

SOAP – Exemplo Prático

Usando Eclipse e Axis 2

Requisitos

- Eclipse versão J2EE
 - Estou usando o eclipse Version: Neon.3 Release (4.6.3)
- Tomcat
 - Estou usando a versão 9.x

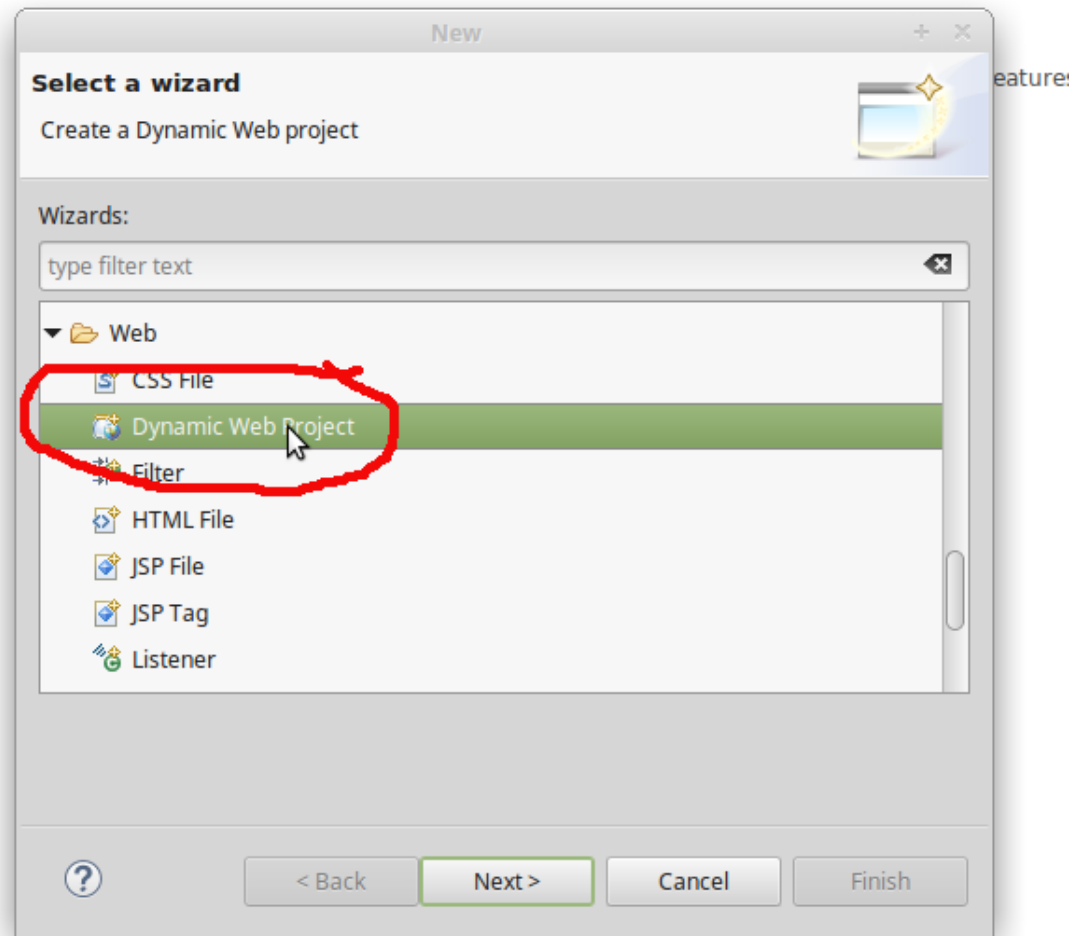
Preparação do Ambiente

- Instale o tomcat e configure na sua IDE
 - Adicione o container (servers)

Siga as imagens

- Busquei incluir pouco texto e muita imagem a fim de fazer funcionar um exemplo simples de SOAP

Crie um projeto java web dinâmico



Dê nome ao seu projeto

- O meu projeto se chama
 - ExemploSOAPWeb3
- Se quiser e já estiver instalado, selecione seu container
 - Vamos usar o tomcat
- Depois “Next”

New Dynamic Web Project

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

Project name: **ExemploSOAPWeb3**

Project location

☒ Use default location

Location: /home/tiago/workspace2/ExemploSOAPWeb3 [Browse...](#)

Target runtime

<None> [New Runtime...](#)

Dynamic web module version

3.0

Configuration

Default Configuration [Modify...](#)

The default configuration provides a good starting point. Additional facets can later be installed to add new functionality to the project.

EAR membership

☐ Add project to an EAR

EAR project name: EAR [New Project...](#)

Working sets

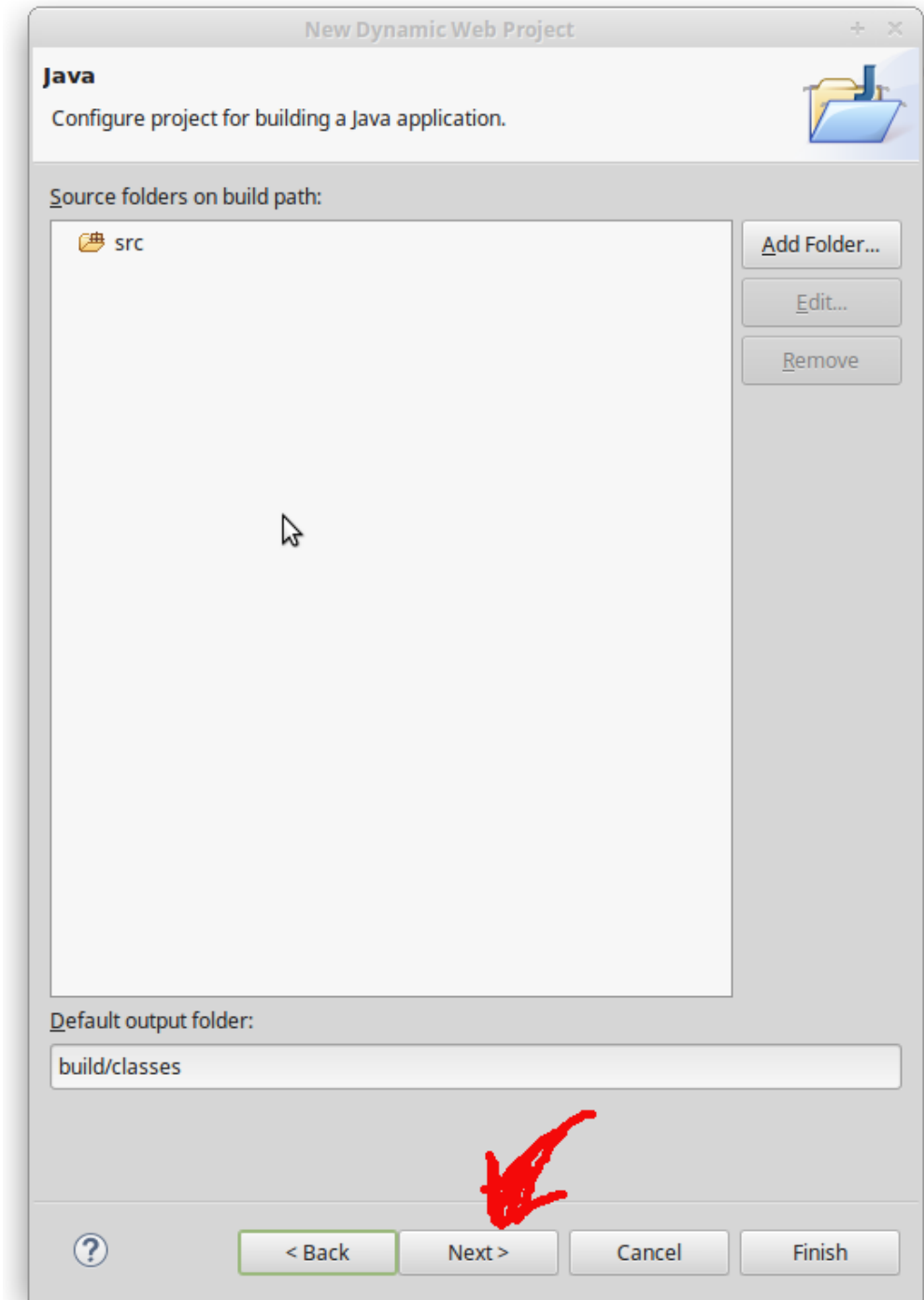
☐ Add project to working sets [New...](#)

Working sets: [Select...](#)

[? < Back Next > Cancel Finish](#)

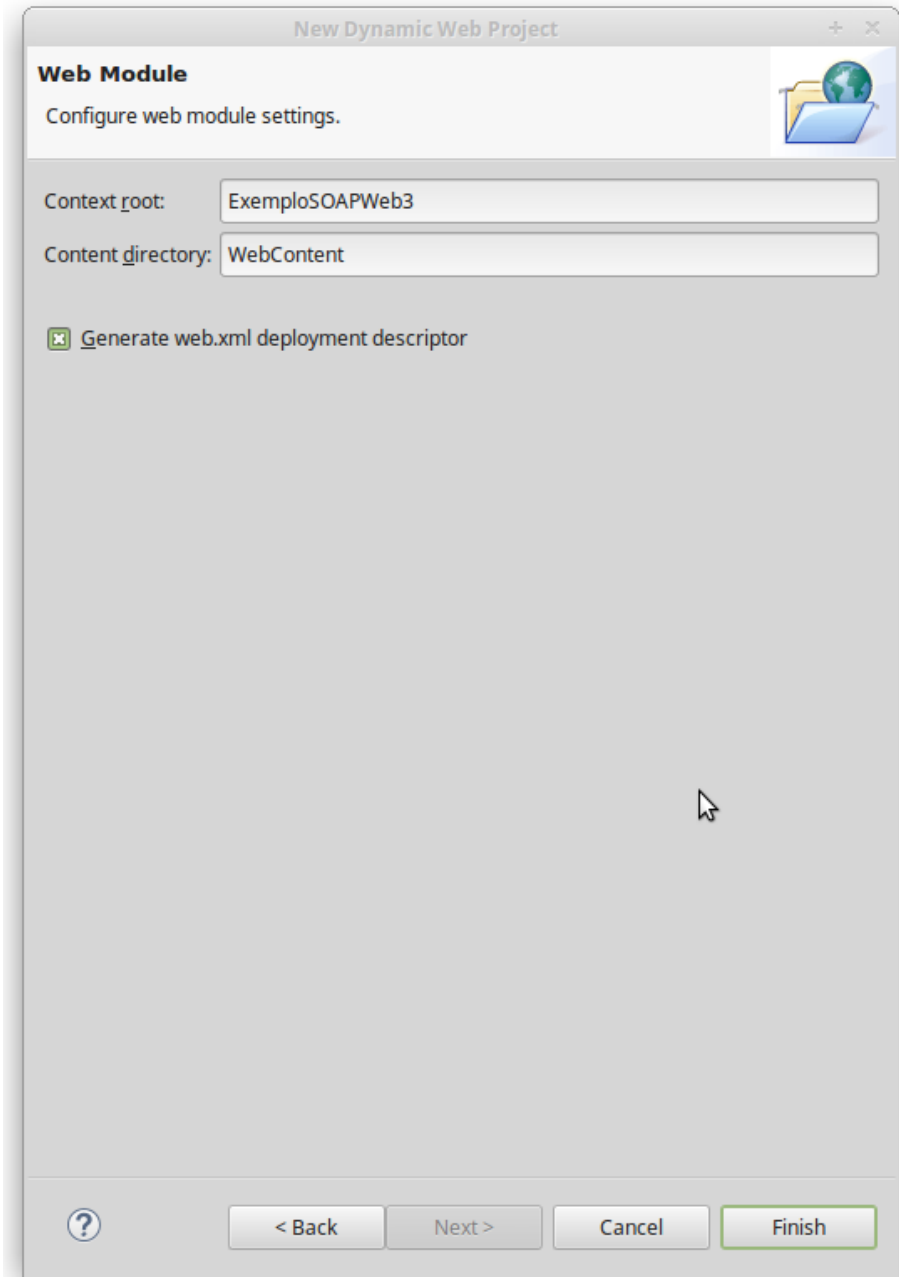
“Next”

- **Se quiser, mudar o folder**
 - Eu deixo o default

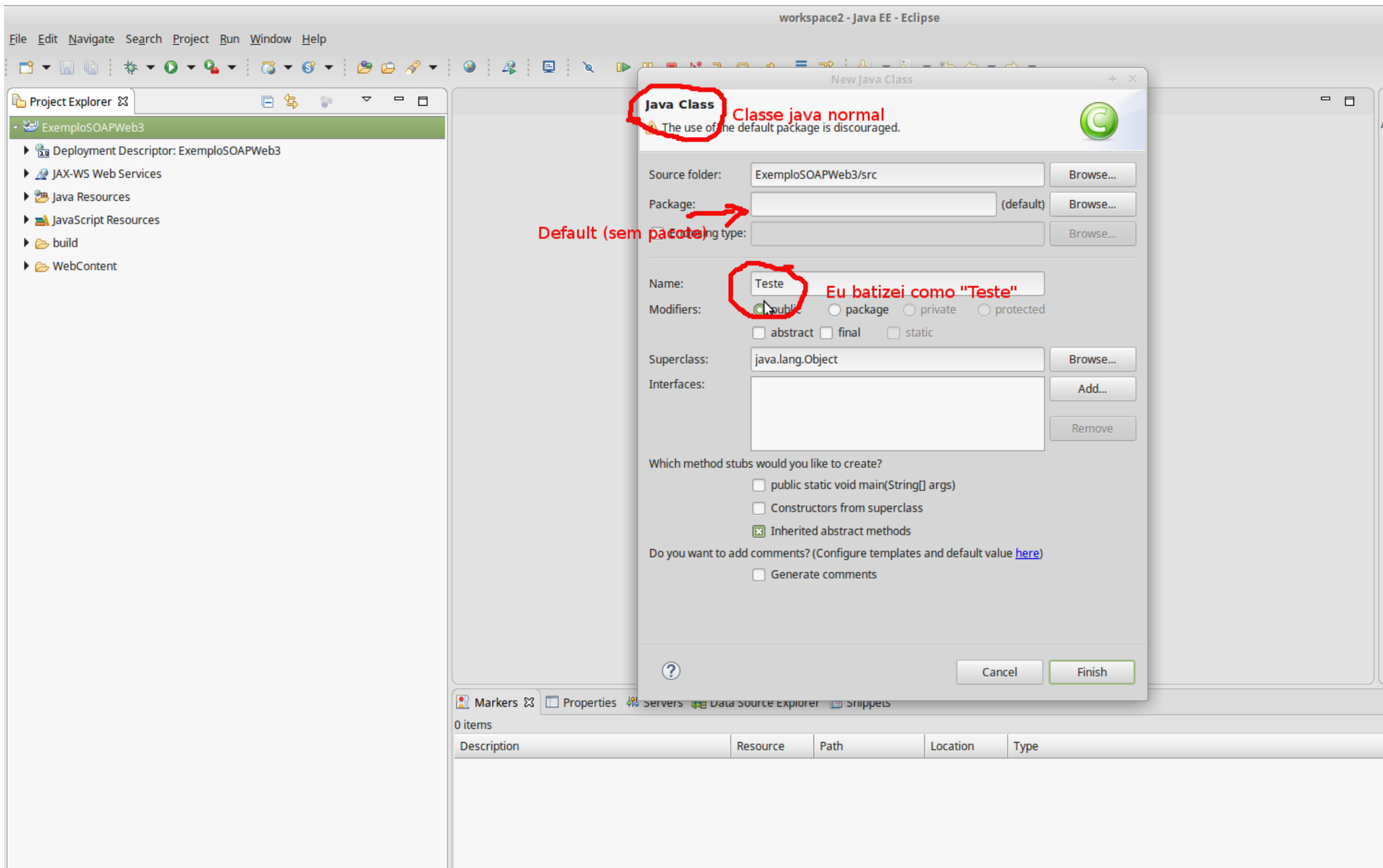


“Finish”

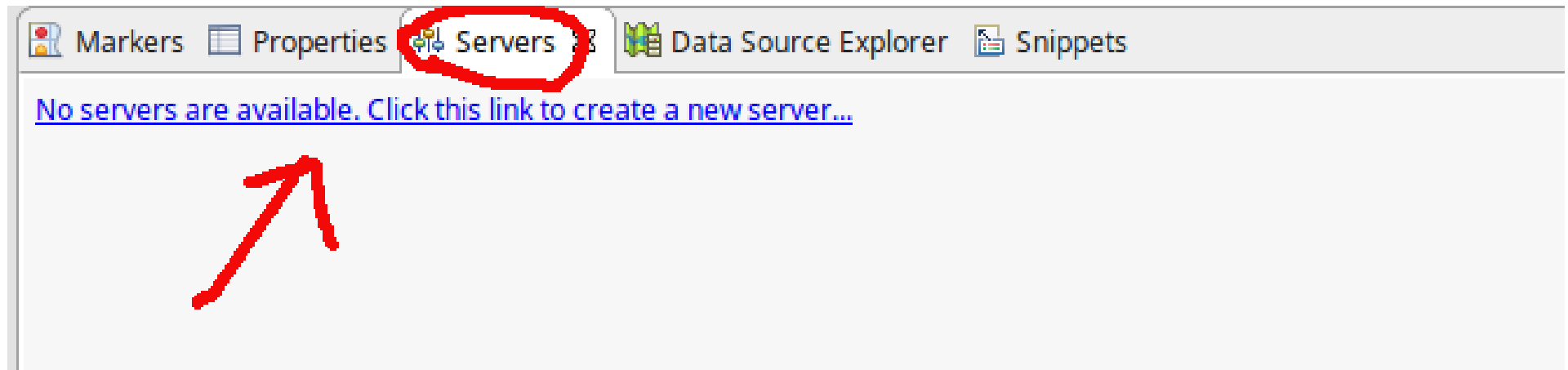
- Não precisa mudar nada



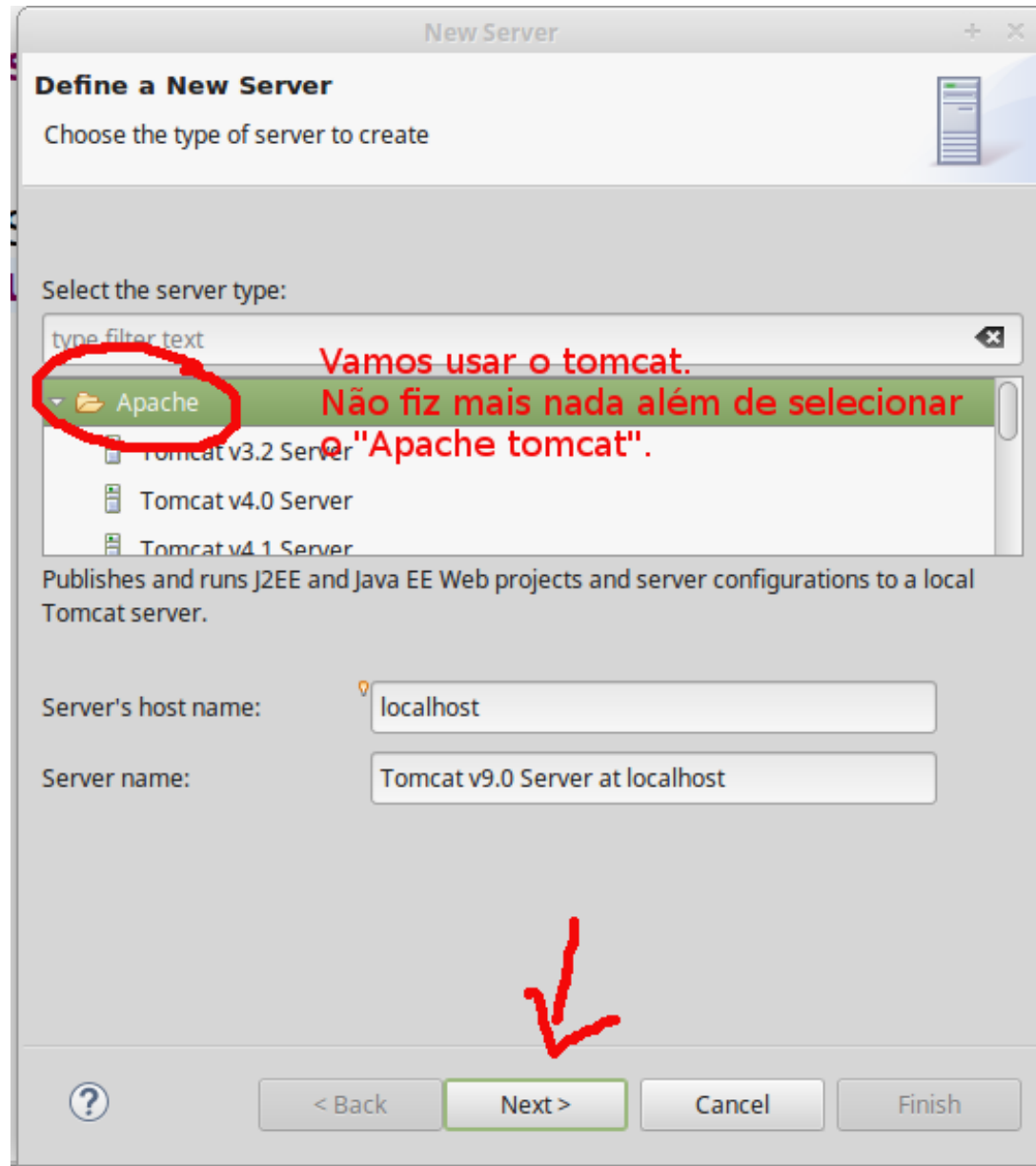
Crie uma classe Java comum



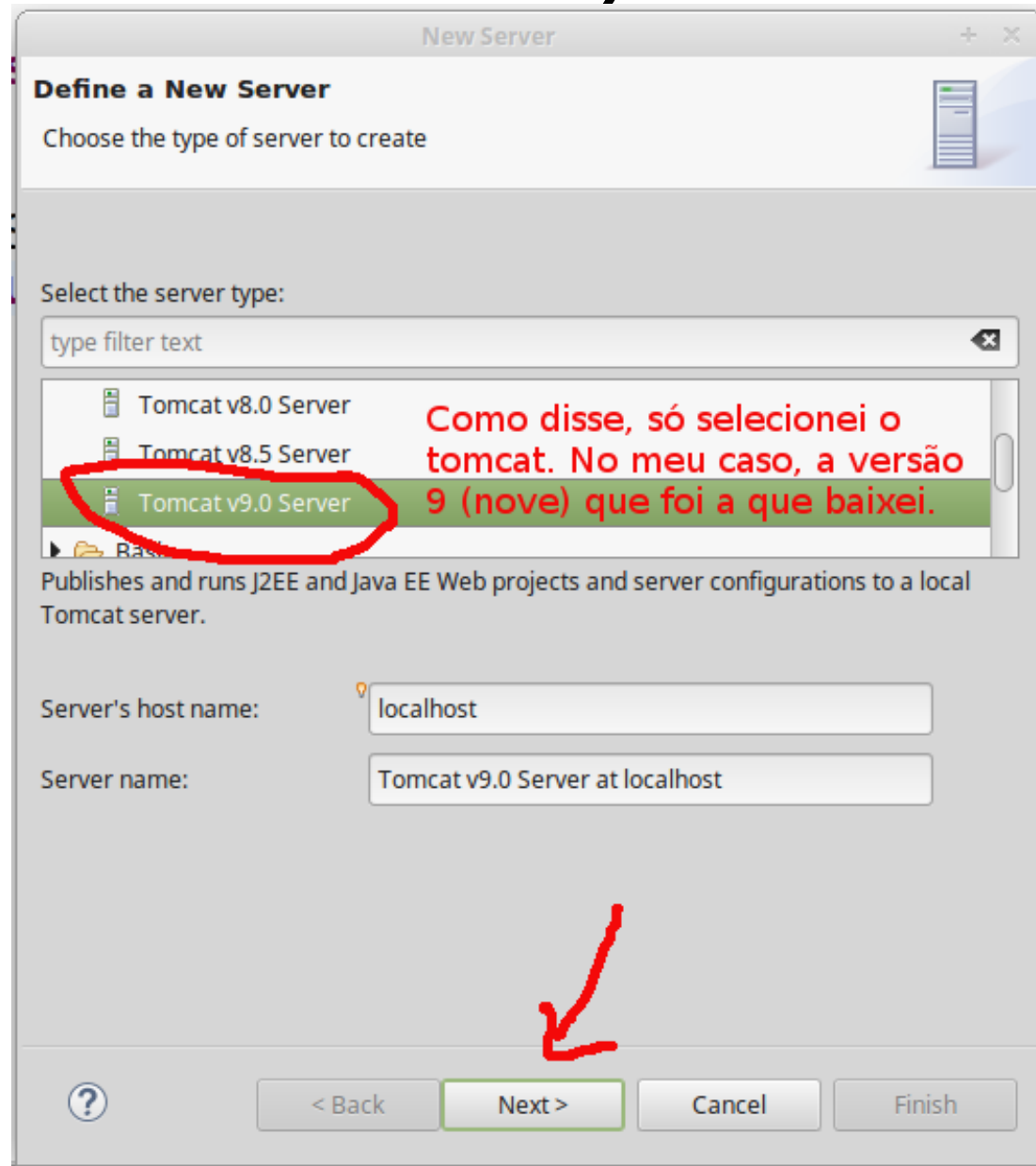
Vamos adicionar o container (se já tiver pule para o slide **16**)



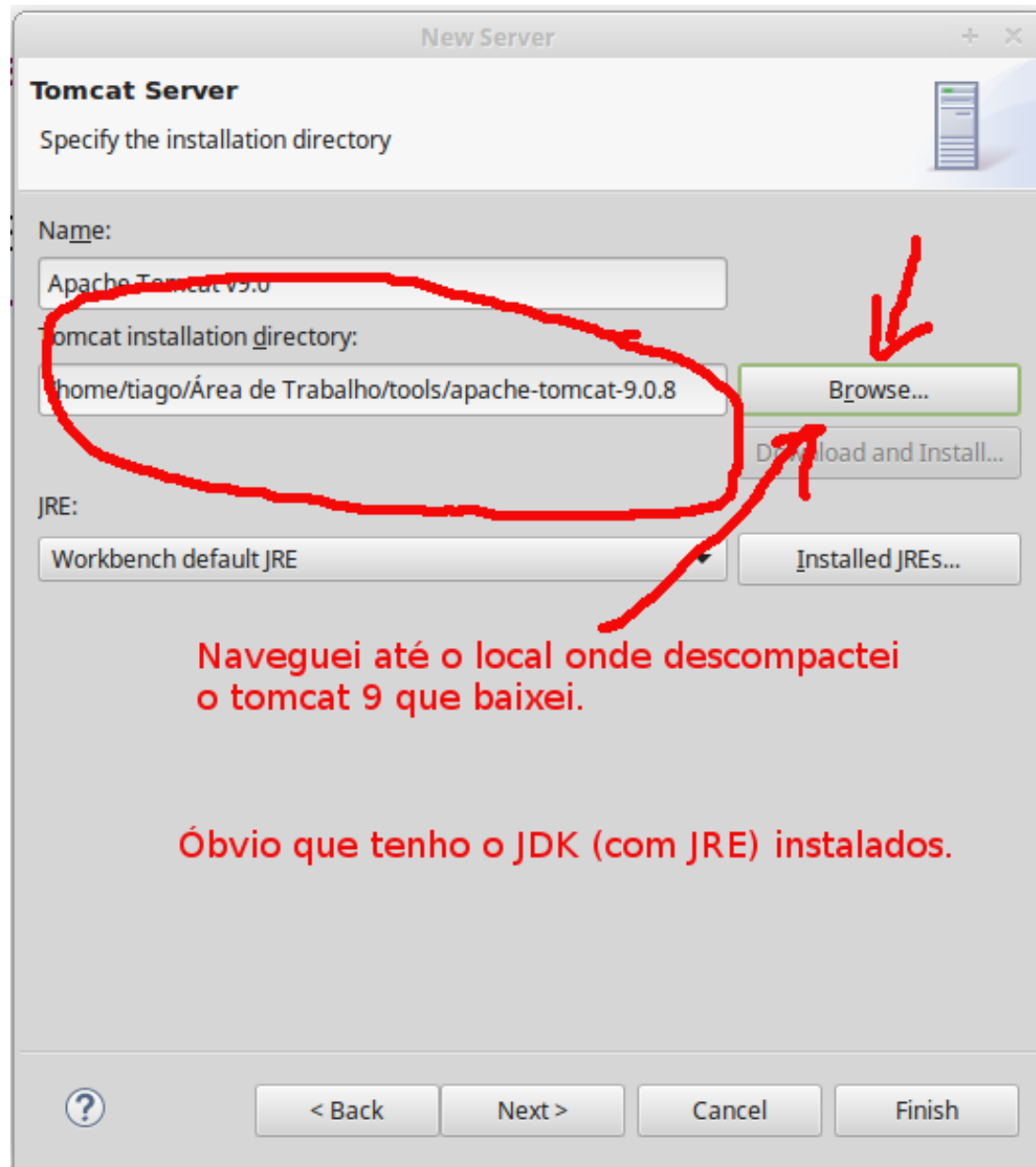
Vamos usar o Apache Tomcat



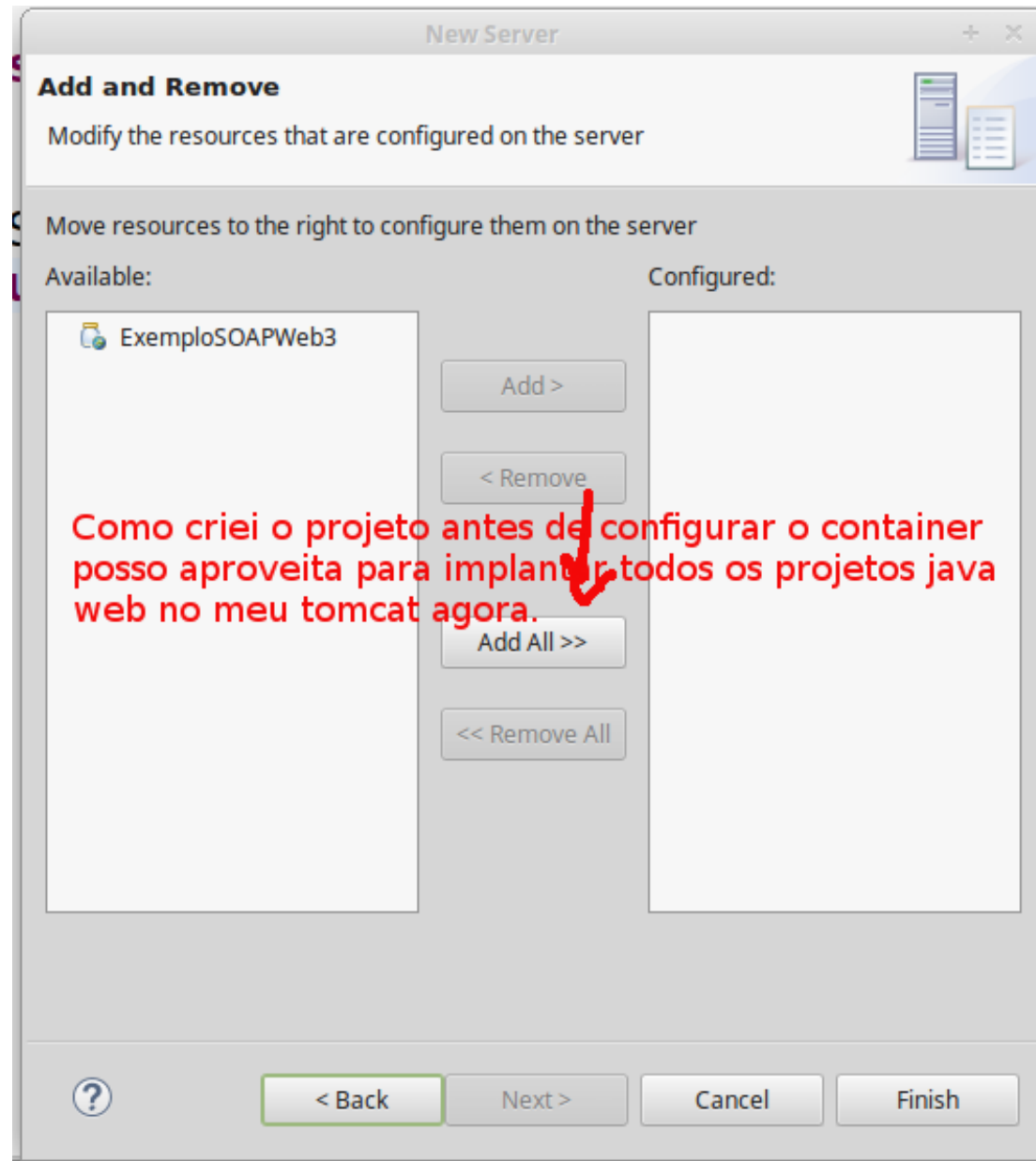
Apache Tomcat (versão 9 foi a que baixei)



Aponte para o diretório onde está seu container web Java



Passe o projeto para o outro lado



Depois de tudo, tem que ficar como a imagem



Código da classe Java “Teste”

workspace2 - Java EE - ExemploSOAPWeb3/src/Teste.java - Eclipse

File Edit Source Refactor Navigate Search Project Run Window Help

Project Explorer

- ExemploSOAPWeb3
 - Deployment Descriptor: ExemploSOAPWeb3
 - JAX-WS Web Services
 - Java Resources
 - src
 - (default package)
 - Teste.java**
 - Libraries
 - JavaScript Resources
 - build
 - WebContent

Teste.java

```
1
2 public class Teste { Classe java comum
3
4
5 public String getPreco(){
6     return "14.40";
7 }
8
9 }
10
```

Método básico que retorna uma String para simular um preço.

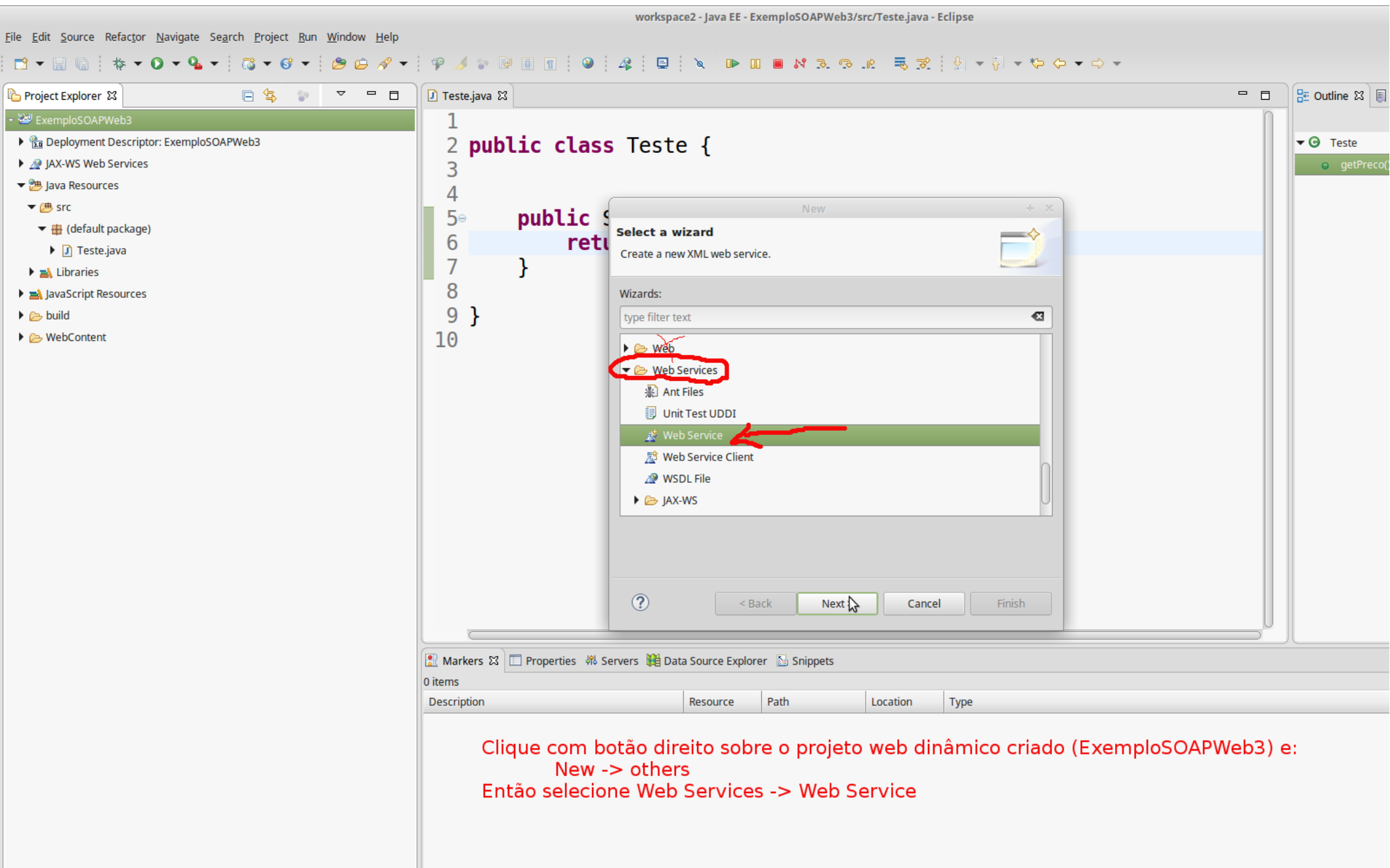
Observem que não tem uma lógica de programação, estamos apenas testando o SOAP com Axis2 no eclipse.

Markers Properties Servers Data Source Explorer Snippets

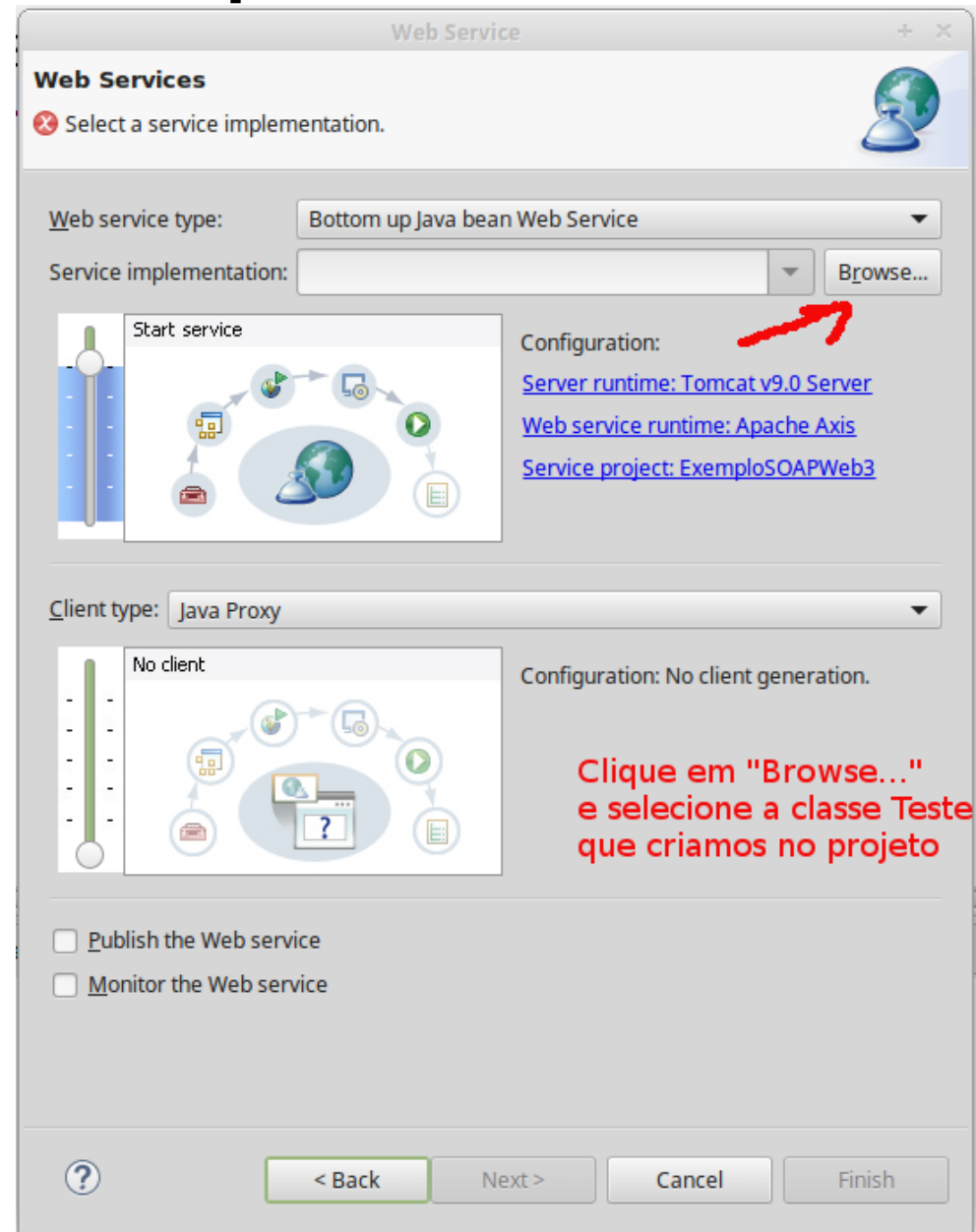
0 items

Description	Resource	Path	Location	Type
-------------	----------	------	----------	------

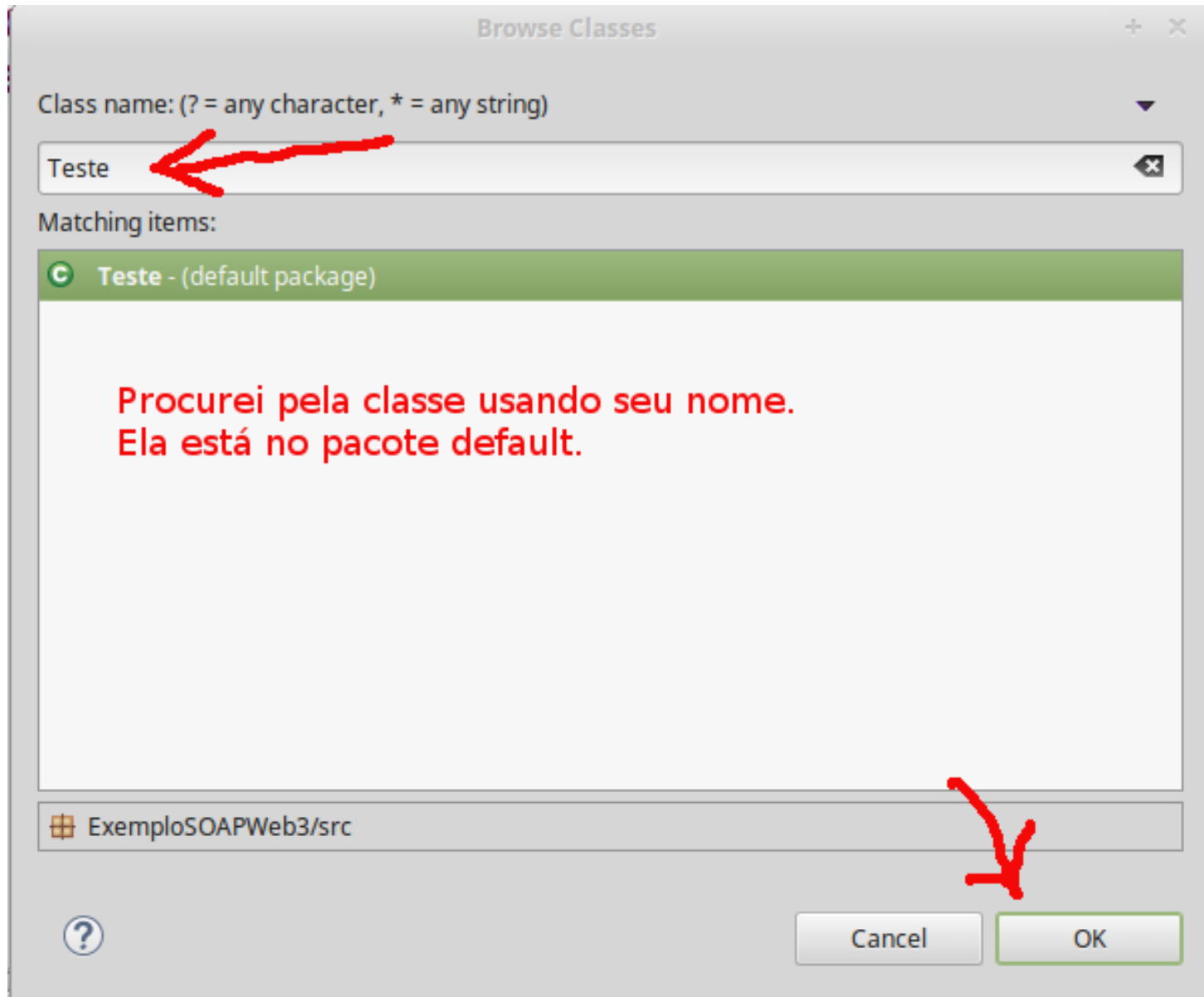
Criando o Web Service



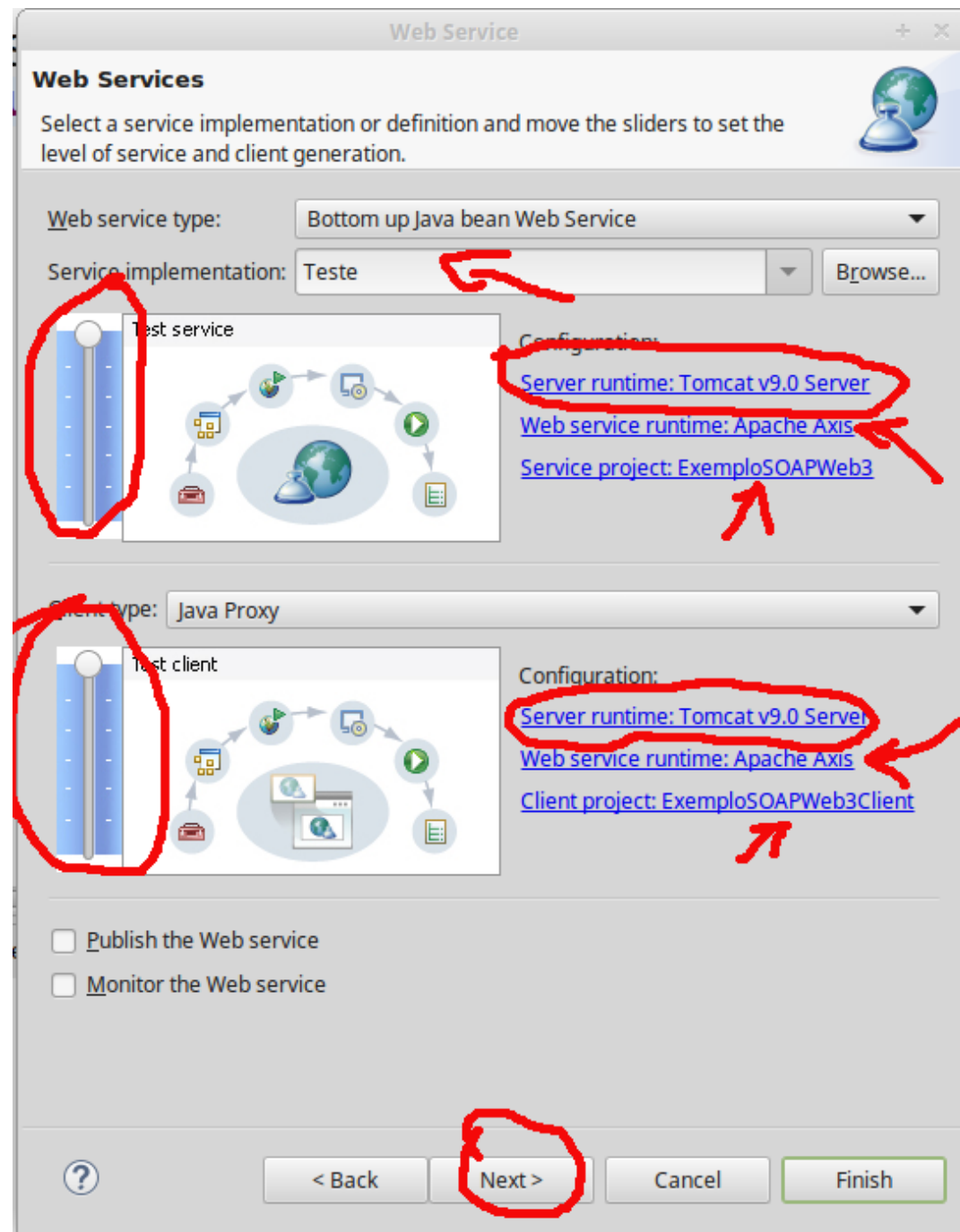
Ache a classe Teste.java do projeto ExemploSOAPWeb3



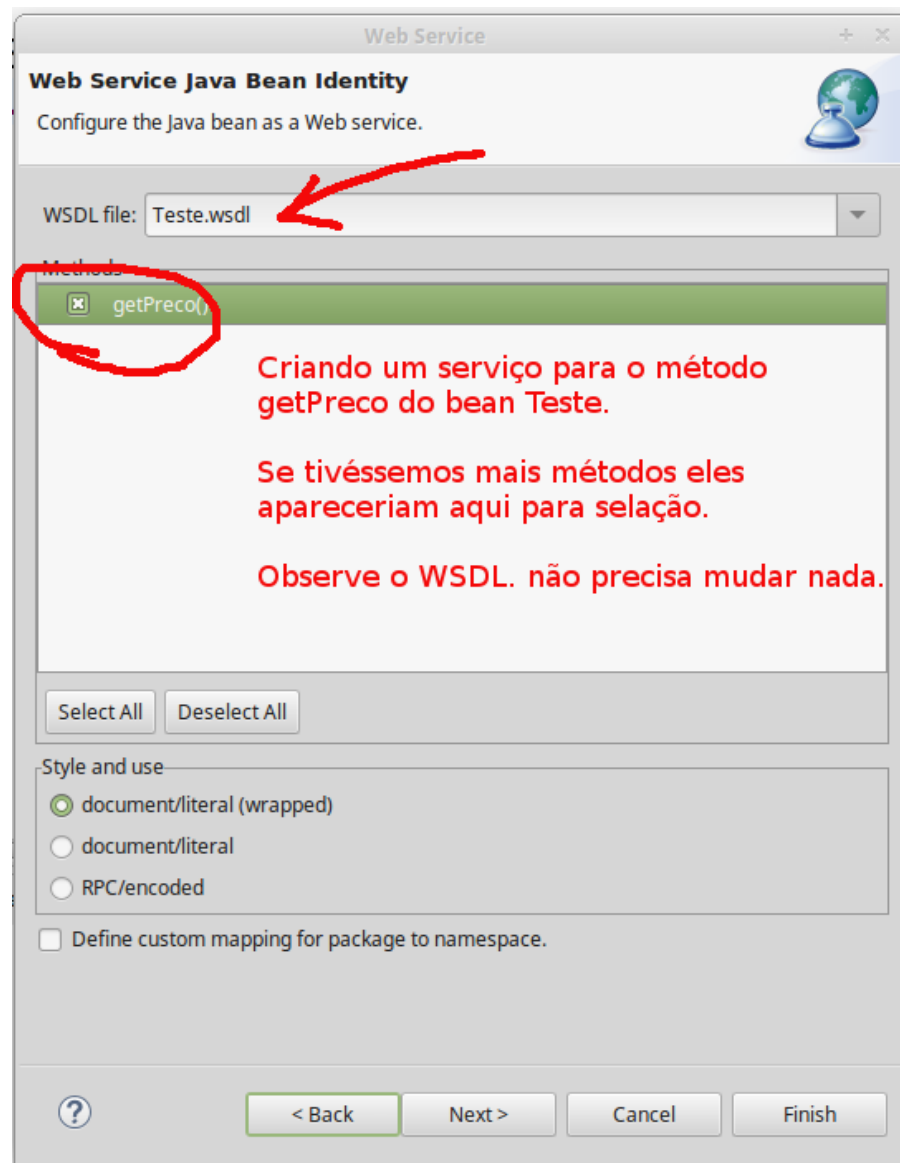
Classe Teste



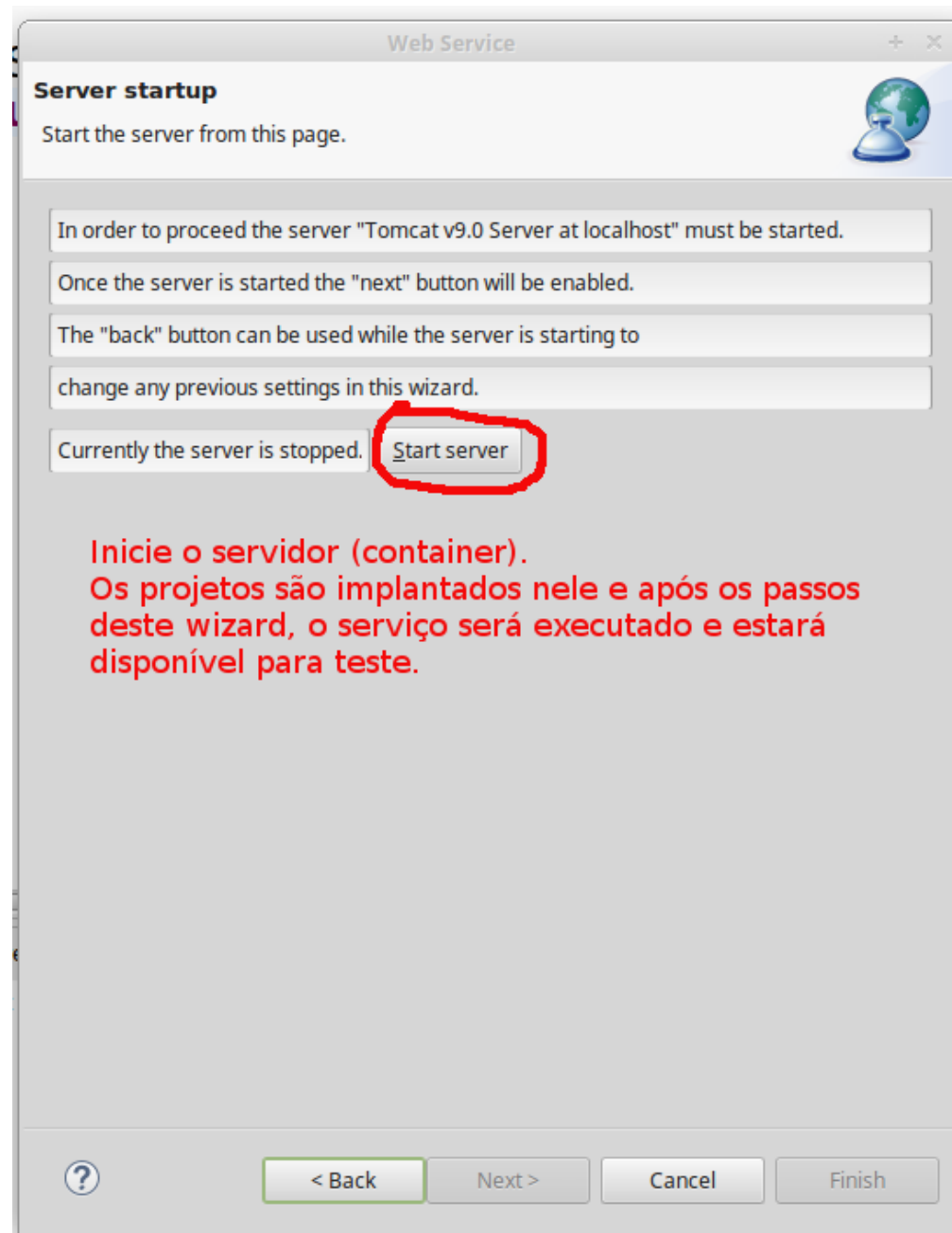
Ajuste para tudo
ficar conforme
demonstrado na
Figura. Observe
tudo com
cuidado.



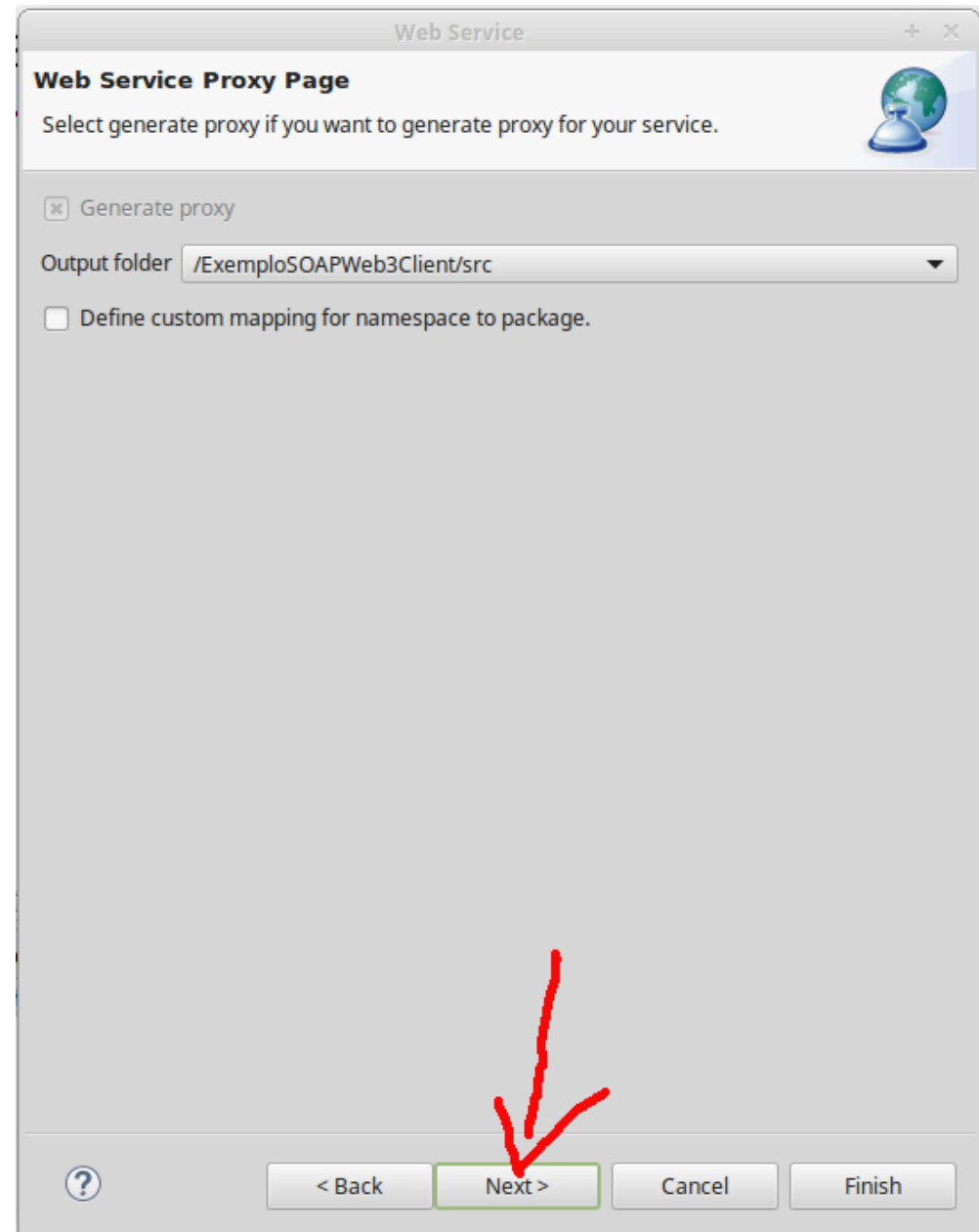
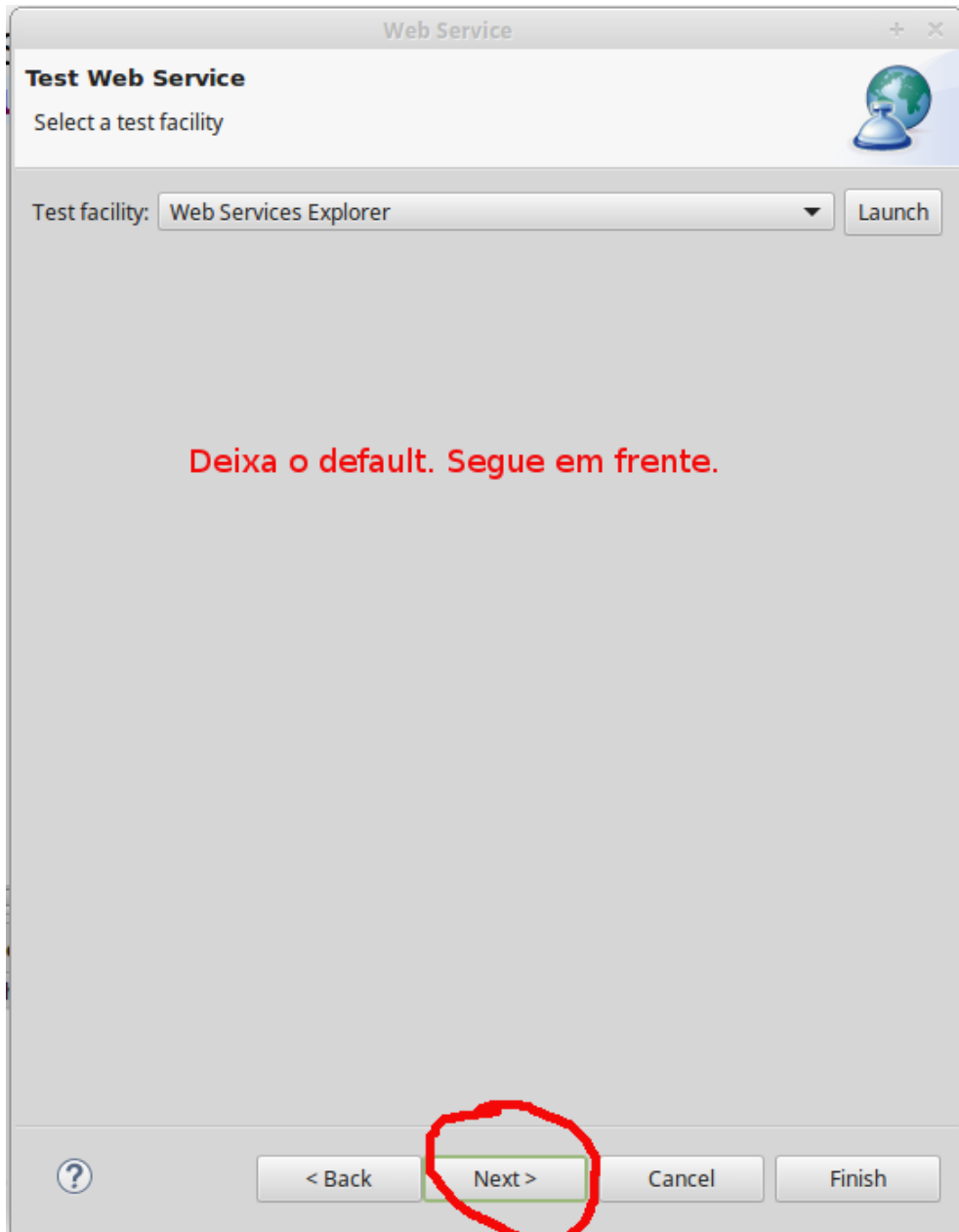
Criando o serviço web automaticamente



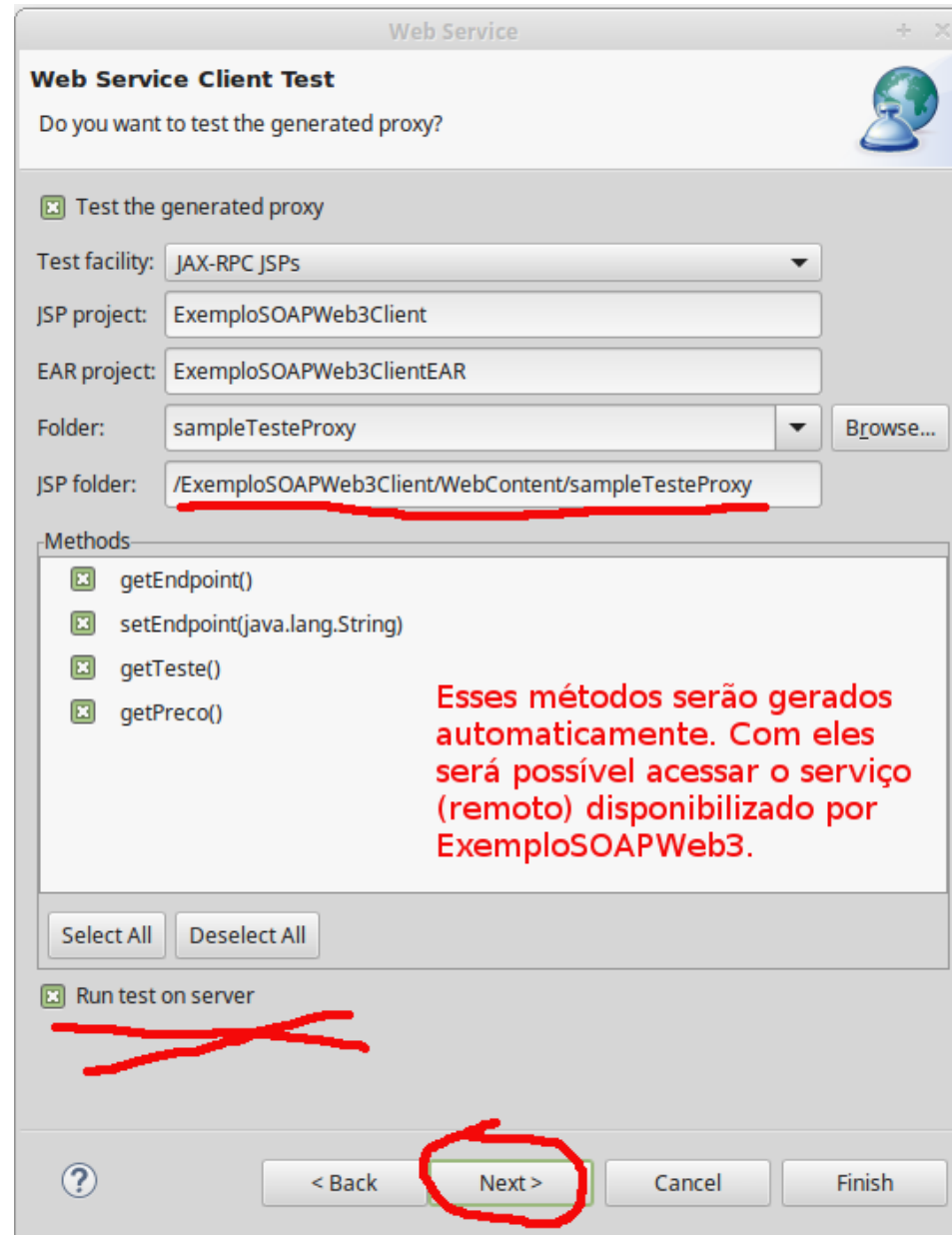
Inicie o servidor



Não mudei anda, basta o “Next”



Métodos criados no cliente. Observe o path do serviço



The image shows a 'Web Service Client Test' dialog box. It has a title bar 'Web Service' and a subtitle 'Web Service Client Test'. Below the subtitle is a question 'Do you want to test the generated proxy?' with a globe icon. There are several input fields: 'Test facility' (JAX-RPC JSPs), 'JSP project' (ExemploSOAPWeb3Client), 'EAR project' (ExemploSOAPWeb3ClientEAR), 'Folder' (sampleTesteProxy), and 'JSP folder' (/ExemploSOAPWeb3Client/WebContent/sampleTesteProxy). The 'JSP folder' field is underlined in red. Below these fields is a 'Methods' section with a list of methods: getEndpoint(), setEndpoint(java.lang.String), getTeste(), and getPreco(). Each method has a checkbox next to it, all of which are checked. To the right of this list is a red text annotation. Below the methods list are 'Select All' and 'Deselect All' buttons. At the bottom, there is a checkbox 'Run test on server' which is checked and crossed out with a red X. At the very bottom are buttons for '?', '< Back', 'Next >', 'Cancel', and 'Finish'. The 'Next >' button is circled in red.

Web Service

Web Service Client Test

Do you want to test the generated proxy?

☒ Test the generated proxy

Test facility: JAX-RPC JSPs

JSP project: ExemploSOAPWeb3Client

EAR project: ExemploSOAPWeb3ClientEAR

Folder: sampleTesteProxy

JSP folder: /ExemploSOAPWeb3Client/WebContent/sampleTesteProxy

Methods

- ☒ getEndpoint()
- ☒ setEndpoint(java.lang.String)
- ☒ getTeste()
- ☒ getPreco()

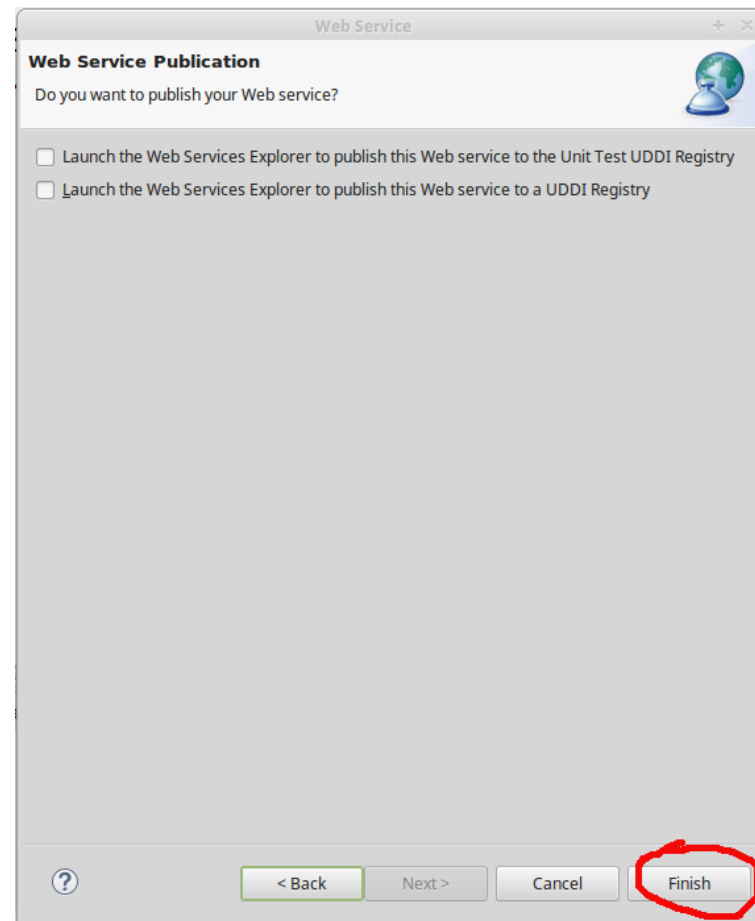
Esse métodos serão gerados automaticamente. Com eles será possível acessar o serviço (remoto) disponibilizado por ExemploSOAPWeb3.

Select All Deselect All

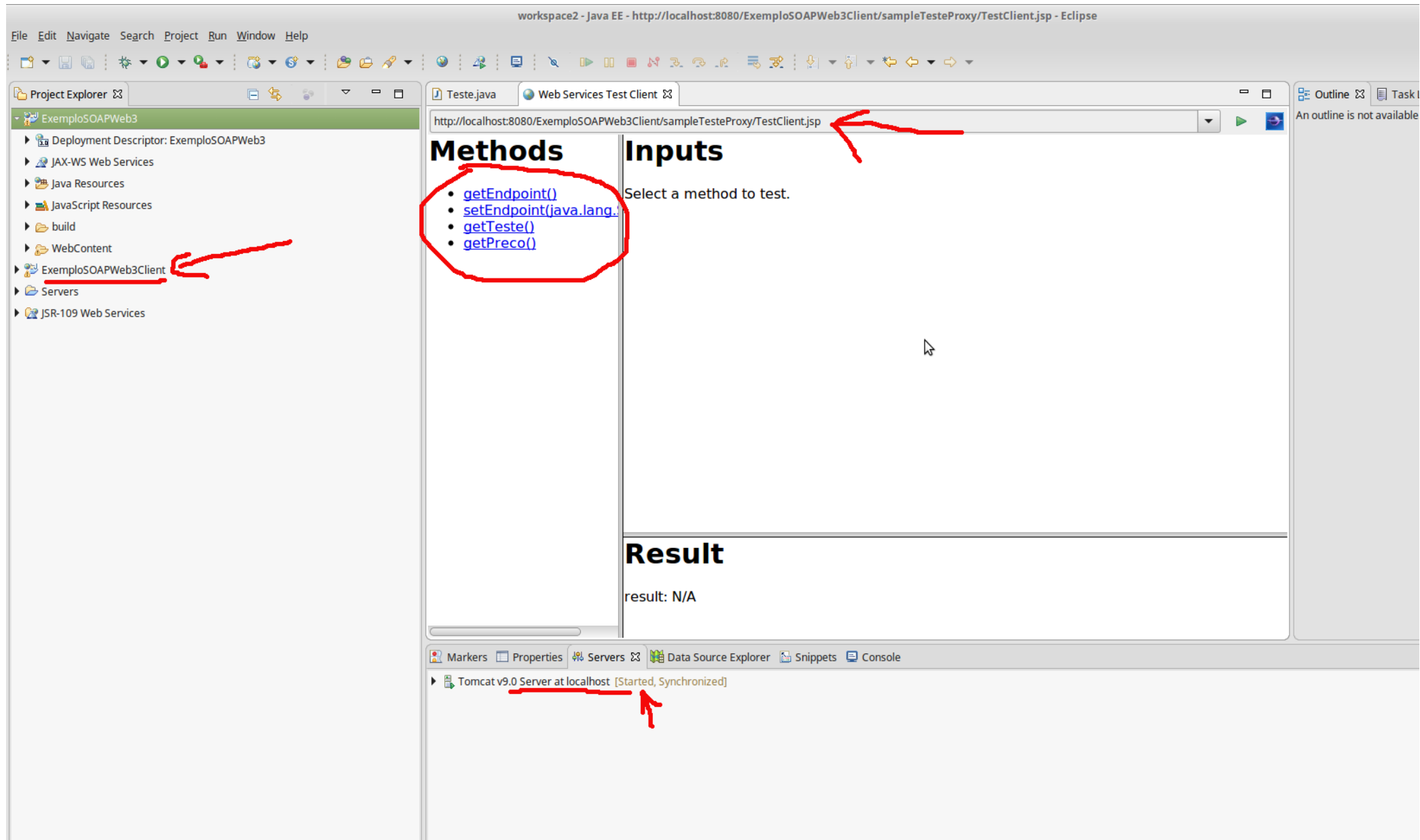
☒ Run test on server

? < Back Next > Cancel Finish

Apenas finalizei



Gerou código do cliente (skeleton), implantou e rodou



Selecione o serviço getPreco do Cliente, clique em “Invoke” e veja o resultado obtido de lá do servidor

workspace2 - Java EE - http://localhost:8080/ExemploSOAPWeb3Client/sampleTesteProxy/TestClient.jsp - Eclipse

File Edit Navigate Search Project Run Window Help

Project Explorer

- ExemploSOAPWeb3
 - Deployment Descriptor: ExemploSOAPWeb3
 - JAX-WS Web Services
 - Java Resources
 - JavaScript Resources
 - build
 - WebContent
- ExemploSOAPWeb3Client ←
- Servers
- JSR-109 Web Services

Teste.java Web Services Test Client

http://localhost:8080/ExemploSOAPWeb3Client/sampleTesteProxy/TestClient.jsp

Methods

- getEndpoint()
- setEndpoint(java.lang.String)
- getTeste()
- getPreco()** 1

Inputs

Invoke Clear 2

Result

14.40 ←

Markers Properties Servers Data Source Explorer Snippets Console

Tomcat v9.0 Server at localhost [Started, Synchronized]

Observe os serviços criados e em qual classe estão o endereço e os métodos que precisam ser alterados

The screenshot displays the Eclipse IDE interface. On the left, the Project Explorer shows the project structure: ExemploSOAPWeb3, ExemploSOAPWeb3Client, Deployment Descriptor: ExemploSOAPWeb3Client, JAX-WS Web Services, Java Resources, src, DefaultNamespace, Teste.java, TesteProxy.java, TesteService.java, and TesteServiceLocator.java (highlighted with a red bracket). The main editor shows the code for TesteServiceLocator.java. The code includes package declarations, class declarations, and methods. A red underline is drawn under the variable 'Teste_address' in the 'getTesteAddress()' method, and a red annotation 'Observe o endereço.' points to it.

```
workspace2 - Java EE - ExemploSOAPWeb3Client/src/DefaultNamespace/TesteServiceLocator.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
ExemploSOAPWeb3
ExemploSOAPWeb3Client
Deployment Descriptor: ExemploSOAPWeb3Client
JAX-WS Web Services
Java Resources
src
DefaultNamespace
Teste.java
TesteProxy.java
TesteService.java
TesteServiceLocator.java
TesteSoapBindingStub.java
Libraries
JavaScript Resources
build
WebContent
Servers
JSR-109 Web Services
TesteServiceLocator.java
2 * TesteServiceLocator.java
7
8 package DefaultNamespace;
9
10 public class TesteServiceLocator extends org.apache.axis.client.Service implements De
11
12 public TesteServiceLocator() {
13 }
14
15
16 public TesteServiceLocator(org.apache.axis.EngineConfiguration config) {
17     super(config);
18 }
19
20 public TesteServiceLocator(java.lang.String wsdlLoc, javax.xml.namespace.QName sN
21     super(wsdlLoc, sName);
22 }
23
24 // Use to get a proxy class for Teste
25 private java.lang.String Teste_address = "http://localhost:8080/ExemploSOAPWeb3/s
26
27 public java.lang.String getTesteAddress() {
28     return Teste_address;
29 }
30
31 // The WSDO service name defaults to the port name.
32 private java.lang.String TesteWSDOServiceName = "Teste";
33
```

Observe o endereço.