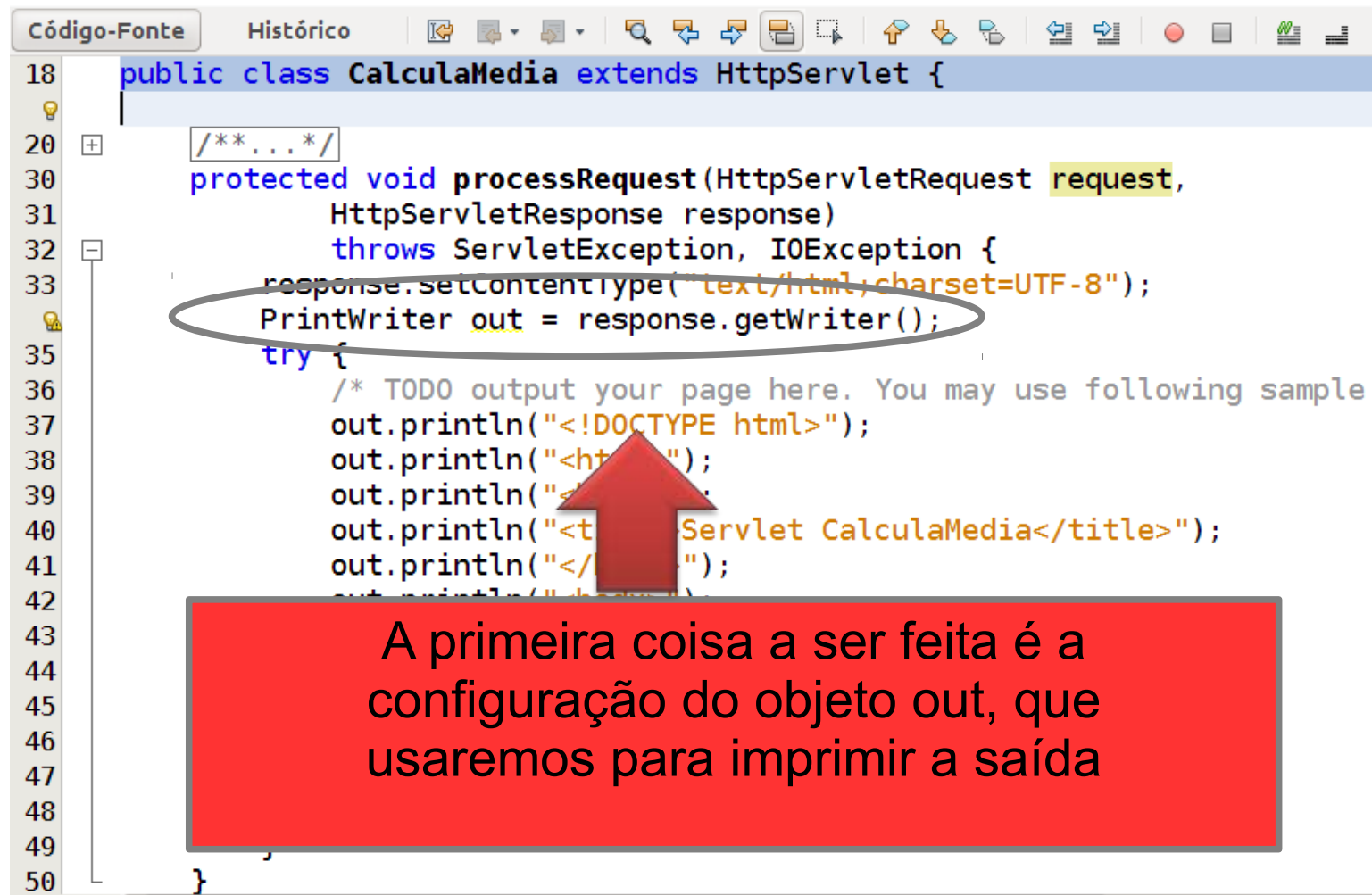


```
18 public class CalculaMedia extends HttpServlet {
19
20     /** ... */
21
22     protected void processRequest(HttpServletRequest request,
23                                 HttpServletResponse response)
24         throws ServletException, IOException {
25         response.setContentType("text/html;charset=UTF-8");
26         PrintWriter out = response.getWriter();
27         try {
28             /* TODO output your page here. You may use following sample
29             out.println("<!DOCTYPE html>");
30             out.println("<html>");
31
32         } finally {
33             out.close();
34         }
35     }
36 }
```

Quando uma request chega ao contentor, ela é repassada para esse método, juntamente com uma reponse para armazenarmos a saída

SERVLETS NO NETBEANS

- Editando o Servlet
 - Vamos olhar o Servlet Mais de perto

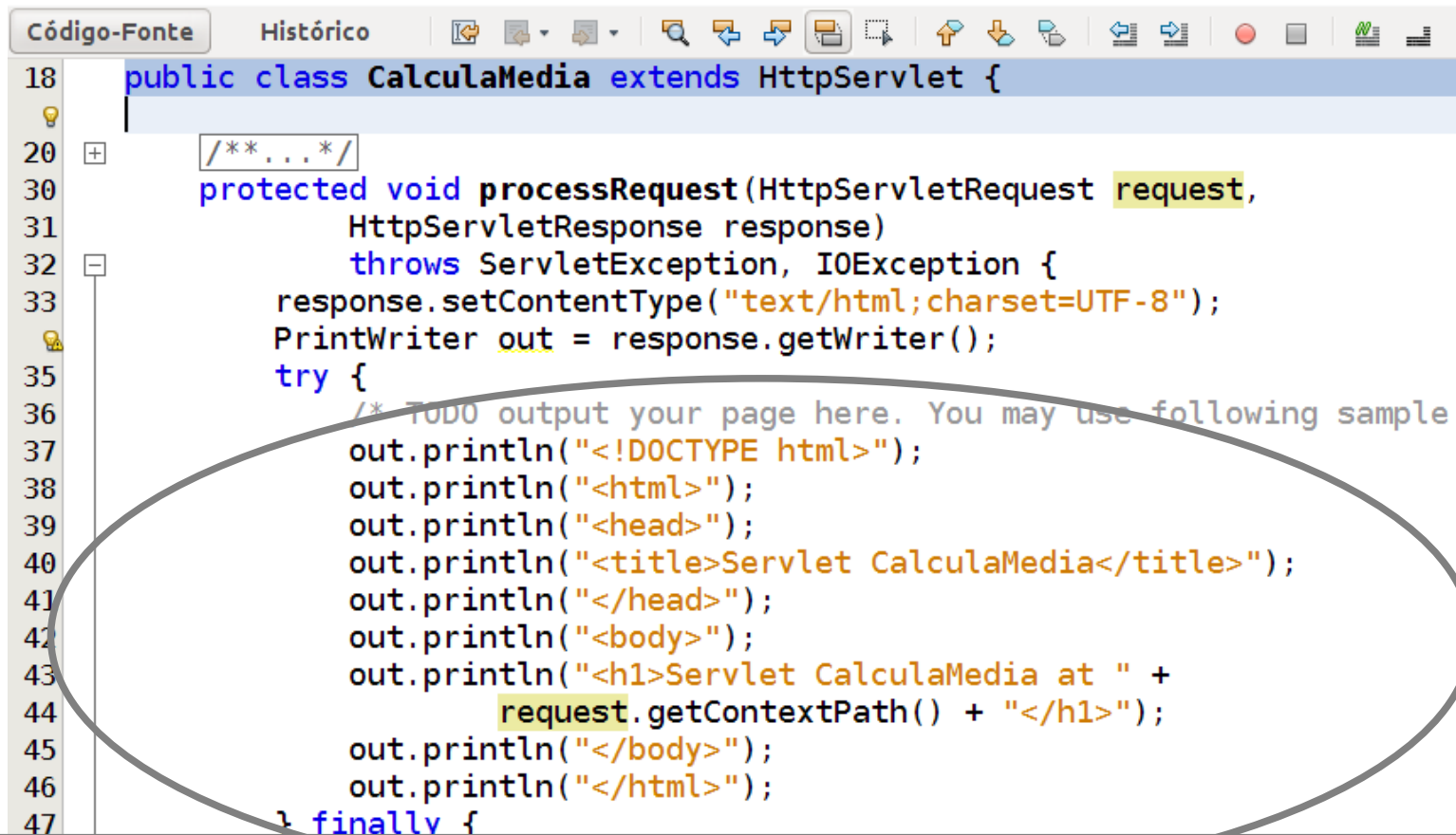


```
18 public class CalculaMedia extends HttpServlet {
19
20     /** ... */
21
22     protected void processRequest(HttpServletRequest request,
23                                   HttpServletResponse response)
24                                   throws ServletException, IOException {
25         response.setContentType("text/html; charset=UTF-8");
26         PrintWriter out = response.getWriter();
27         try {
28             /* TODO output your page here. You may use following sample
29             out.println("<!DOCTYPE html>");
30             out.println("<html>");
31             out.println("<title>");
32             out.println("<title>Servlet CalculaMedia</title>");
33             out.println("</title>");
34             out.println("<body>");
35         } catch (Exception e) {
36             // Handle exception
37         }
38     }
39 }
```

A primeira coisa a ser feita é a configuração do objeto out, que usaremos para imprimir a saída

SERVLETS NO NETBEANS

- Editando o Servlet
 - Vamos olhar o Servlet Mais de perto

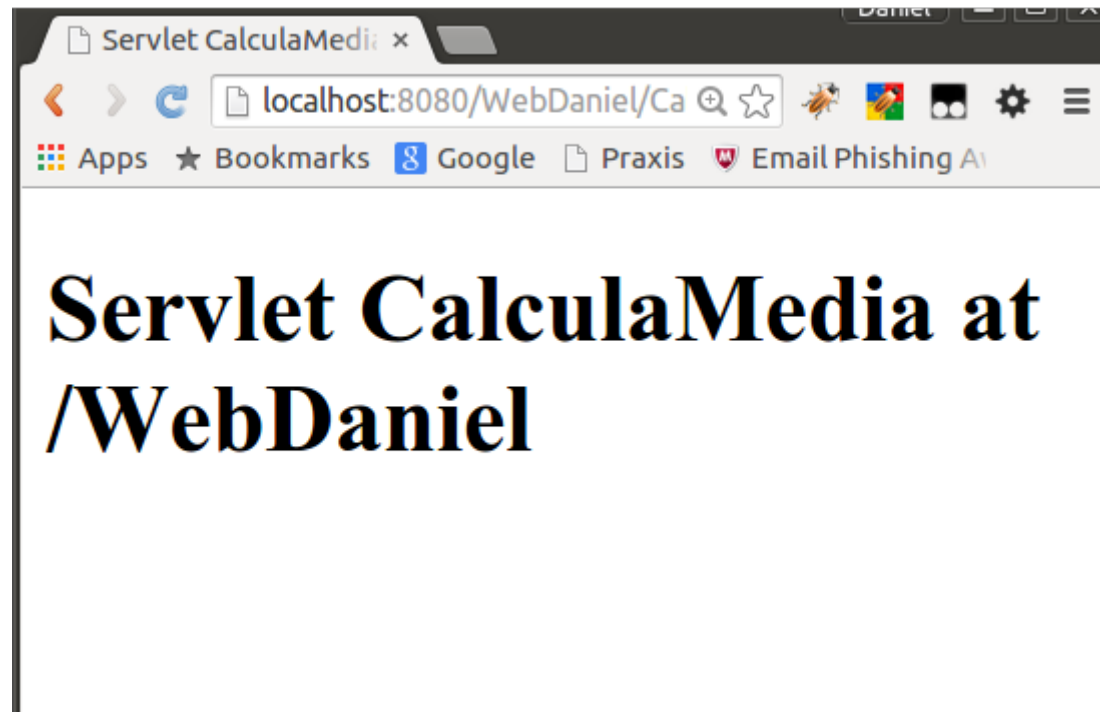


```
Código-Fonte  Histórico
18 public class CalculaMedia extends HttpServlet {
19
20     /** ... */
30     protected void processRequest(HttpServletRequest request,
31                                   HttpServletResponse response)
32                                   throws ServletException, IOException {
33         response.setContentType("text/html;charset=UTF-8");
34         PrintWriter out = response.getWriter();
35         try {
36             /* TODO output your page here. You may use following sample
37             out.println("<!DOCTYPE html>");
38             out.println("<html>");
39             out.println("<head>");
40             out.println("<title>Servlet CalculaMedia</title>");
41             out.println("</head>");
42             out.println("<body>");
43             out.println("<h1>Servlet CalculaMedia at " +
44                        request.getContextPath() + "</h1>");
45             out.println("</body>");
46             out.println("</html>");
47         } finally {
```

Aqui é onde escrevemos o código utilizando o objeto out

SERVLETS NO NETBEANS

- Editando o Servlet
 - Salve o projeto e clique no Enviar de sua página inicial para ver o que acontece



SERVLETS NO NETBEANS

- Editando o Servlet
 - Precisamos agora receber os dados do formulário
 - Falta fazer o cálculo:
 - $Media = (ValorA + ValorB) / 2$
 - Mas como pegamos os valores desse parâmetros
 - Ora, estão na requisição, ou seja, no objeto request!!
 - Para recuperar um parâmetro na requisição:
 - `request.getParameter("nome")`
 - Os nomes são “valorA” e “valorB” (conforme *inputs* no *form*!)
 - Os parâmetros sempre são retornados como Strings

SERVLETS NO NETBEANS

- Editando o Servlet

- Vamos alterar o nosso Servlet, salvar o projeto e testar para ver o que acontece

```
PrintWriter out = response.getWriter();
try {
    out.println("<!DOCTYPE html>");
    out.println("<html>");
    out.println("<head>");
    out.println("<title>Servlet CalculaMedia</title>");
    out.println("</head>");
    out.println("<body>");
    out.println("<h1>Calculo Media </h1>");

    String valorA = request.getParameter("valorA");
    String valorB = request.getParameter("valorB");

    out.println("ValorA: " + valorA);
    out.println("ValorB: " + valorB);

    out.println("</body>");
    out.println("</html>");
} finally {
    out.close();
}
```

SERVLETS NO NETBEANS

- Editando o Servlet
 - Resultado ...



OPA! Não pulou linha!

Devemos sempre lembrar que estamos
imprimindo HTML

SERVLETS NO NETBEANS

- Editando o Servlet
 - Calculando a média ...
 - Antes de fazer os cálculos, porém...
 - Temos que converter as Strings para números
 - Use ponto para separar casas
 - Existe método pronto para isso
 - Double.valueOf
 - Criar uma variável para
 - Receber o valor calculado da média
 - Imprimir na resposta

SERVLETS NO NETBEANS

- Editando o Servlet
 - Alterando o JSP para exibir a media

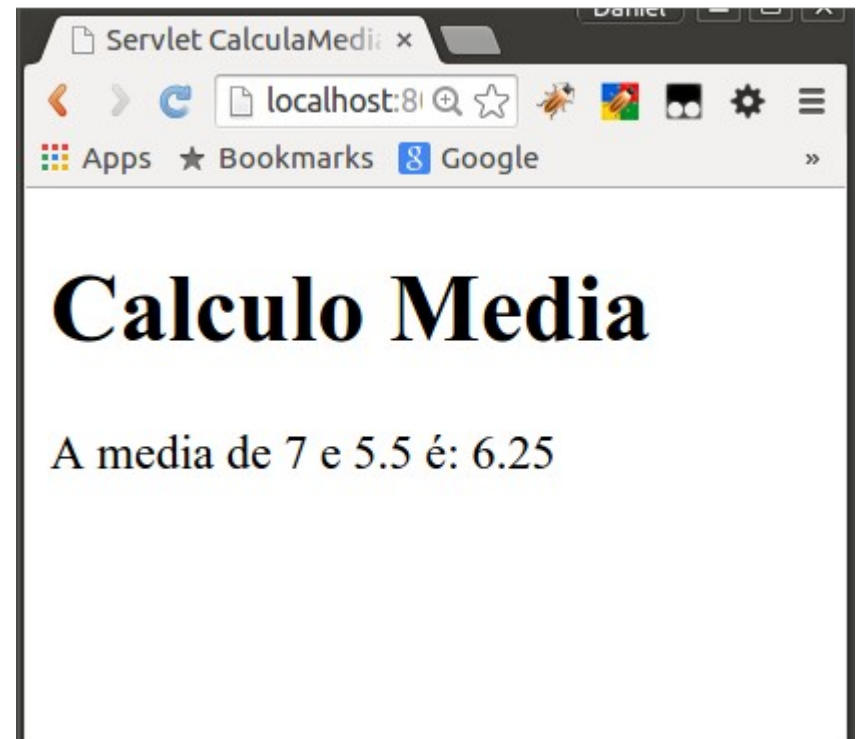
```
try {  
    out.println("<!DOCTYPE html>");  
    out.println("<html>");  
    out.println("<head>");  
    out.println("<title>Servlet CalculaMedia</title>");  
    out.println("</head>");  
    out.println("<body>");  
    out.println("<h1>Calculo Media </h1>");  
  
    String valorA = request.getParameter("valorA");  
    String valorB = request.getParameter("valorB");  
    double dValorA = Double.valueOf(valorA);  
    double dValorB = Double.valueOf(valorB);  
    double media = ( dValorA + dValorB )/2;  
    out.println("<p>A media de " + valorA + " e "  
        + valorB + " &acute;;: "  
        + media + "</p>");  
    out.println("</body>");  
    out.println("</html>");  
} finally {  
    out.close();  
}
```

SERVLETS NO NETBEANS

- Editando o Servlet
 - Resultado



A screenshot of a web browser window with the title 'Media Aritmética'. The address bar shows 'localhost:8080'. The page content includes the heading 'Digite os dados!' followed by two input fields labeled 'A:' and 'B:'. The value '7' is entered in field 'A' and '5.5' is entered in field 'B'. Below the fields is a button labeled 'Enviar!'.



A screenshot of a web browser window with the title 'Servlet CalculaMedia'. The address bar shows 'localhost:8080'. The page content includes the heading 'Calculo Media' followed by the text 'A media de 7 e 5.5 é: 6.25'.

SERVLETS NO NETBEANS

- Editando o Servlet
 - Podemos melhorar um pouco
 - Usabilidade
 - Conversão de , (vírgula) para . (ponto) como separador
 - Tratamento de erros no Servlet
 - Entrada não numérica
 - Apresentação
 - Limite de 2 casas decimais
 - Organização
 - Estruturação do código fonte: apresentação x lógica

Até a Próxima Aula!