



Desenvolvimento Web

Eclipse com Tomcat – Exemplo com Servlet

Professor Vicente Paulo de Camargo



Eclipse – Tomcat – Exemplo com Servlet

Servlet é uma classe Java que é executada em um servidor web de tal forma que reconhece as solicitações de um cliente web

Um servlet, após receber uma requisição, pode capturar parâmetros da requisição, efetuar qualquer processamento e devolver uma página HTML, XML, JSON e outros formatos para o cliente solicitante



Eclipse – Tomcat – Exemplo com Servlet

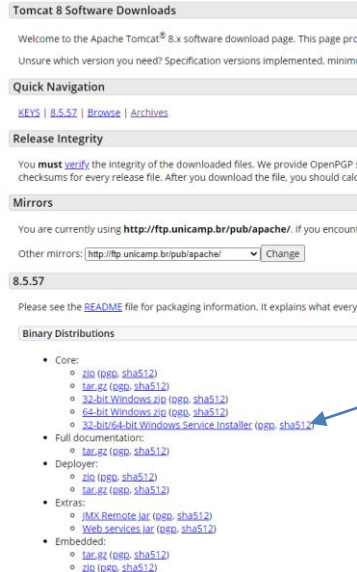
Páginas JSP (Java Server Pages)

- As páginas JSP foram criadas para resolver certas limitações dos Servlets
- O Servlet mistura lógica de negócio com processamento de requisições
- Página JSP permite estabelecer um processamento dinâmico a uma pagina web
- Uma página JSP permite códigos Java junto com HTML

Eclipse – Tomcat – Exemplo com Servlet

1. Instalar o JDK
2. Instalar o Eclipse (Versão developers JAVA EE)
3. Instalar o Tomcat (versão 8, por exemplo)

Dê preferência pela instalação com arquivo executável (<https://tomcat.apache.org/download-80.cgi>)



Tomcat 8 Software Downloads

Welcome to the Apache Tomcat® 8.x software download page. This page provides information on which version you need? Specification versions implemented, minimum Java version, etc.

Quick Navigation

[KEYS](#) | [8.5.57](#) | [Browse](#) | [Archives](#)

Release Integrity

You **must** [verify](#) the integrity of the downloaded files. We provide OpenPGP signatures and checksums for every release file. After you download the file, you should calculate the checksums and compare them with the ones provided here.

Mirrors

You are currently using <http://ftp.unicamp.br/pub/apache/>. If you encounter any problems, please contact the mirror maintainer.

Other mirrors: [http://ftp.unicamp.br/pub/apache/](#)

8.5.57

Please see the [README](#) file for packaging information. It explains what every file is for.

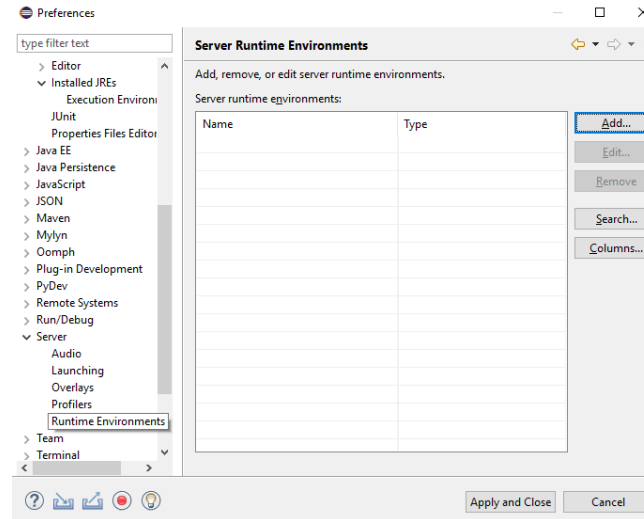
Binary Distributions

- Core:
 - [zip \(osgi, sha512\)](#)
 - [tar.gz \(osgi, sha512\)](#)
 - [32-bit Windows zip \(osgi, sha512\)](#)
 - [64-bit Windows zip \(osgi, sha512\)](#)
 - [32-bit/64-bit Windows Service Installer \(osgi, sha512\)](#)
- Full documentation:
 - [tar.gz \(osgi, sha512\)](#)
- Deployer:
 - [zip \(osgi, sha512\)](#)
 - [tar.gz \(osgi, sha512\)](#)
- Extras:
 - [jmx-remote.jar \(osgi, sha512\)](#)
 - [web-services.jar \(osgi, sha512\)](#)
- Embedded:
 - [tar.gz \(osgi, sha512\)](#)
 - [zip \(osgi, sha512\)](#)

Eclipse – Tomcat – Exemplo com Servlet

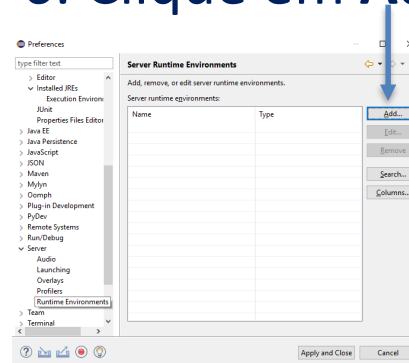
4. Abra o Eclipse

5. Acesse **window>Preferences>Server>Runtime Environments**

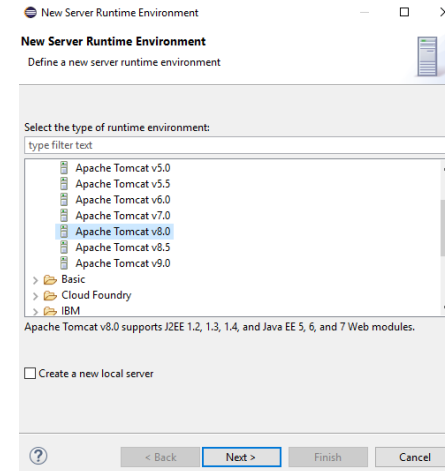


Eclipse – Tomcat – Exemplo com Servlet

6. Clique em **Add**



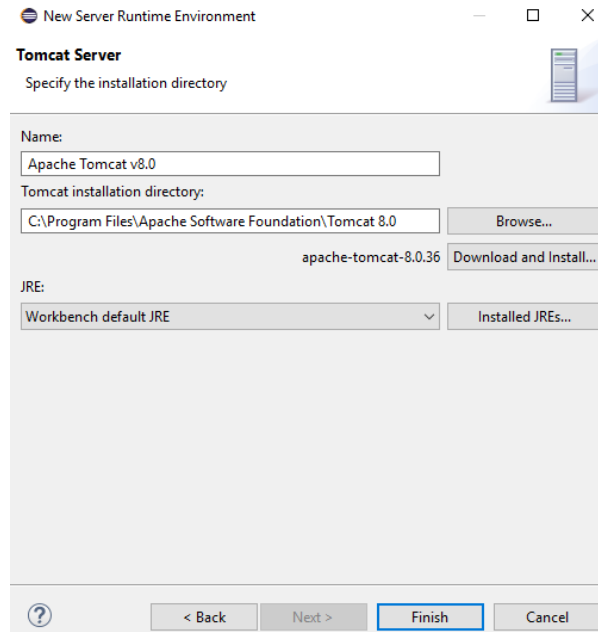
7. Selecione a versão do Tomcat que você instalou



8. Clique em **Next**

Eclipse – Tomcat – Exemplo com Servlet

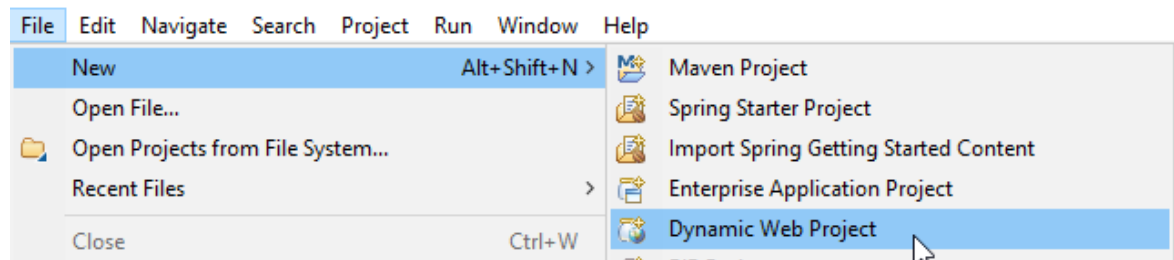
9. Na nova janela, clique em browse e selecione a pasta onde está instalado o Tomcat



10. Clique em Finish

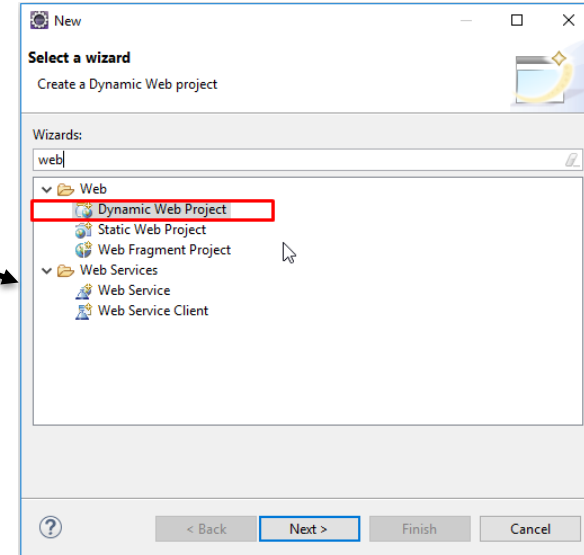
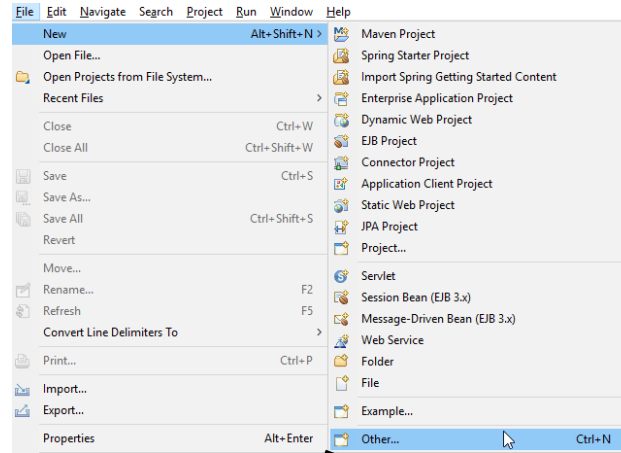
Formulários e Servlets

- Crie um novo projeto Web, conforme indicado:



Formulários e Servlets

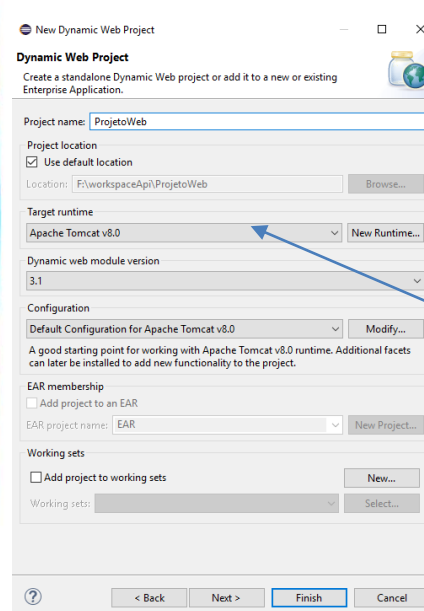
- Caso a opção anterior não esteja disponível, faça o seguinte:



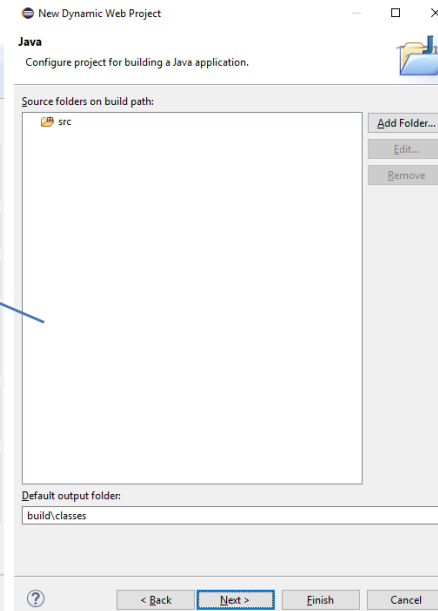
Eclipse – Tomcat – Exemplo com Servlet

Criando um projeto web no Eclipse

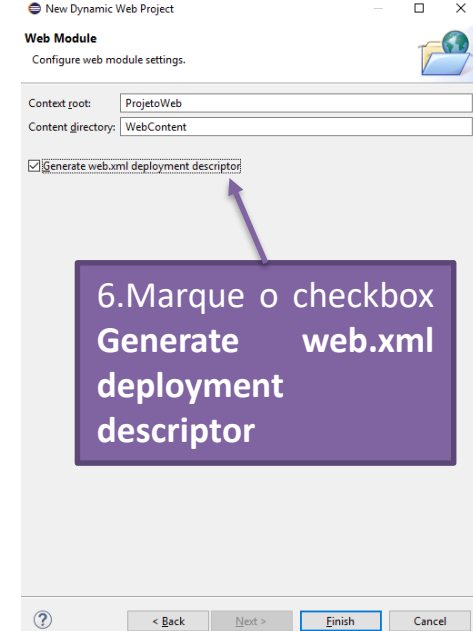
3. Na nova janela (Dynamic Web Project) informe o nome do projeto (**ProjetoWeb**), deixando as outras informações sem alterações



4. Clique em Next



5. Clique em Next



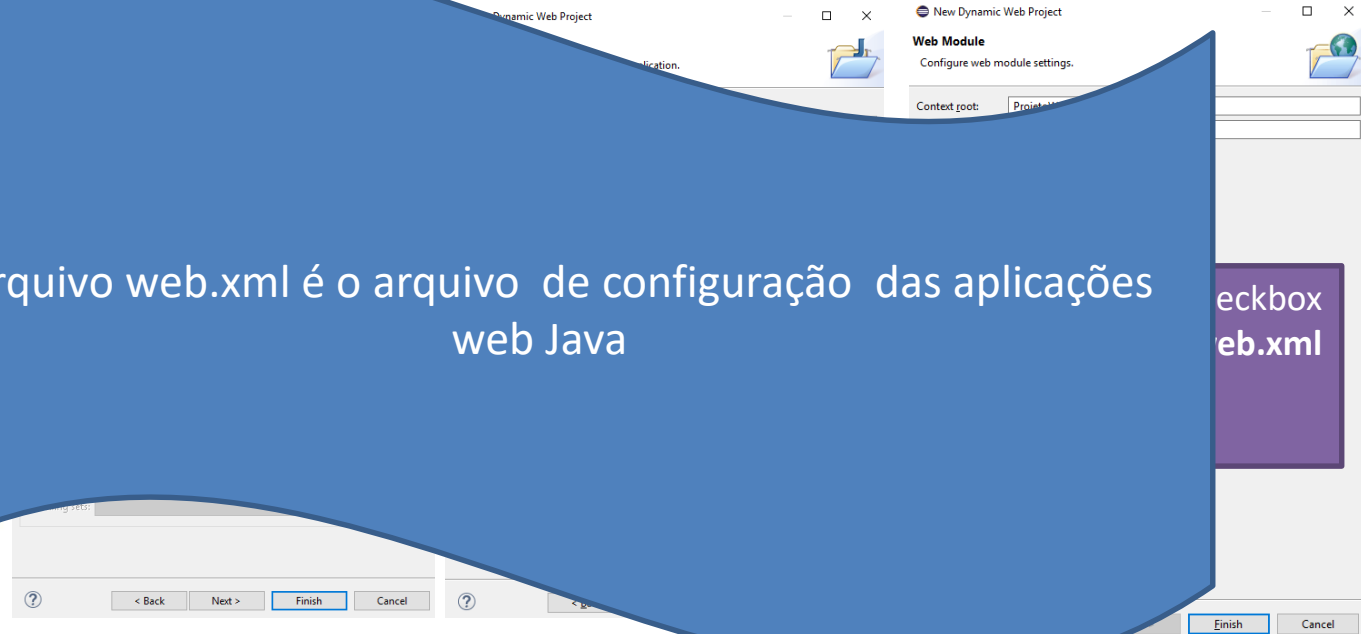
7. Clique em Finish

Eclipse – Tomcat – Exemplo com Servlet

Criando um projeto web no Eclipse

3. Na nova janela (Dynamic Web Project) informe o nome do projeto (**ProjetoWeb**),
informações sem alterações

O arquivo web.xml é o arquivo de configuração das aplicações web Java



4. Clique em Next

5. Clique em Next

7. Clique em Finish

Eclipse – Tomcat – Exemplo com Servlet

Criando um projeto web no Eclipse

Não esqueça!!

New Dynamic Web Project

Dynamic Web Project
Create a standalone Dynamic Web project or add it to a new or existing Enterprise Application.

Project name:

Project location
☒ Use default location
Location:

Target runtime

Dynamic web module version

Configuration

A good starting point for working with Apache Tomcat v8.0 runtime. Additional facets can later be installed to add new functionality to the project.

EAR membership
☐ Add project to an EAR
EAR project name:

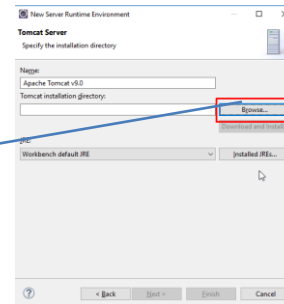
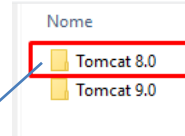
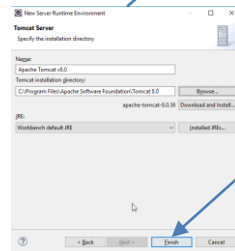
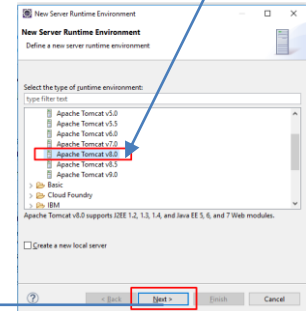
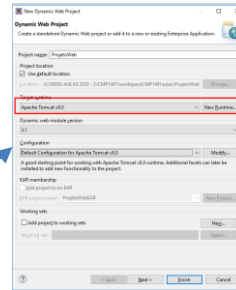
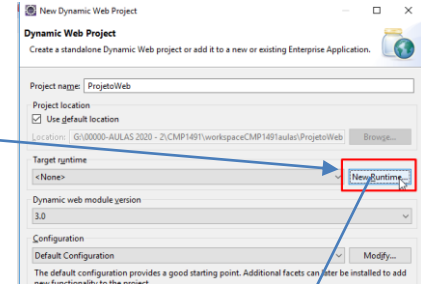
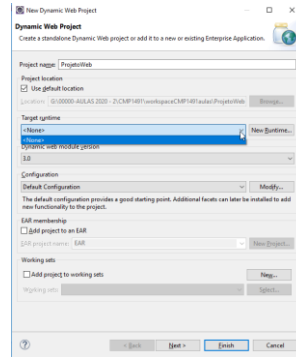
Working sets
☐ Add project to working sets
Working sets:

Software do Servidor web container (que pode ser Jboss, Jetty, Tomcat...)

Eclipse – Tomcat – Exemplo com Servlet

Criando um projeto web no Eclipse

Se não há
servidor
instalado
no Eclipse,
o que fazer?



Eclipse – Tomcat – Exemplo com Servlet

Criando um projeto web no Eclipse

Observe a estrutura do projeto web

- ▼ ProjetoWeb
 - > Deployment Descriptor: ProjetoWeb
 - > JAX-WS Web Services
 - ▼ Java Resources
 - src
 - > Libraries
 - > JavaScript Resources
 - > build
 - ▼ WebContent
 - > META-INF
 - ▼ WEB-INF
 - lib
 - web.xml

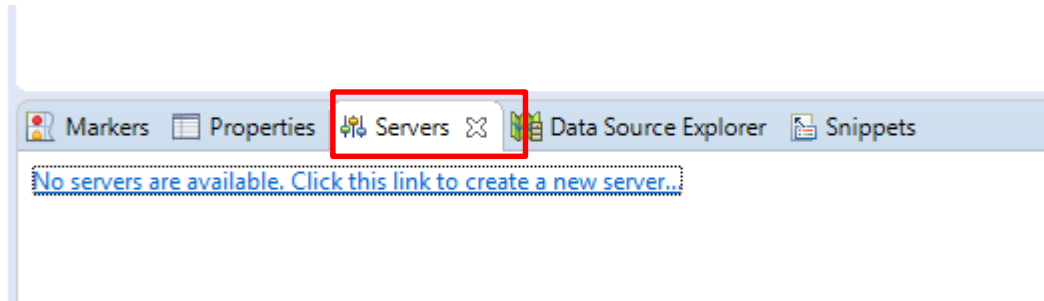
Local dos pacotes das classes java da aplicação

Local das páginas web da aplicação

Eclipse – Tomcat – Exemplo com Servlet

Criando um projeto web no Eclipse

A aba Servers do Eclipse permite visualizar o servidor web ativo. Caso não esteja ativo, ela será visualizada como ilustrado

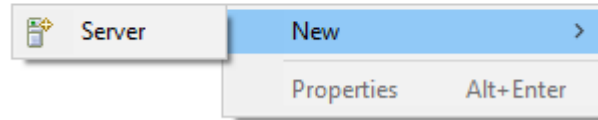
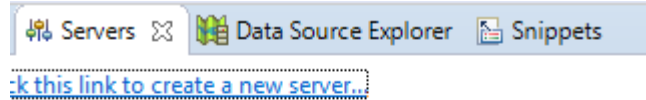


Caso a aba Severs não esteja visualizada, acesse **window>show view>servers** ou **Window>show view>other>servers**

Eclipse – Tomcat – Exemplo com Servlet

Criando um projeto web no Eclipse

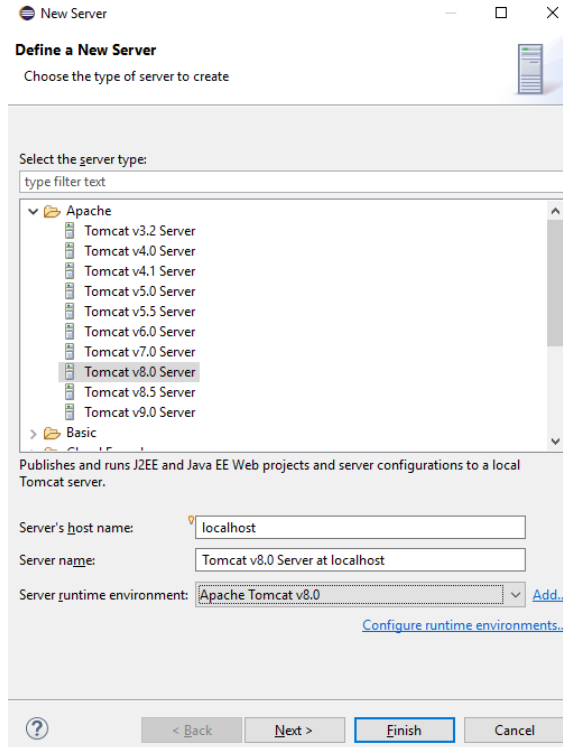
Para ativar um servidor web, clique com o botão direito do mouse dentro da aba Servers e selecione **New**



Eclipse – Tomcat – Exemplo com Servlet

Criando um projeto web no Eclipse

A janela Define a New Server será mostrada



Selecione Server runtime environment.

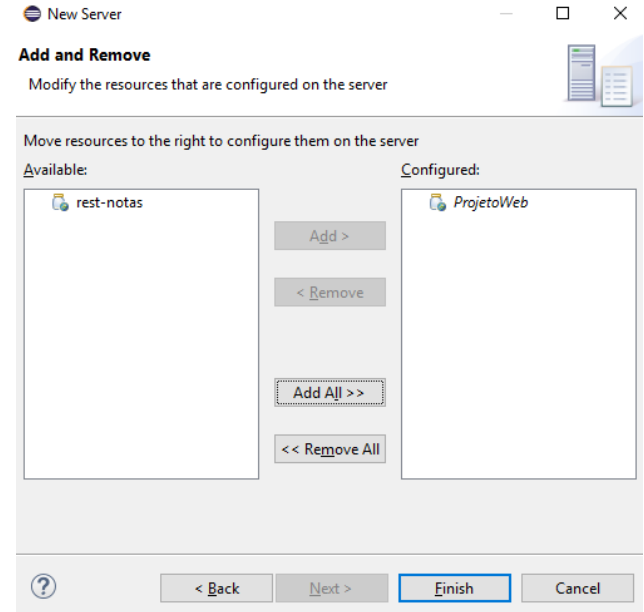
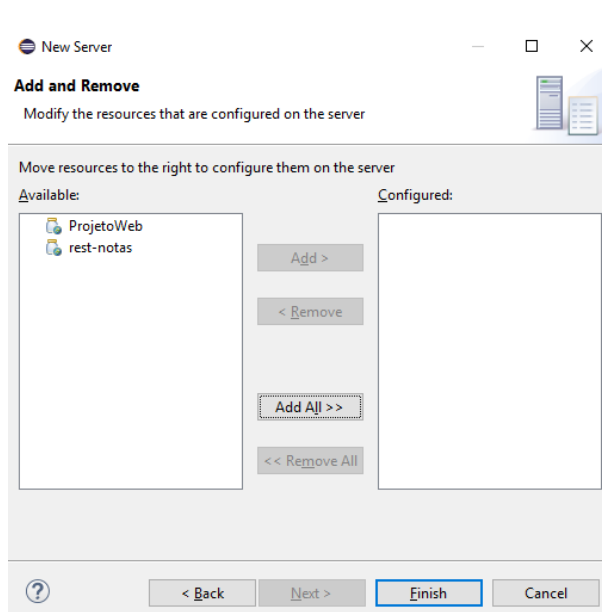
Caso essa opção esteja vazia, clique em Add para selecionar o software de servidor web instalado em seu computador

Clique em Next

Eclipse – Tomcat – Exemplo com Servlet

Criando um projeto web no Eclipse

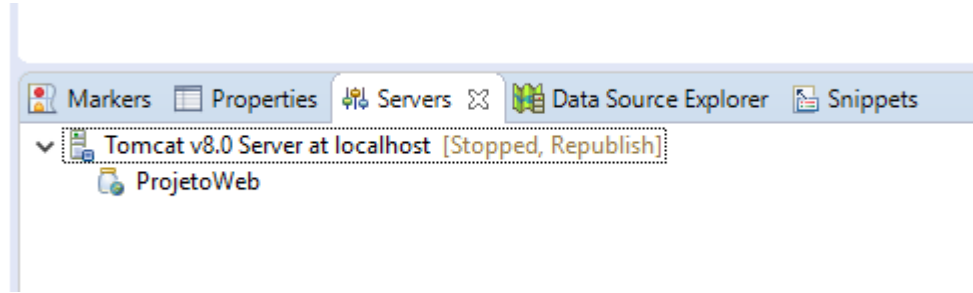
Na próxima janela, selecione o projeto **ProjetoWeb** para o quadro da direita



Eclipse – Tomcat – Exemplo com Servlet

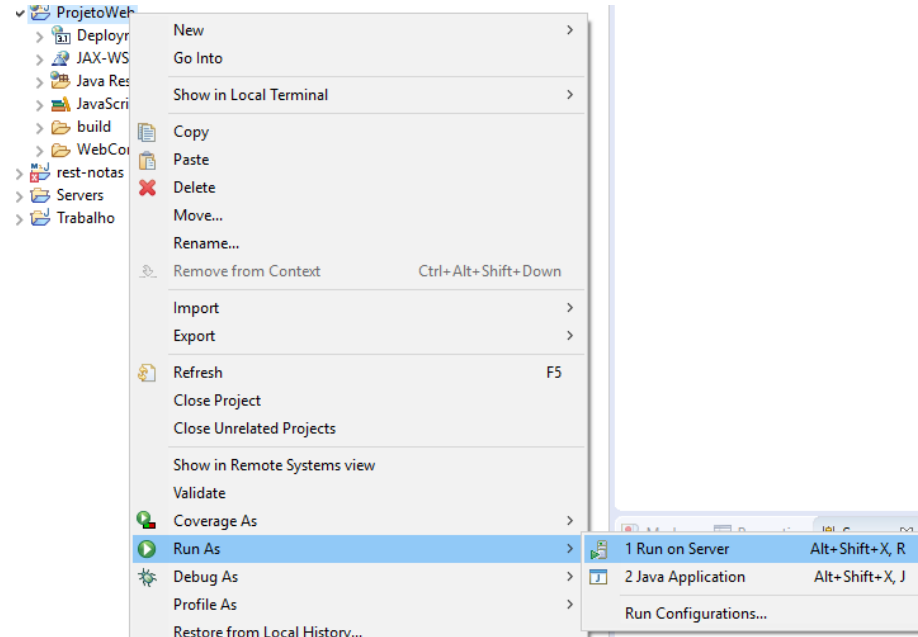
Criando um projeto web no Eclipse

O novo projeto aparece na aba Servers, como indicado:



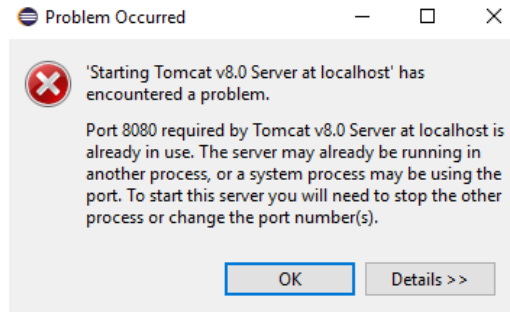
Eclipse – Tomcat – Exemplo com Servlet

Uma das maneiras de iniciar o servidor do Tomcat é utilizar o botão direito sobre o nome do projeto, selecionar Run As e, depois, Run on Server. Na próxima janela, selecione Finish



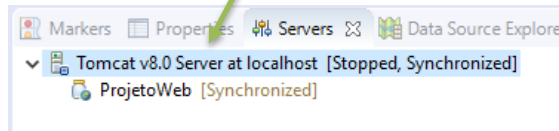
Eclipse – Tomcat – Exemplo com Servlet

Às vezes, poderá ocorrer um erro semelhante ao indicado a seguir:



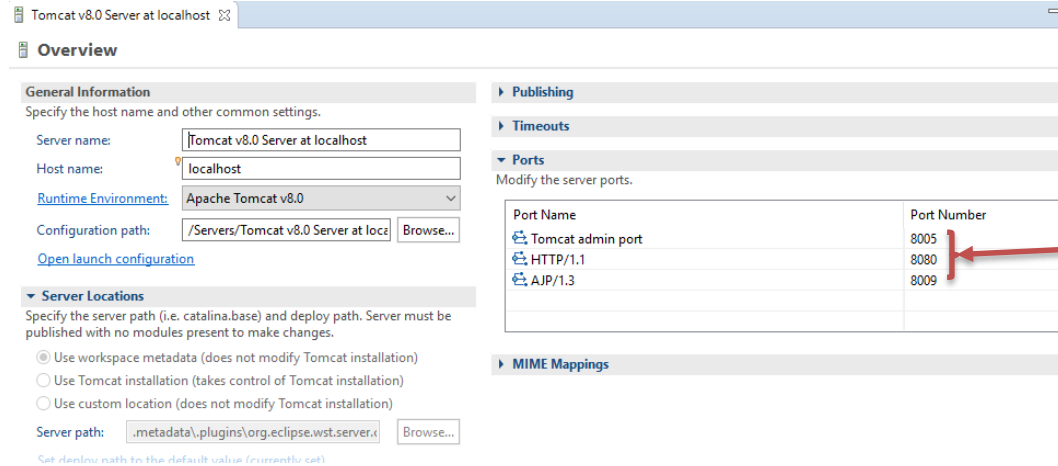
É um erro de incompatibilidade de portas

Neste caso, efetue um duplo clique no nome do servidor do Tomcat na aba Servers



Eclipse – Tomcat – Exemplo com Servlet

A janela de configuração do servidor do Tomcat será mostrada



Tomcat v8.0 Server at localhost

Overview

General Information
Specify the host name and other common settings.

Server name: Tomcat v8.0 Server at localhost

Host name: localhost

Runtime Environment: Apache Tomcat v8.0

Configuration path: /Servers/Tomcat v8.0 Server at localhost/conf/ Browse...

[Open launch configuration](#)

Server Locations
Specify the server path (i.e. catalina.base) and deploy path. Server must be published with no modules present to make changes.

☒ Use workspace metadata (does not modify Tomcat installation)
☐ Use Tomcat installation (takes control of Tomcat installation)
☐ Use custom location (does not modify Tomcat installation)

Server path: .metadata\plugins\org.eclipse.wst.server.ui\ Browse...

Ports
Modify the server ports.

Port Name	Port Number
Tomcat admin port	8005
HTTP/1.1	8080
AJP/1.3	8009

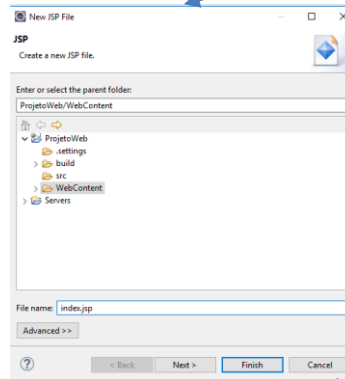
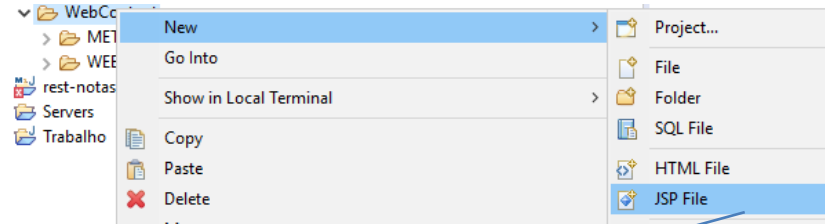
MIME Mappings

Altere os valores indicados para as portas

Informe, por exemplo, os respectivos valores: 8006, 8282 e 8010
Pressione CTRL+S para salvar
Feche o arquivo de configuração do Servidor Tomcat

Eclipse – Tomcat – Exemplo com Servlet

Para avaliar a execução do ProjetoWeb, crie a página **index.jsp**
Para isso, dê um clique com o botão direito sobre a pasta **WebContent** e selecione as opções indicadas.



Informe **index.jsp** e confirme

Eclipse – Tomcat – Exemplo com Servlet

O eclipse cria uma página jsp básica, como ilustra a figura:

```
index.jsp
1  <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2    pageEncoding="ISO-8859-1"%>
3  <!DOCTYPE html>
4  <html>
5  <head>
6    <meta charset="ISO-8859-1">
7    <title>Insert title here</title>
8  </head>
9  <body>
10
11 </body>
12 </html>
```

Não se preocupe
com esse conteúdo

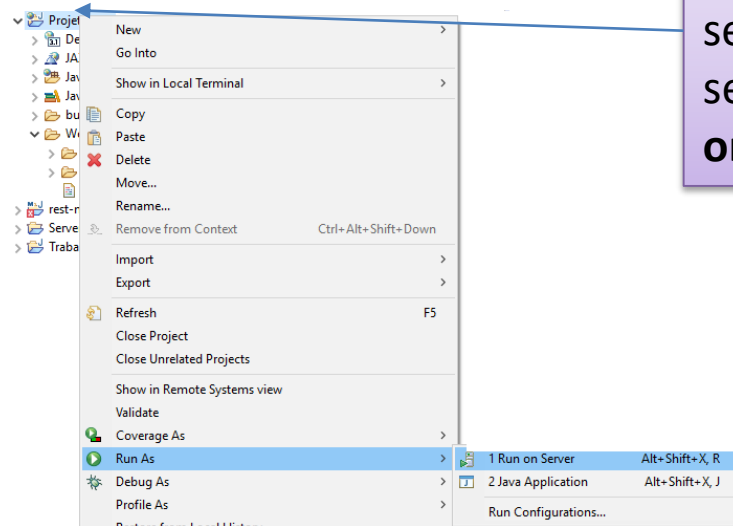
Altere apenas as linhas 7 e 10, como indicado. Salve esse arquivo

```
index.jsp
1  <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2    pageEncoding="ISO-8859-1"%>
3  <!DOCTYPE html>
4  <html>
5  <head>
6    <meta charset="ISO-8859-1">
7    <title>jsp</title>
8  </head>
9  <body>
10 Primeira página JSP
11 </body>
12 </html>
```

Eclipse – Tomcat – Exemplo com Servlet

Para executar a aplicação, utilize o esquema:

Botão direito sobre o nome do projeto e selecione **Run As**. Em seguida, selecione **Run on Server**



Eclipse – Tomcat – Exemplo com Servlet

Um navegador será apresentado dentro do ambiente do Eclipse, como ilustra a figura:



Portanto, o ambiente está corretamente configurado
A aplicação web está, portanto, funcionando, pois visualizou corretamente o conteúdo da página index.jsp

Observe como foi composta a url da aplicação:
http://localhost:8282/ProjetoWeb



Eclipse – Tomcat – Exemplo com Servlet

Criando um Servlet

A partir da especificação 3.0, correspondente ao Tomcat 7, é possível utilizar **annotations** em um Servlet.

Annotation é um recurso da Plataforma Java que permite a comunicação com outras bibliotecas e obter recursos complementares. São indicadas com um @ seguido de identificadores específicos

Exemplo: Um **@Entity** antes da definição de uma classe faz com a plataforma crie uma tabela correspondente no banco de dados

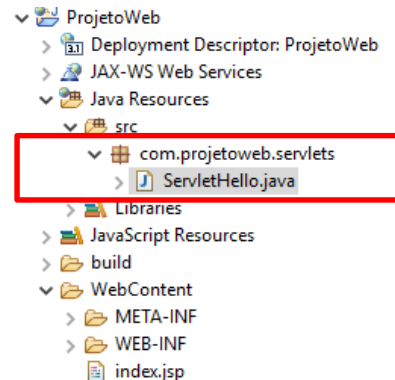
Eclipse – Tomcat – Exemplo com Servlet

Criando um Servlet

Para criar um Servlet, basta criar uma classe em um pacote da aplicação.

Crie o pacote **com.projetoweb.servlets**

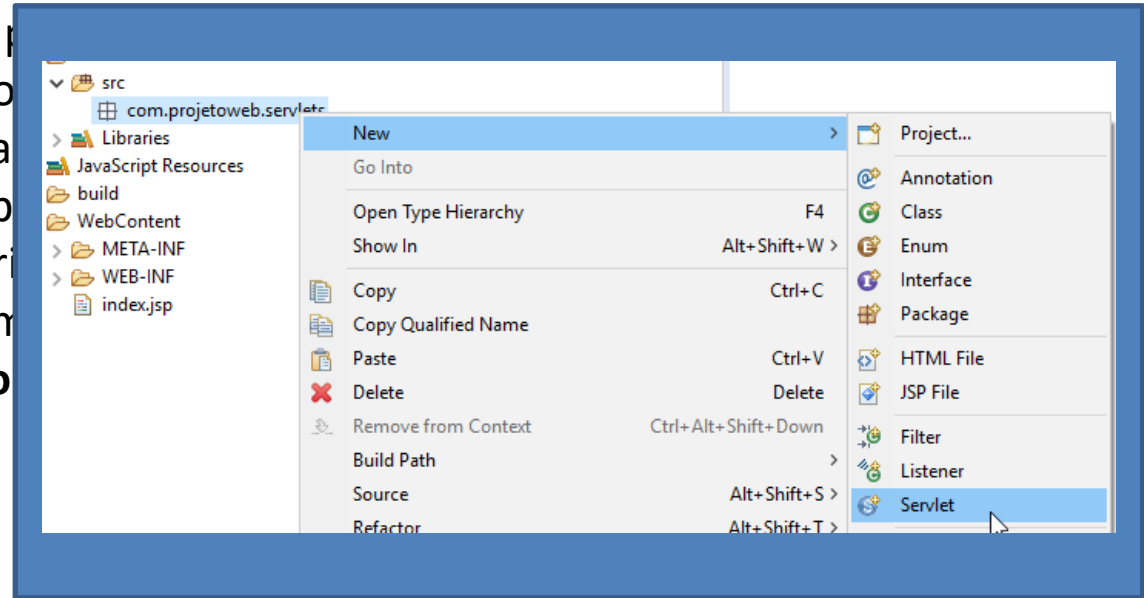
Em seguida, crie o Servlet **ServletHello.java** no pacote **com.projetoweb.servlets**, como está ilustrado a seguir:



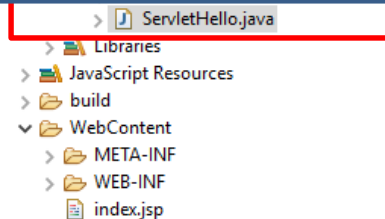
Eclipse – Tomcat – Exemplo com Servlet

Criando um Servlet

A p
po
Pa
ap
Cri
En
co



a



Eclipse – Tomcat – Exemplo com Servlet

Criando um Servlet

Observe o código do Servlet criado:

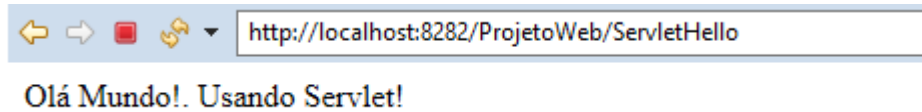
```
1 package com.projeto.web.servlets;
2
3 import java.io.IOException;
4
5
6
7
8
9
10 /**
11  * Servlet implementation class ServletHello
12  */
13 @WebServlet("/ServletHello")
14 public class ServletHello extends HttpServlet {
15     private static final long serialVersionUID = 1L;
16
17     /**
18      * @see HttpServlet#HttpServlet()
19      */
20     public ServletHello() {
21         super();
22         // TODO Auto-generated constructor stub
23     }
24
25     /**
26      * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
27      */
28     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
29         // TODO Auto-generated method stub
30         response.getWriter().print("Olá Mundo!. Usando Servlet!");
31     }
32
33     /**
34      * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
35      */
36     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
37         // TODO Auto-generated method stub
38         doGet(request, response);
39     }
40
41 }
```

A annotation `@WebServlet`, linha 10, configura o Servlet **ServletHello.java** com a url `"/ServletHello"`. Dessa forma, para acessar esse Servlet, basta digitar a url: **`http://localhost:8282/ProjetoWeb/ServletHello`**

Eclipse – Tomcat – Exemplo com Servlet

Criando um Servlet

Execute o servlet ServletHello.





Eclipse – Tomcat – Exemplo com Servlet

Criando um Servlet

URL com parâmetros

Uma URL pode receber parâmetros, os quais podem ser utilizados em um Servlet

Os parâmetros são informados em um par **chave=valor**

O primeiro parâmetro é iniciado com um “?” e os demais são iniciados (e separados) por um “&”

Exemplo:

`http://localhost:8282/ProjetoExemplo?nome=Carlos Pereira&idade=30`

Para obter um parâmetro no Servlet utiliza-se o método **`getParameter(“nomeDoParâmetro”)`**

Que sempre retorna uma String

Eclipse – Tomcat – Exemplo com Servlet

Criando um Servlet

URL com parâmetros

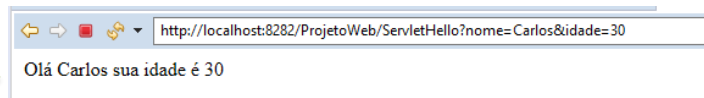
Altere a classe Servlet **ServletHello**, conforme indicado no trecho a seguir:



Note que o nome dos parâmetros ficam entre aspas no `getParameter`

```
29- protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
30-     // TODO Auto-generated method stub  
31-     String nome = request.getParameter("nome");  
32-     int idade = Integer.parseInt(request.getParameter("idade"));  
33-     response.setContentType("text/html;charset=UTF-8");  
34-     response.getWriter().print("Olá "+nome+" sua idade é "+idade);  
35- }
```

Observe o servlet em execução com os parâmetros





Eclipse – Tomcat – Exemplo com Servlet

Criando um Servlet

Passagem de parâmetros por formulário

- Um formulário é utilizado pelo cliente para cadastrar ou atualizar informações em um banco de dados
- Um formulário utiliza, geralmente, o método POST
- A vantagem do POST sobre o método GET está no fato de as informações serem enviadas no corpo da mensagem HTTP
- Ao contrário do GET cujos dados são enviados junto com a URL
- Se existir alguma senha para ser enviada via GET, o seu conteúdo irá aparecer na URL
- O formulário possui o atributo **action** que indica o caminho da página para a qual as informações serão enviadas

Eclipse – Tomcat – Exemplo com Servlet

Criando um Servlet

Passagem de parâmetros por formulário

Crie uma nova página, com o nome **pag01.html**, na pasta WebContent

Codifique a página como indicado a seguir

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="ISO-8859-1">
5   <title>Insert title here</title>
6 </head>
7 <body>
8
9   <form method="post" action="http://localhost:8282/ServletHello01">
10     Nome: <input type="text" name="nome" />
11
12     Idade: <input type="text" name="idade" />
13
14     <input type="submit" name="enviar" value="Enviar"/>
15   </form>
16
17 </body>
18 </html>
```


Eclipse – Tomcat – Exemplo com Servlet

Criando um Servlet

Passagem de parâmetros por formulário

Copie a classe **ServletHello.java** para **ServletHello01.java**

Altere o método **doGet** para **doPost** e vice-versa. Ajuste o código como indicado:

```
1 package com.projetoweb.servlets;
2
3 import java.io.IOException;
4
5
6
7
8
9
10
11 /**
12  * Servlet implementation class ServletHello
13  */
14 @WebServlet("/ServletHello01")
15 public class ServletHello01 extends HttpServlet {
16     private static final long serialVersionUID = 1L;
17
18     /**
19      * @see HttpServlet#HttpServlet()
20      */
21     public ServletHello01() {
22         super();
23         // TODO Auto-generated constructor stub
24     }
25
26     /**
27      * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
28      */
29     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
30         // TODO Auto-generated method stub
31         String nome = request.getParameter("nome");
32         int idade = Integer.parseInt(request.getParameter("idade"));
33         response.setContentType("text/html;charset=UTF-8");
34         response.getWriter().print("Recebendo via post - "+nome+" sua idade é "+idade);
35     }
36
37     /**
38      * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
39      */
40     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
41         // TODO Auto-generated method stub
42         doPost(request, response);
43     }
44
45 }
```

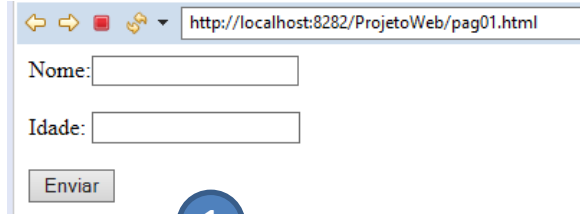
Eclipse – Tomcat – Exemplo com Servlet

Criando um Servlet

Passagem de parâmetros por formulário

Salve as mudanças

Execute a aplicação, informando a página **pag01.html** na URL, como indicado na figura

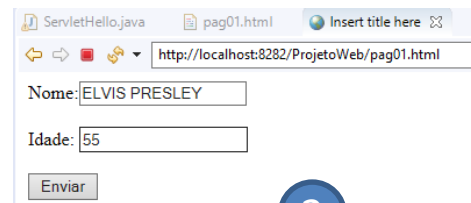


Nome:

Idade:

Enviar

1

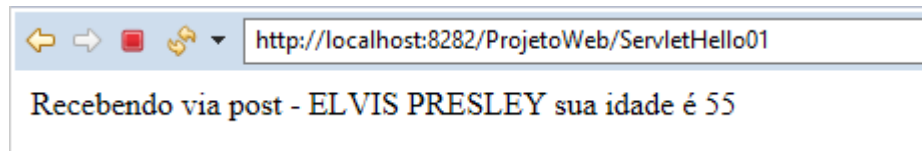


Nome:

Idade:

Enviar

2



Recebendo via post - ELVIS PRESLEY sua idade é 55

3

Eclipse – Tomcat – Exemplo com Servlet

Atividade prática

01.Utilize o formulário da página html-05.html da unidade anterior como referência. Crie o projeto java web **FormServlet**. Esse projeto deve ter um Servlet para receber a requisição de um formulário de uma página html, contendo os campos: nome, cpf, telefone e salário. O Servlet deve validar as informações do formulário considerando as seguintes regras:

- a)O nome deve ser obrigatório
- b)Deve-se validar o cpf (pesquisar como é realizado o cálculo dessa validação)
- c)O telefone só deve receber os dígitos de 0 a 9.
- d)O salário só deve receber os dígitos de 0 a 9 e a vírgula decimal.
- e)O Servlet deve apresentar uma mensagem indicando os erros de validação
- e)Se os dados estiverem corretos, apresentar o conteúdo dos dados, sendo que o nome deve apresentar todas as letras minúsculas, com a primeira letra de cada parte do nome em maiúscula

Nota: UTILIZE MÉTODOS ESPECÍFICOS PARA CADA UMA DAS VALIDAÇÕES E TAMBÉM PARA AJUSTAR O NOME

FIM

