



**UNIFESSPA**

UNIVERSIDADE FEDERAL DO SUL E SUDESTE DO PARÁ

# Universidade Federal do Sul e Sudeste do Pará

---

## Sistemas Distribuídos

*Prof.: Warley Junior*

[wmvj@unifesspa.edu.br](mailto:wmvj@unifesspa.edu.br)

# Agenda


---

- ❑ AULA 11 (PARTE I):
- ❑ Middlewares
  - WebServices
    - JAX-WS

# Verificação da versão do Java 8

---

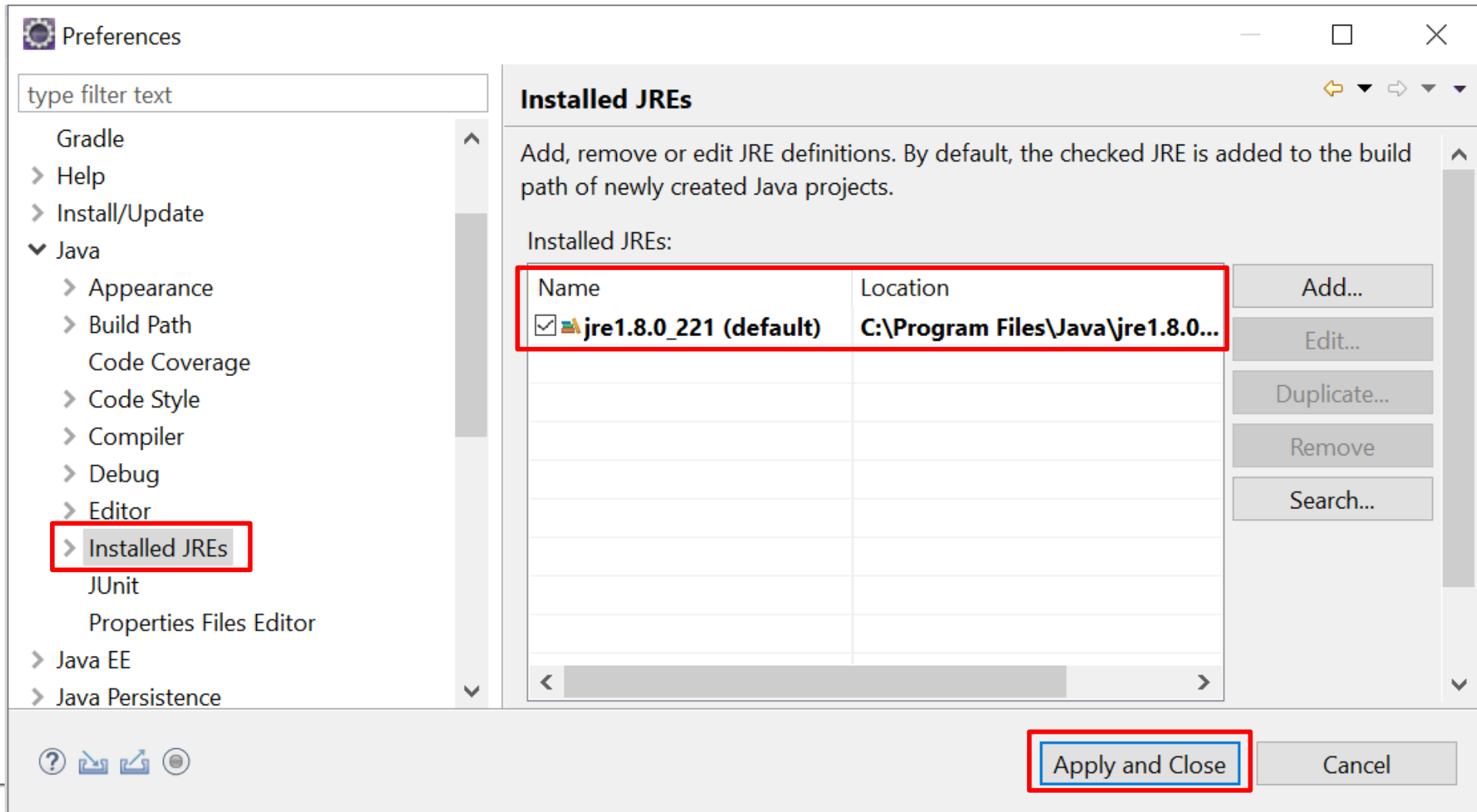
- ❑ Entre na página web do Apache CXF:
  - <http://cxf.apache.org/>
  
- ❑ Verifique a versão do java, pois é necessário ter o JAVA 8:

 C:\Windows\system32\cmd.exe

```
C:\Users\warle>java -version
java version "1.8.0_221"
Java(TM) SE Runtime Environment (build 1.8.0_221-b11)
Java HotSpot(TM) 64-Bit Server VM (build 25.221-b11, mixed mode)

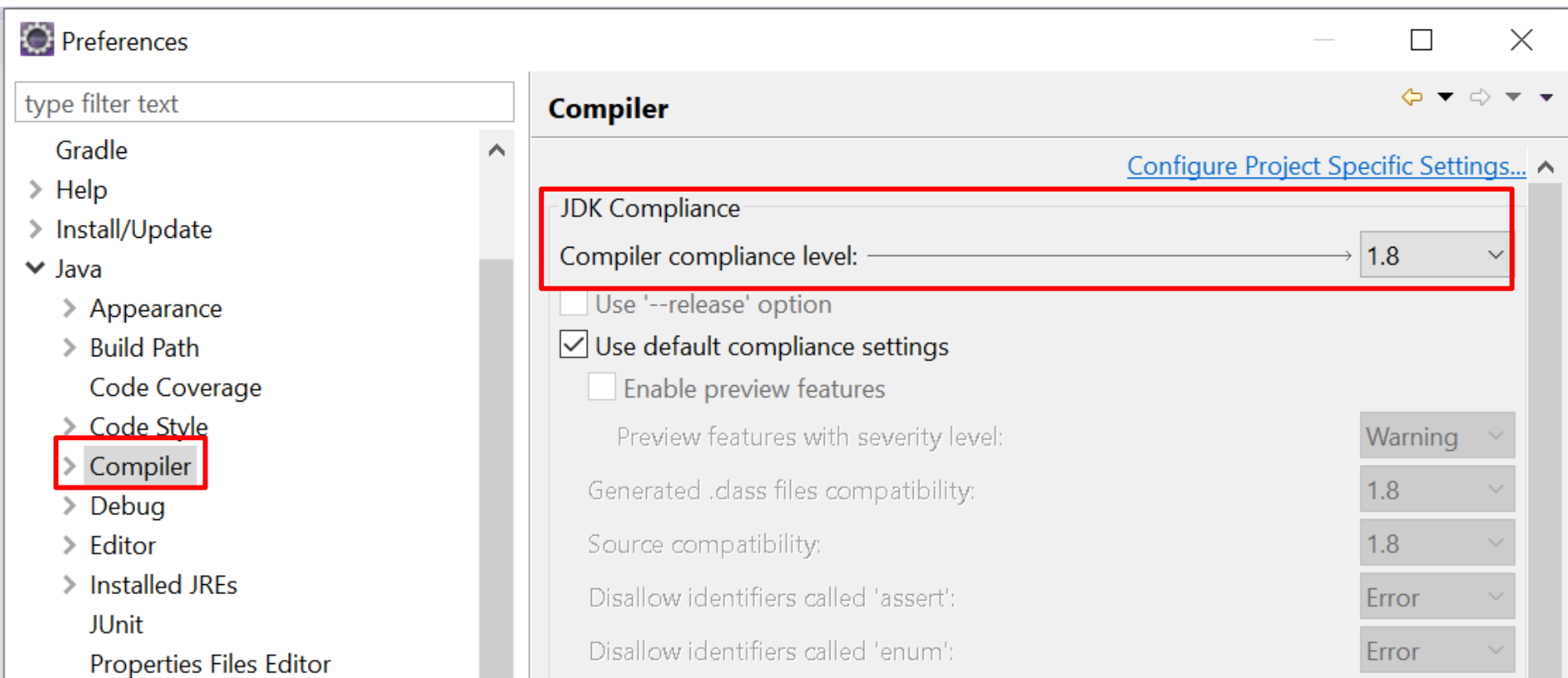
C:\Users\warle>
```

# Criando um Dynamic Web Project no Eclipse



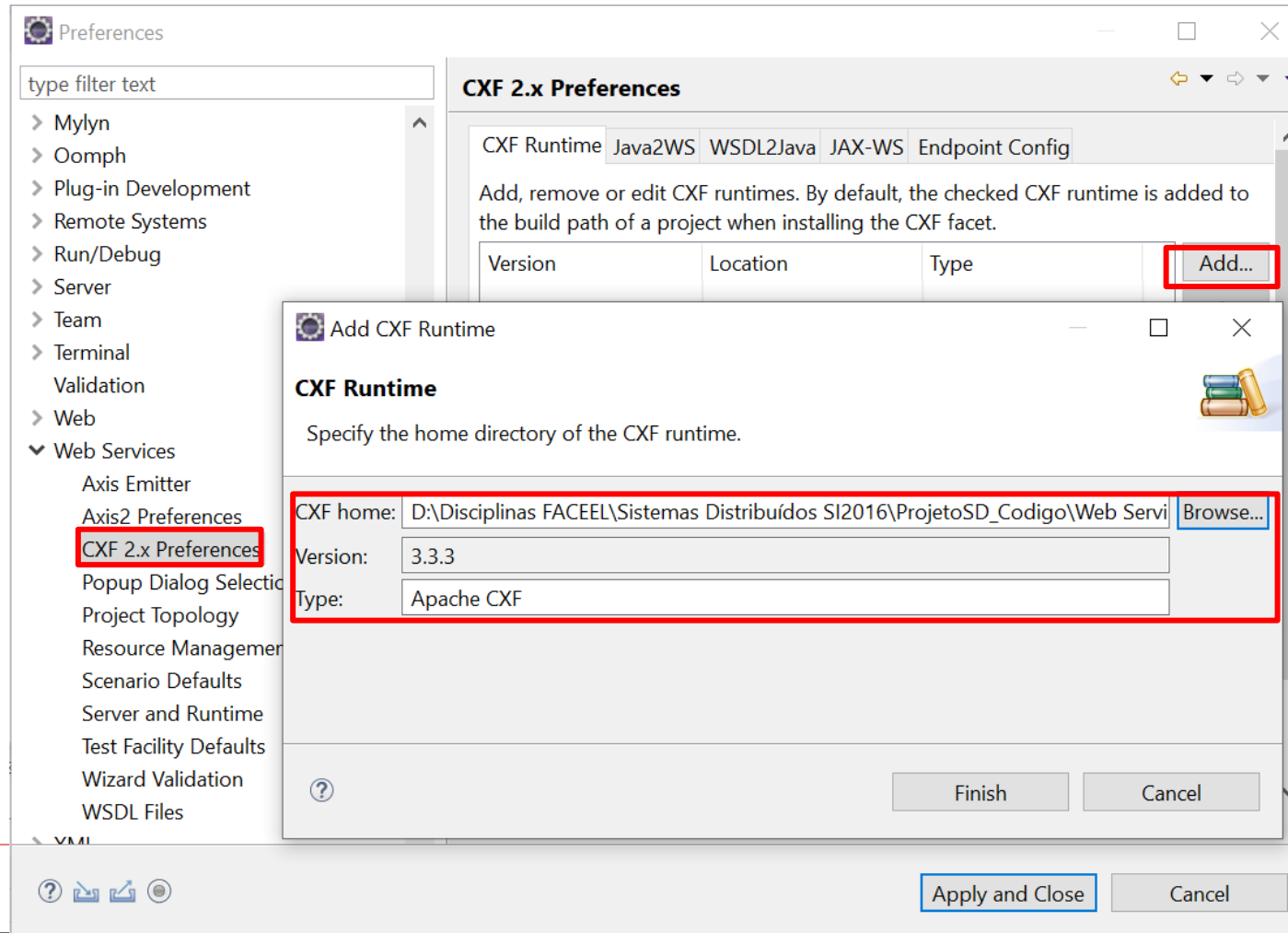
# Criando um Dynamic Web Project no Eclipse

- ❑ Configurar o compilador para usar os recursos do Java 8.



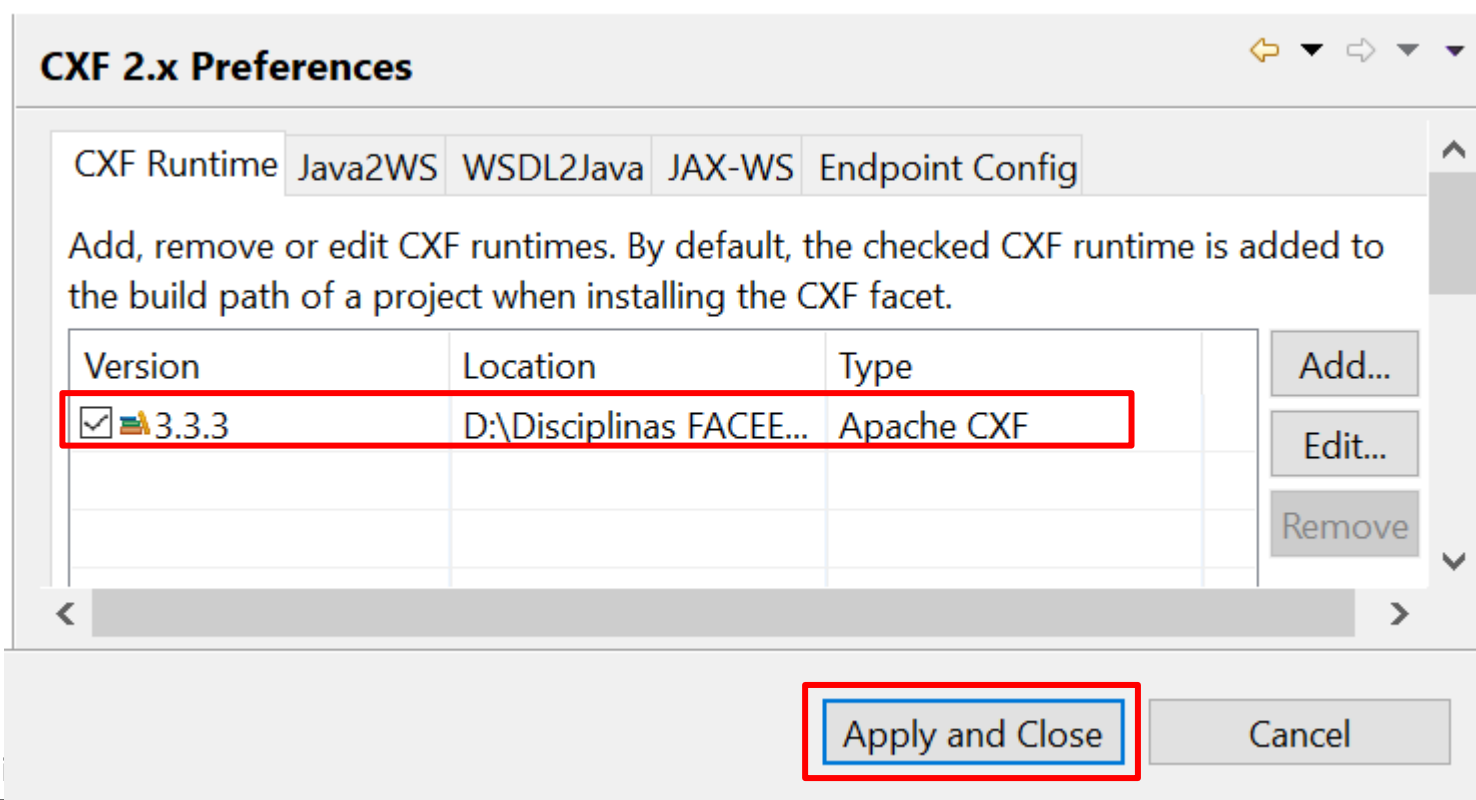
# Criando um Dynamic Web Project no Eclipse

- ❑ Adicionar o Apache CXF para integrar com o Eclipse



# Criando um Dynamic Web Project no Eclipse

- ❑ Selecionar o checkbox do Apache CXF e clicar em "Apply and Close"



# Criar um projeto que use o apache CXF.

The screenshot shows the Eclipse IDE interface with the 'Web Services - Eclipse IDE' window. The 'Project Explorer' on the left indicates no projects are in the workspace. A list of project creation options is shown, with 'Create a Dynamic Web project' highlighted by a red box. To the right, the 'New Dynamic Web Project' dialog is open. The 'Project name' field is set to 'sd-calculadora-ws' and is highlighted with a red box. The 'Project location' section shows 'Use default location' checked and the location path 'D:\Disciplinas FACEEL\Sistemas Distribuídos SI2016\ProjetoSD\_Codigo\'. The 'Target runtime' dropdown is set to '<None>' and is also highlighted with a red box, with a 'New Runtime...' button next to it. Below this, the 'New Server Runtime Environment' dialog is open. It prompts to 'Select the type of runtime environment:' and shows a list of options. 'Apache Tomcat v9.0' is selected and highlighted with a red box. Below the list, it states 'Apache Tomcat v9.0 supports J2EE 1.2, 1.3, 1.4, and Java EE 5, 6, 7, and 8 Web modules.' and includes an unchecked checkbox for 'Create a new local server'. At the bottom, the 'Next >' button is highlighted with a red box, indicating the next step in the wizard.

Web Services - Eclipse IDE

File Edit Navigate Search Project Run Window Help

Project Explorer

There are no projects in your workspace.  
To add a project:

- Create a Maven project
- Create a Java EE EAR project
- Create a Dynamic Web project**
- Create an EJB project
- Create a Connector project
- Create a Java EE application client project
- Create a generic deployable web project.
- Create a JPA project
- Create a project...
- Import projects...

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: sd-calculadora-ws

Project location

☒ Use default location

Location: D:\Disciplinas FACEEL\Sistemas Distribuídos SI2016\ProjetoSD\_Codigo\ Browse...

Target runtime

<None> New Runtime...

New Server Runtime Environment

**New Server Runtime Environment**

Define a new server runtime environment

Select the type of runtime environment:

type filter text

- Apache Tomcat v8.5
- Apache Tomcat v9.0**

Apache Tomcat v9.0 supports J2EE 1.2, 1.3, 1.4, and Java EE 5, 6, 7, and 8 Web modules.

☐ Create a new local server

< Back **Next >** Finish Cancel



# Criando um Dynamic Web Project no Eclipse

**New Server Runtime Environment**

**Tomcat Server**

Specify the installation directory

Name:  
Apache Tomcat v9.0

Tomcat installation directory:  
D:\Disciplinas FACEEL\Sistemas Distribuídos SI2016\ProjetoSD\_Codigo\Web Se

Browse...

Download and Install...

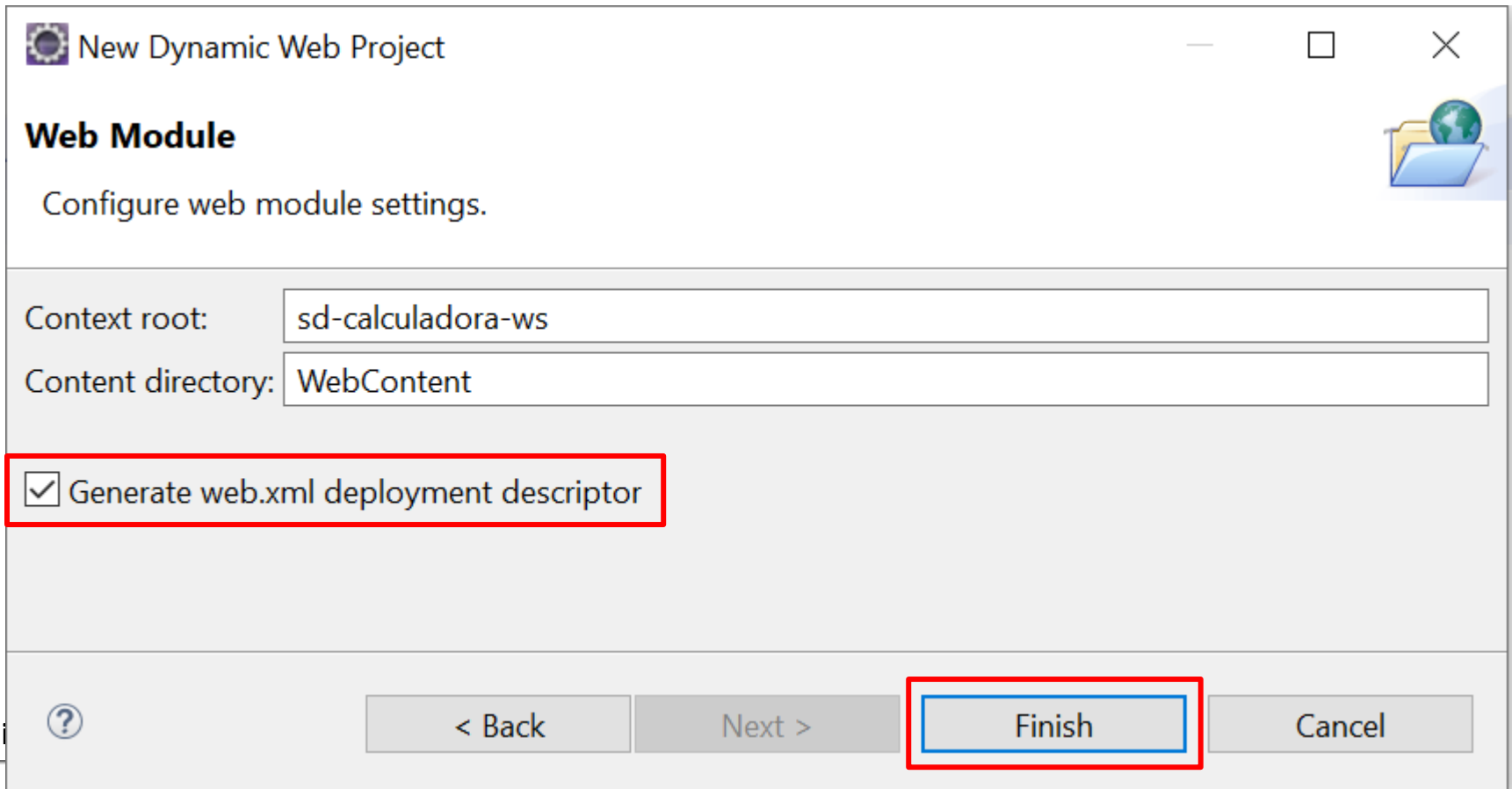
JRE:  
jre1.8.0\_221

Installed JREs...

< Back   Next >   **Finish**   Cancel

# Criando um Dynamic Web Project no Eclipse

- ❑ Clicar no checkbox para gerar o “web.xml”



**New Dynamic Web Project**

**Web Module**  
Configure web module settings.

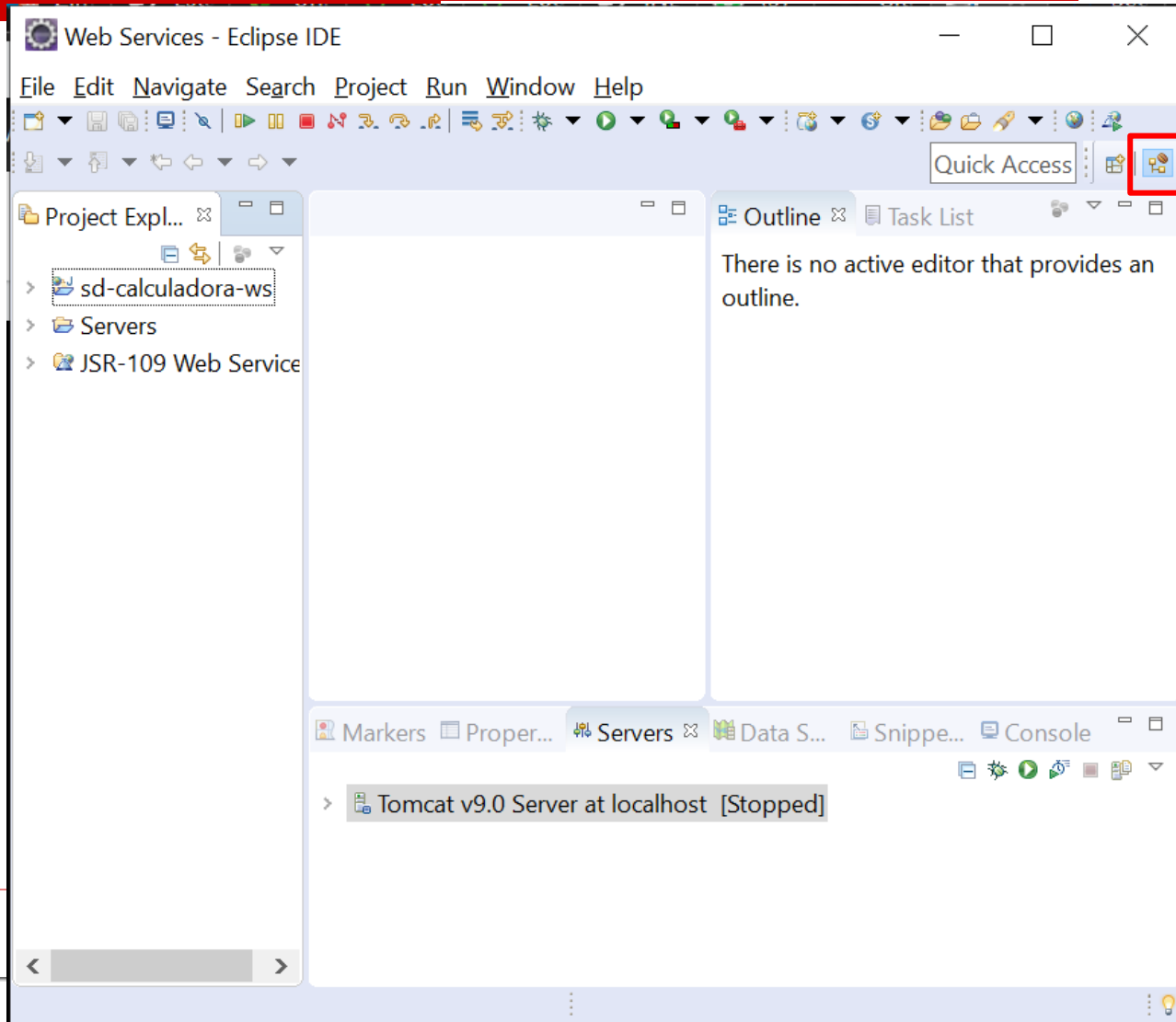
Context root:

Content directory:

☒ Generate web.xml deployment descriptor

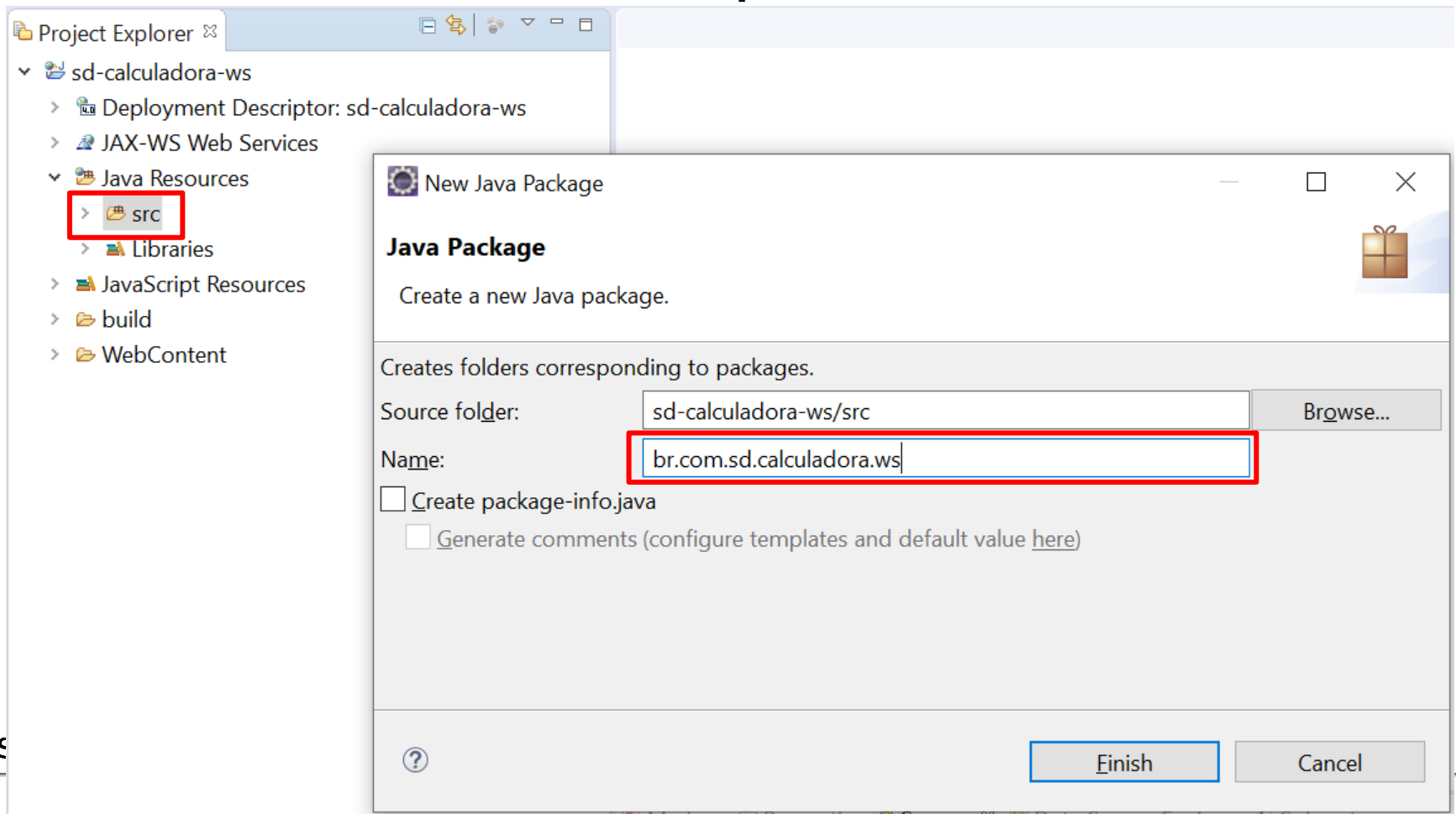
# Criando um Dynamic Web Project no Eclipse

- ❑ Verificar a perspectiva de que está em uso se é o JAVA EE.



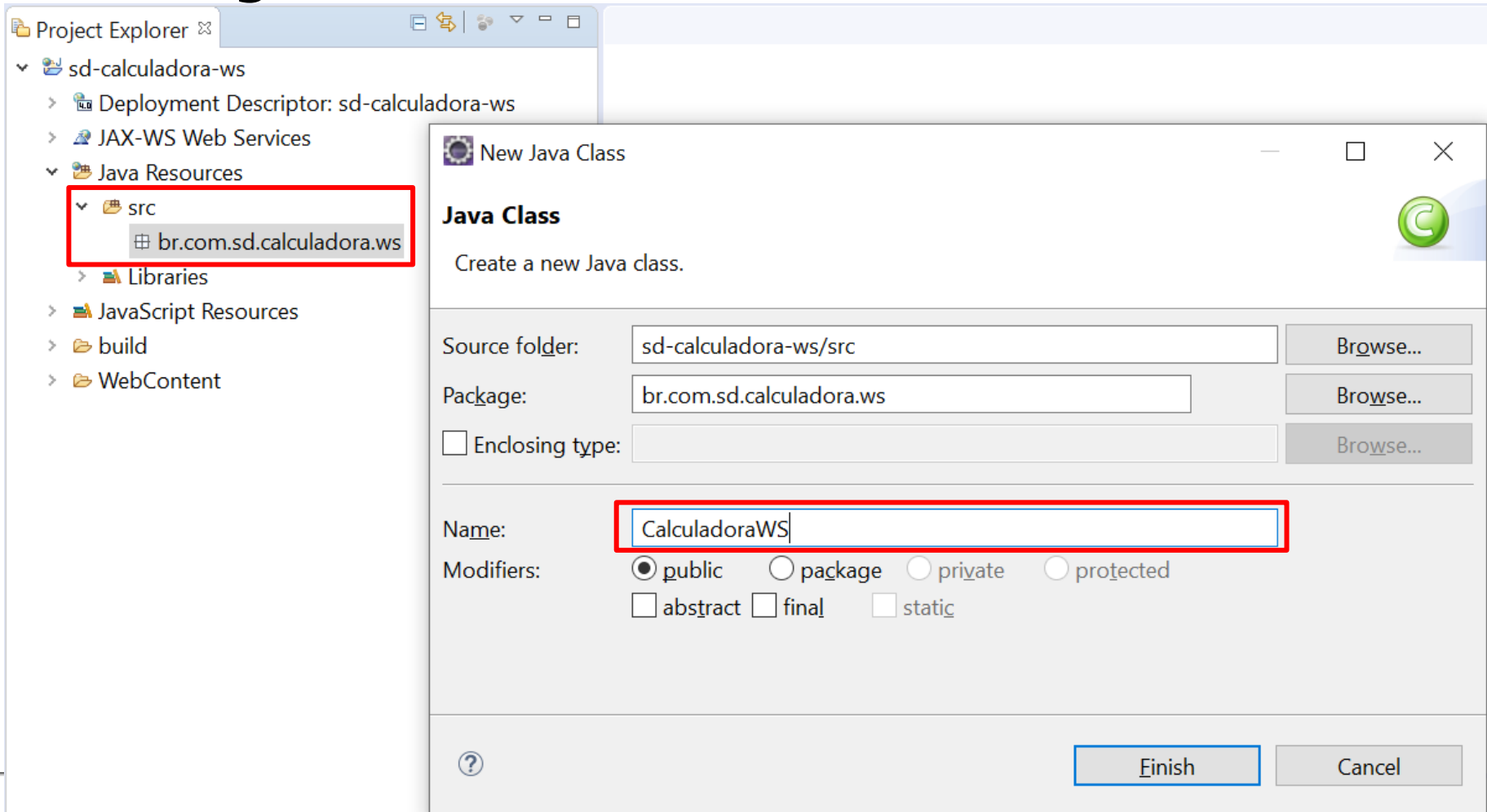
# Implementando as estruturas básicas do Apache CXF

□ Primeiro criar um pacote.



# Implementando as estruturas básicas do Apache CXF

□ Segundo criar uma classe.



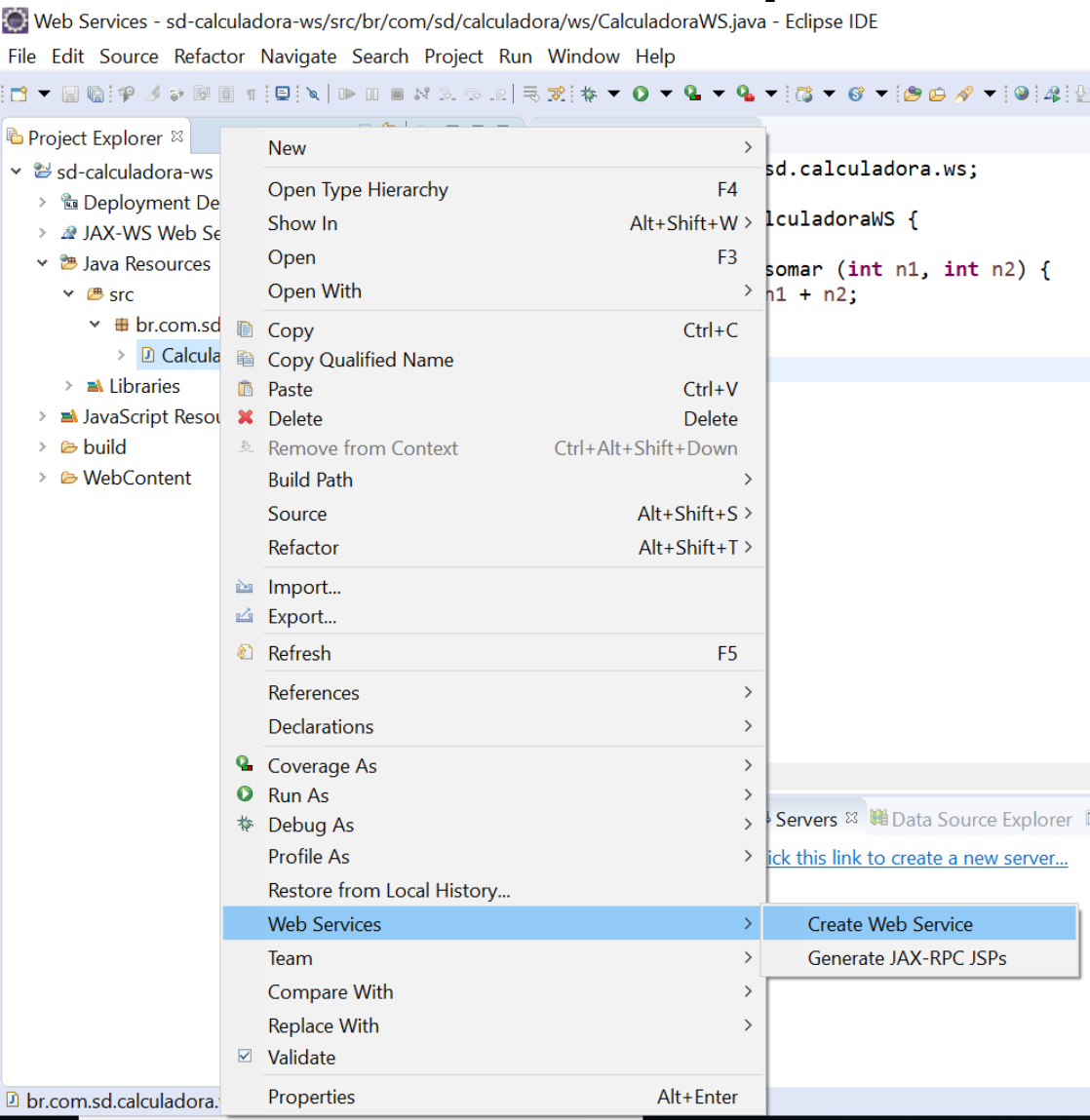
# Implementando as estruturas básicas do Apache CXF

---

- ❑ Criar um método da calculadora chamado “somar”


```
package br.com.sd.calculadora.ws;  
  
public class CalculadoraWS {  
  
    public int somar (int n1, int n2) {  
        return n1 + n2;  
    }  
}
```

# Implementando as estruturas básicas do Apache CXF



□ A partir da classe criada, criar um web service.

# Implementando as estruturas básicas do Apache CXF

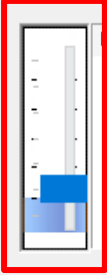
 Web Service

**Web Services**

Select a service implementation or definition and move the sliders to set the level of service and client generation.

Web service type: Bottom up Java bean Web Service

Service implementation: br.com.sd.calculadora.ws.CalculadoraWS [Browse...](#)

 develop service

Configuration:

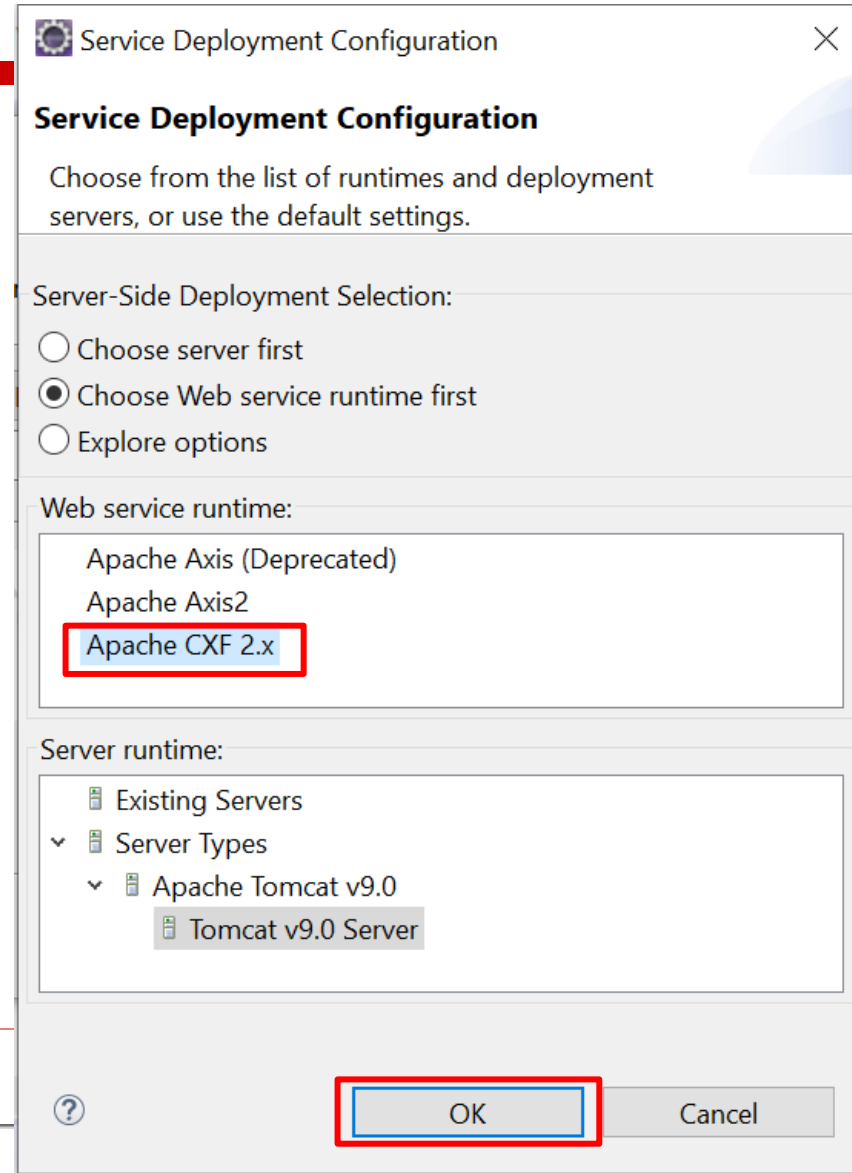
- [Server runtime: Tomcat v9.0 Server](#)
- [Web service runtime: Apache Axis \(Deprecated\)](#)
- [Service project: sd-calculadora-ws](#)

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



# Implementando as estruturas básicas do Apache CXF

- ❑ Selecionar o Apache CXF 2.x como *runtime*.
- ❑ Em seguida pressione OK.



# Implementando as estruturas básicas do Apache CXF

- ❑ Criar uma interface de endpoint.

Web Service

**Apache CXF 3.3.3 Web Service Java Class Starting Point Configuration**

Select an existing Service Endpoint Interface (SEI) or create an SEI by extracting an interface from the implementation class.

☒ Use a Service Endpoint Interface:

☐ Select an SEI:  Browse...

☒ Create an SEI:

Members to declare in the extracted SEI:

☒ • somar(int, int)

Select All...  
Deselect All...

? < Back **Next >** Finish Cancel

# Implementando as estruturas básicas do Apache CXF

Web Service

**Apache CXF 3.3.3 Web Service JAX-WS Annotations Configuration**

Optionally, select the JAX-WS Annotations to add.

Select the Methods to annotate:

	@WebM...	@WebPa...	@RequestWrapper	@ResponseWrapper	@WebResult
▼  CalculadoraWSEndpoint					
• <sup>A</sup> somar(int, int)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Preview:

```
@WebService(name = "CalculadoraWSEndpoint", targetNamespace = "http://ws.calculadora.sd.com.br")
public interface CalculadoraWSEndpoint {

    @WebMethod(operationName = "somar", action = "urn:Somar")
    @WebResult(name = "return")
    int somar(@WebParam(name = "arg0") int n1, @WebParam(name = "arg1") int n2);
}
```

< Back Next > Finish Cancel

# Implementando as estruturas básicas do Apache CXF

- ❑ Selecionar o checkbox "Generate Wrapper and Fault Beans" e "Generate WSDL".

Web Service

**Apache CXF 3.3.3 Web Service Java2WS Configuration**

Configure the Apache CXF Java to WS options.

Java2WS Generation Options:

- ☐ Generate Client
- ☐ Generate Server
- ☒ Generate Wrapper and Fault Beans
- ☒ Generate WSDL

WSDL File: calculadoraws.wsdl

Default SOAP Binding: SOAP 1.1

- ☒ Generate separate XSD for the types

< Back Next > Finish Cancel

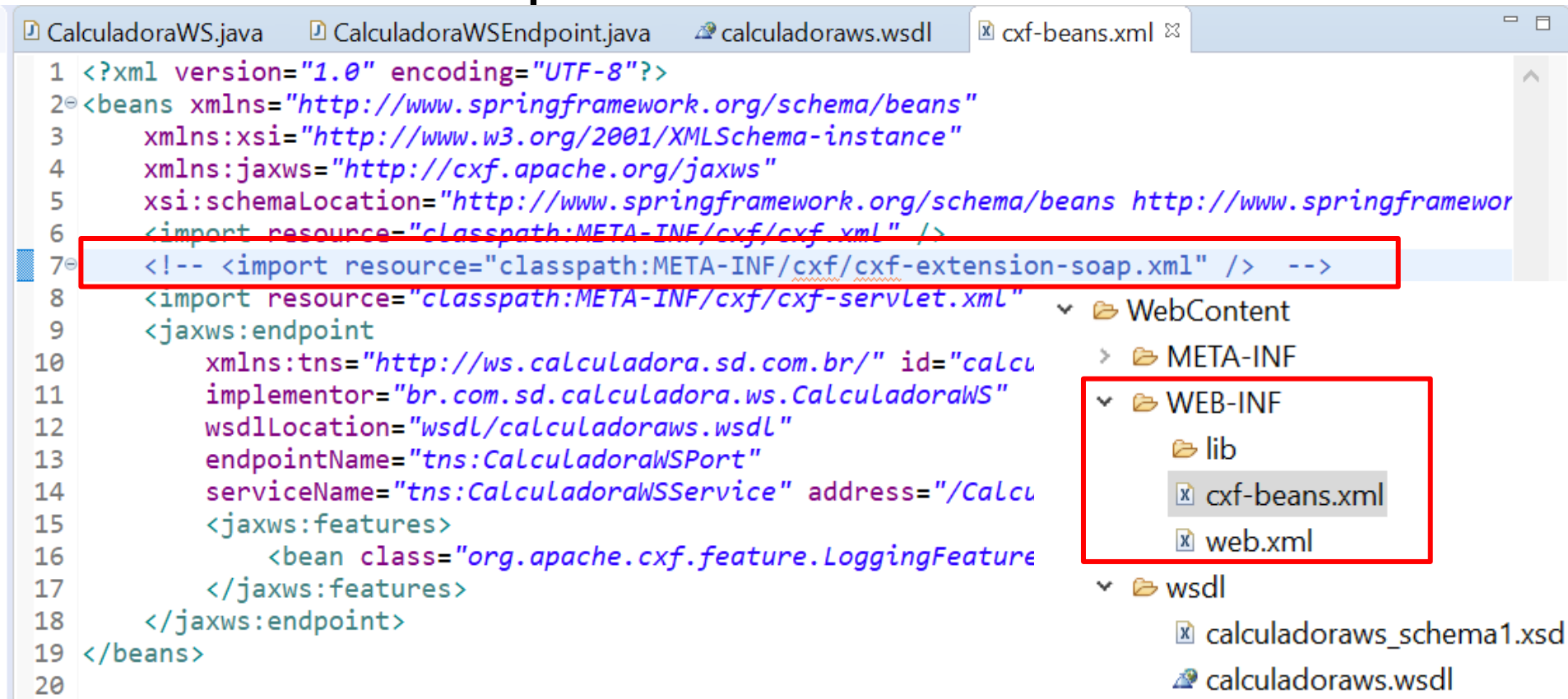
# Implementando as estruturas básicas do Apache CXF

---

- ❑ Na janela “Web Services Publication”, não marcar nada e clicar em “Finish”.

# Implementando as estruturas básicas do Apache CXF

- ❑ Comentar o código da linha "7" dentro arquivo "cxf-beans.xml"



The screenshot shows an IDE with the file `cxf-beans.xml` open. The XML content is as follows:

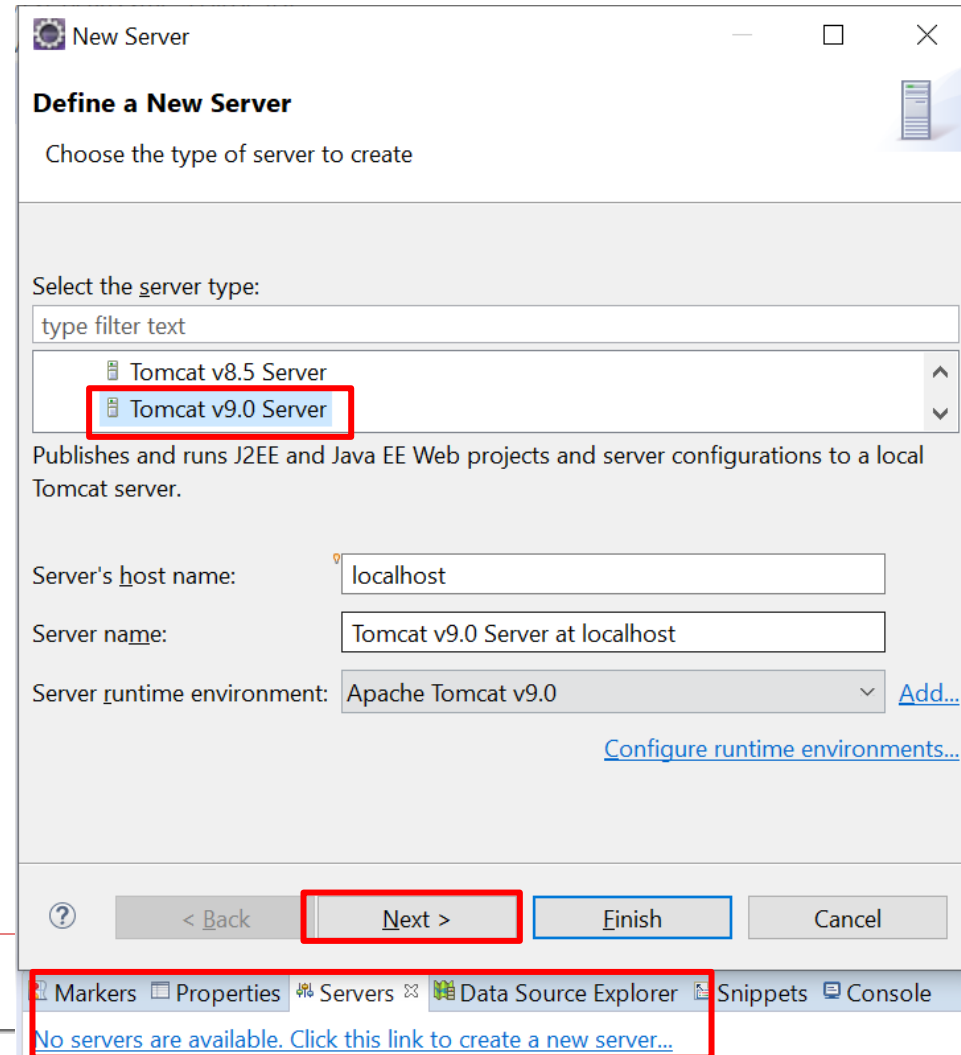
```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     xmlns:jaxws="http://cxf.apache.org/jaxws"
5     xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans
6     <import resource="classpath:META-INF/cxf/cxf.xml" />
7     <!-- <import resource="classpath:META-INF/cxf/cxf-extension-soap.xml" /> -->
8     <import resource="classpath:META-INF/cxf/cxf-servlet.xml" />
9     <jaxws:endpoint
10         xmlns:tns="http://ws.calculadora.sd.com.br/" id="calculadoraWS"
11         implementor="br.com.sd.calculadora.ws.CalculadoraWS"
12         wsdlLocation="wsdl/calculadoraws.wsdl"
13         endpointName="tns:CalculadoraWSPort"
14         serviceName="tns:CalculadoraWSService" address="/CalculadoraWS"
15         <jaxws:features>
16             <bean class="org.apache.cxf.feature.LoggingFeature" />
17         </jaxws:features>
18     </jaxws:endpoint>
19 </beans>
```

Line 7 is highlighted with a red box. The project structure view on the right shows the following hierarchy:

- WebContent
  - META-INF
    - WEB-INF
      - lib
        - `cxf-beans.xml` (highlighted with a red box)
        - `web.xml`
  - wsdl
    - `calculadoraws_schema1.xsd`
    - `calculadoraws.wsdl`

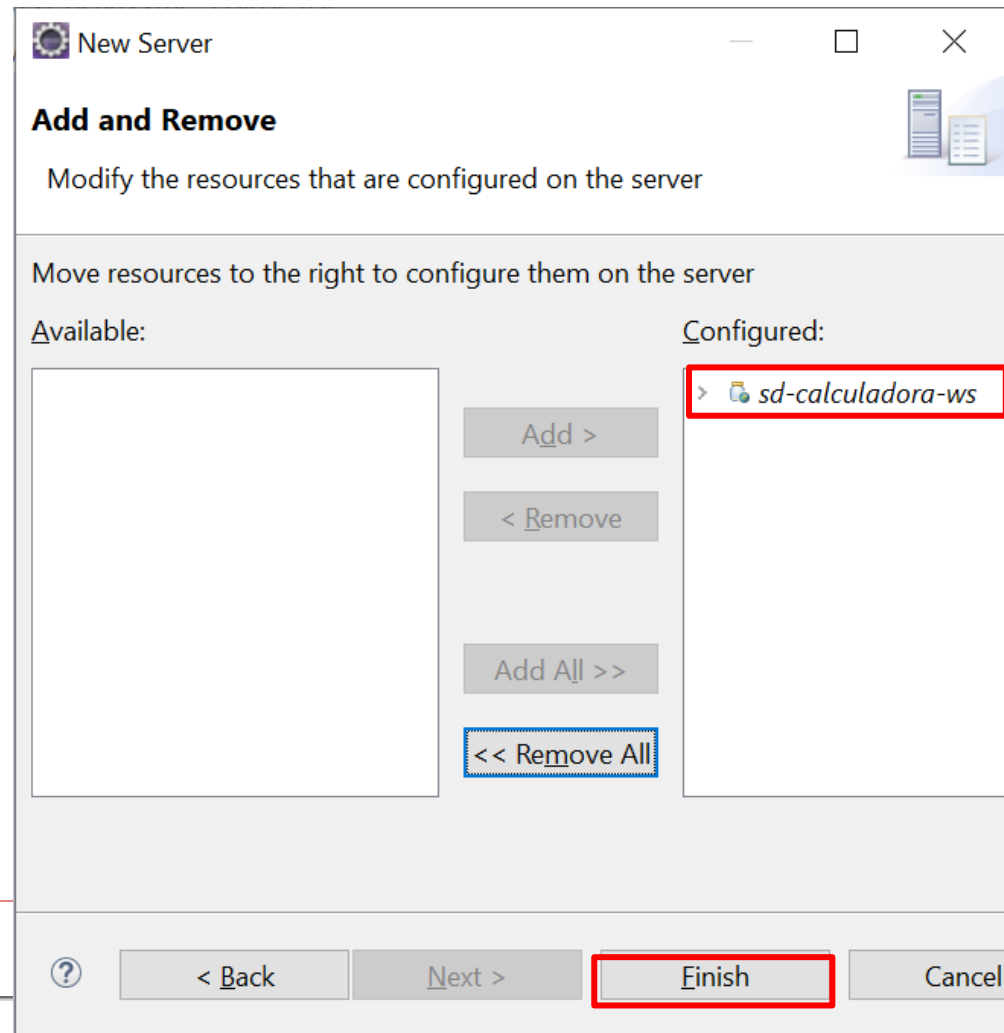
# Implementando as estruturas básicas do Apache CXF

- Ir na aba **SERVES** do Eclipse, e configurar um novo servidor.



# Implementando as estruturas básicas do Apache CXF

- ❑ Jogar a “sd-calculadora-ws” para o meu servidor.





# Implementando as estruturas básicas do Apache CXF

❑ Iniciar o servidor.

The screenshot shows an IDE with the following tabs: `CalculadoraWS.java`, `CalculadoraWSEndpoint.java`, `calculadoraws.wsdl`, and `cxfr-beans.xml`. The `cxfr-beans.xml` file is open, displaying the following XML content:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4       xmlns:jaxws="http://cxf.apache.org/jaxws"
5       xsi:schemaLocation="http://www.springframework.org/schema/beans
6       http://www.springframework.org/schema/beans/spring-beans.xsd"
7       <!-- <import resource="classpath:META-INF/cxf/cxf.xml" -->
8       <import resource="classpath:META-INF/cxf/cxf-spring.xml" -->
9       <jaxws:endpoint
10         xmlns:tns="http://ws.calculadora.com.br"
11         implementor="br.com.sd.calculadora.CalculadoraWS"
12         wsdlLocation="wsdl/calculadora.wsdl"
13         endpointName="tns:CalculadoraWS"
14         serviceName="tns:CalculadoraWS"
15       <jaxws:features>
16         <bean class="org.apache.cxf.feature.FaultInjection" />
17       </jaxws:features>
18     </jaxws:endpoint>
19 </beans>
20
```

A context menu is open over the XML file, with the **Start** option highlighted by a red rectangle. The menu includes the following items:

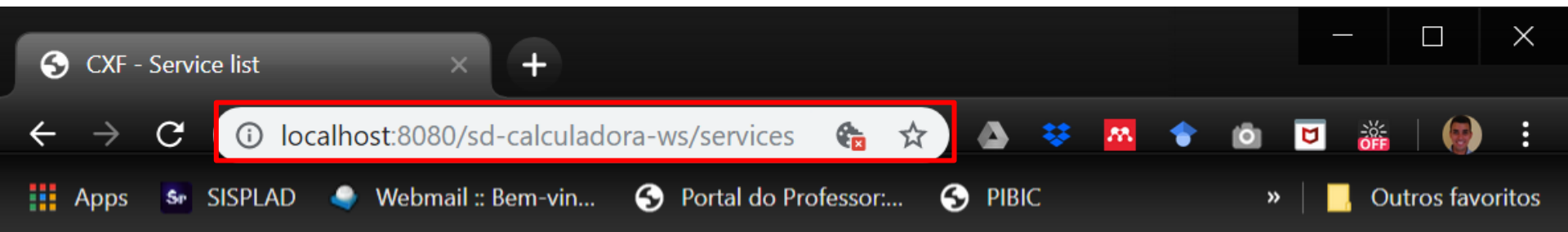
- New
- Open (F3)
- Show In (Alt+Shift+W)
- Copy (Ctrl+C)
- Paste (Ctrl+V)
- Delete (Delete)
- Rename (F2)
- Debug (Ctrl+Alt+D)
- Start (Ctrl+Alt+R)**
- Profile
- Stop (Ctrl+Alt+S)
- Publish (Ctrl+Alt+P)
- Clean...
- Add and Remove...
- Monitoring
- Clean Tomcat Work Directory...
- Update Password...
- Properties (Alt+Enter)

At the bottom of the IDE, the `Tomcat v9.0 Server at localhost` is shown as `[Stopped, Rep...`.

# Implementando as estruturas básicas do Apache CXF

---

- ❑ Webservice rodando.
- ❑ O método somar esta disponível pelo WSDL ao lado.



Available SOAP services:

**CalculadoraWSEndpoint**

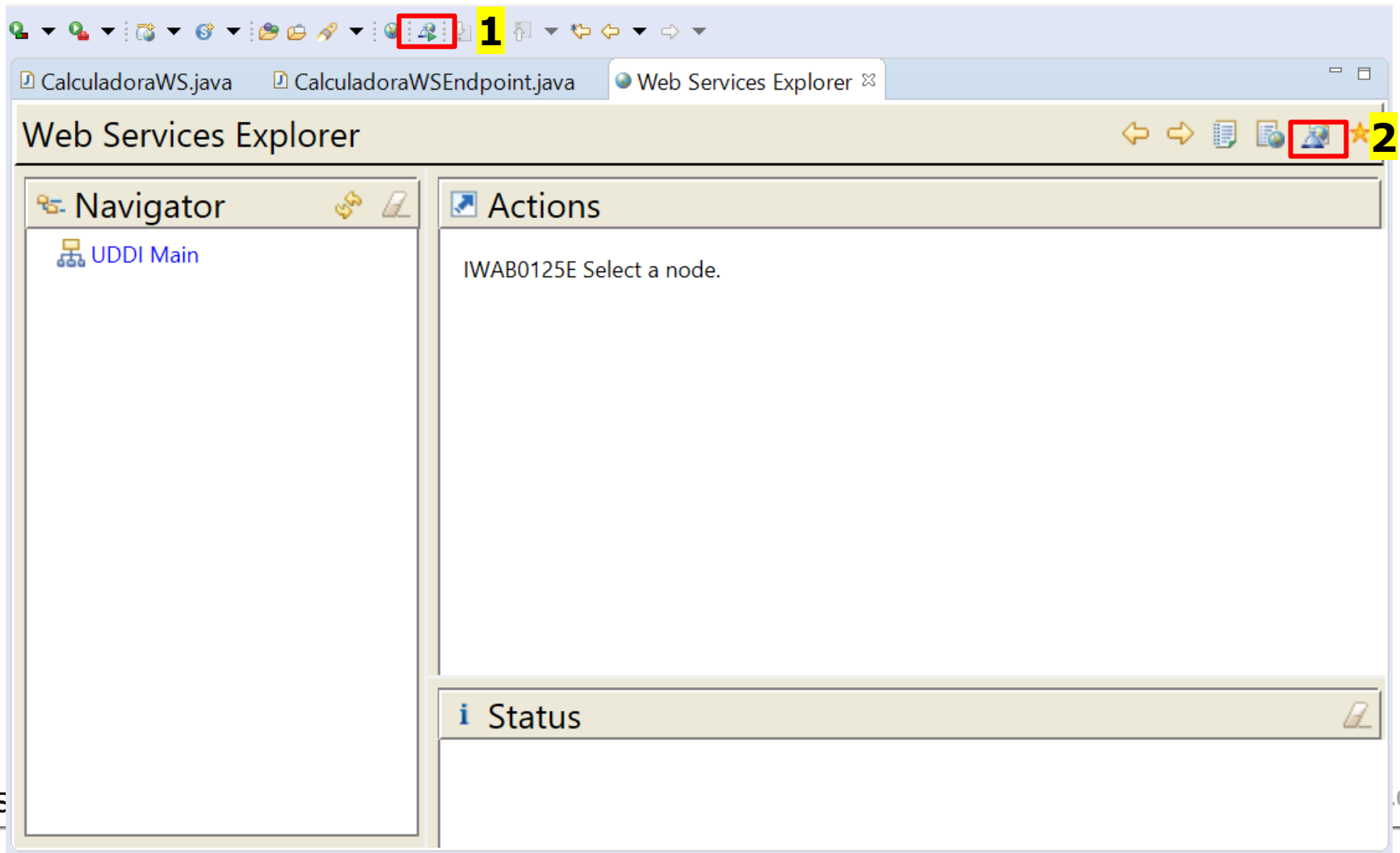
- somar

**Endpoint address:** <http://localhost:8080/sd-calculadora-ws/services/CalculadoraWSPort>

**WSDL :** <http://ws.calculadora.sd.com.br/CalculadoraWSService>

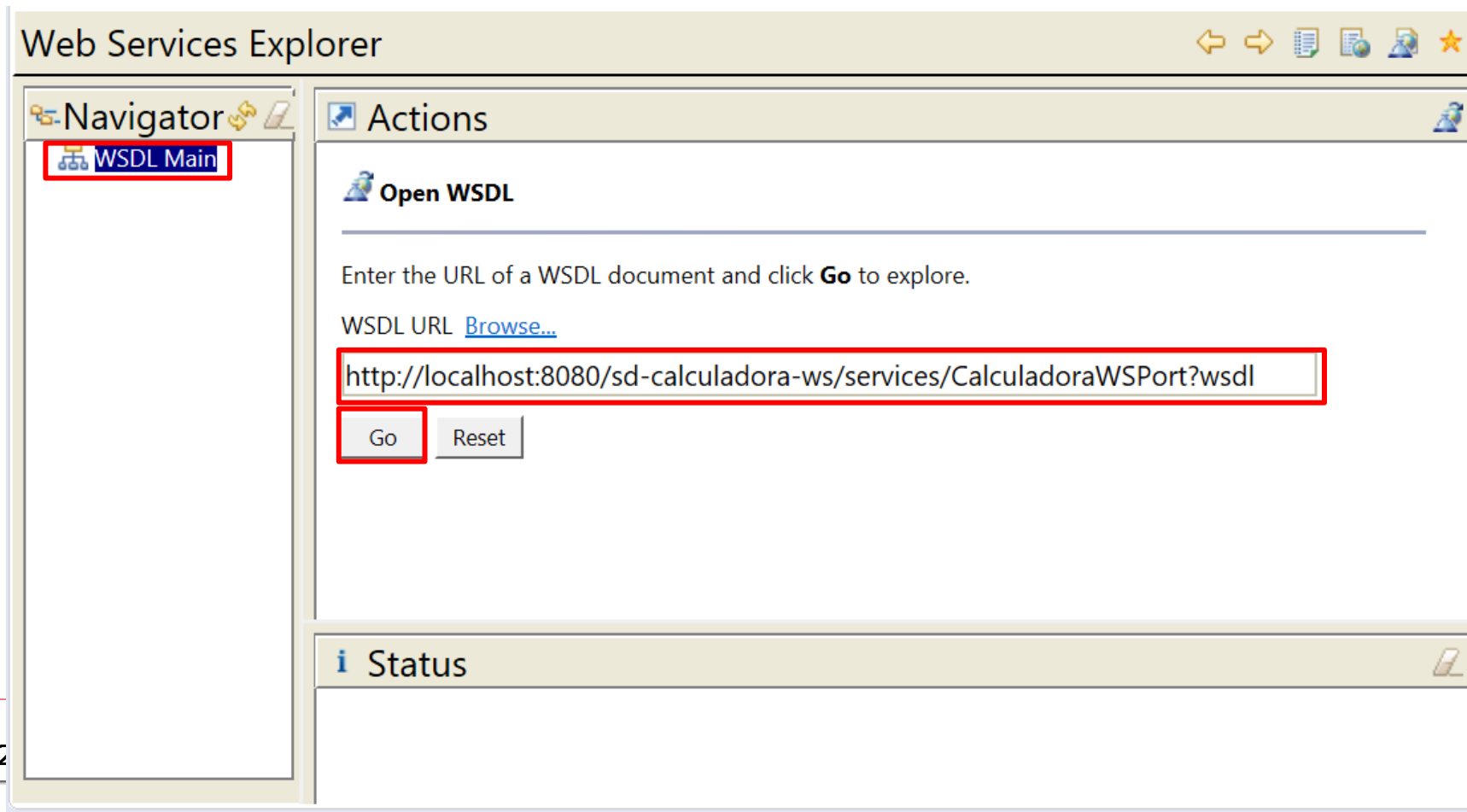
**Target namespace:** <http://ws.calculadora.sd.com.br/>

# Implementando as estruturas básicas do Apache CXF



# Implementando as estruturas básicas do Apache CXF

- ❑ Incluir a URL do serviço, em clicar em “GO”



# Implementando as estruturas básicas do Apache CXF

---

- ❑ Clicar em “SOMAR” da Aba Operations.

# Implementando as estruturas básicas do Apache CXF

The screenshot displays the Web Services Explorer application. On the left, the **Navigator** pane shows a tree structure with the following items: **WSDL Main**, **http://localhost:8080/sd-calculadora**, **CalculadoraWSService**, **CalculadoraWSServiceSoapBinding**, and **somar**. The **somar** item is selected and highlighted with a blue box.

The main area is titled **Actions** and contains the **Invoke a WSDL Operation** section. It includes the instruction: "Enter the parameters for the WSDL operation 'somar' and click **Go** to invoke." Below this, the **Endpoints** section shows a dropdown menu with the selected endpoint: `http://localhost:8080/sd-calculadora-ws/services/CalculadoraWSPort`.

The **Body** section is expanded, showing the **somar** operation. It lists two arguments: **arg0 int** with a value of `2`, and **arg1 int** with a value of `3`. Both input fields are highlighted with red boxes. At the bottom of this section, the **Go** button is also highlighted with a red box, next to a **Reset** button.

The bottom pane is titled **Status** and shows the result of the invocation: **somarResponse** with the return value `return (int): 5`.

# A função dos arquivos XSD e WSDL dentro de um serviço SOAP

- ❑ O arquivo XSD (*XML Schema Definition*) – responsável por expor um esquema que vai nos informar o formato de determinada sequência de dados.



```
1 <xs:schema xmlns:tns="http://ws.calculadora.sd.com.br/" xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="u
2
3 <xs:element name="somar" type="tns:somar"/>
4
5 <xs:element name="somarResponse" type="tns:somarResponse"/>
6
7 <xs:complexType name="somar">
8 <xs:sequence>
9 <xs:element name="arg0" type="xs:int"/>
10 <xs:element name="arg1" type="xs:int"/>
11 </xs:sequence>
12 </xs:complexType>
13
14 <xs:complexType name="somarResponse">
15 <xs:sequence>
16 <xs:element name="return" type="xs:int"/>
17 </xs:sequence>
18 </xs:complexType>
19
20 </xs:schema>
21
```

# A função dos arquivos XSD e WSDL dentro de um serviço SOAP

- ❑ A principal função de um WSDL é descrever como um webservice deve funcionar.

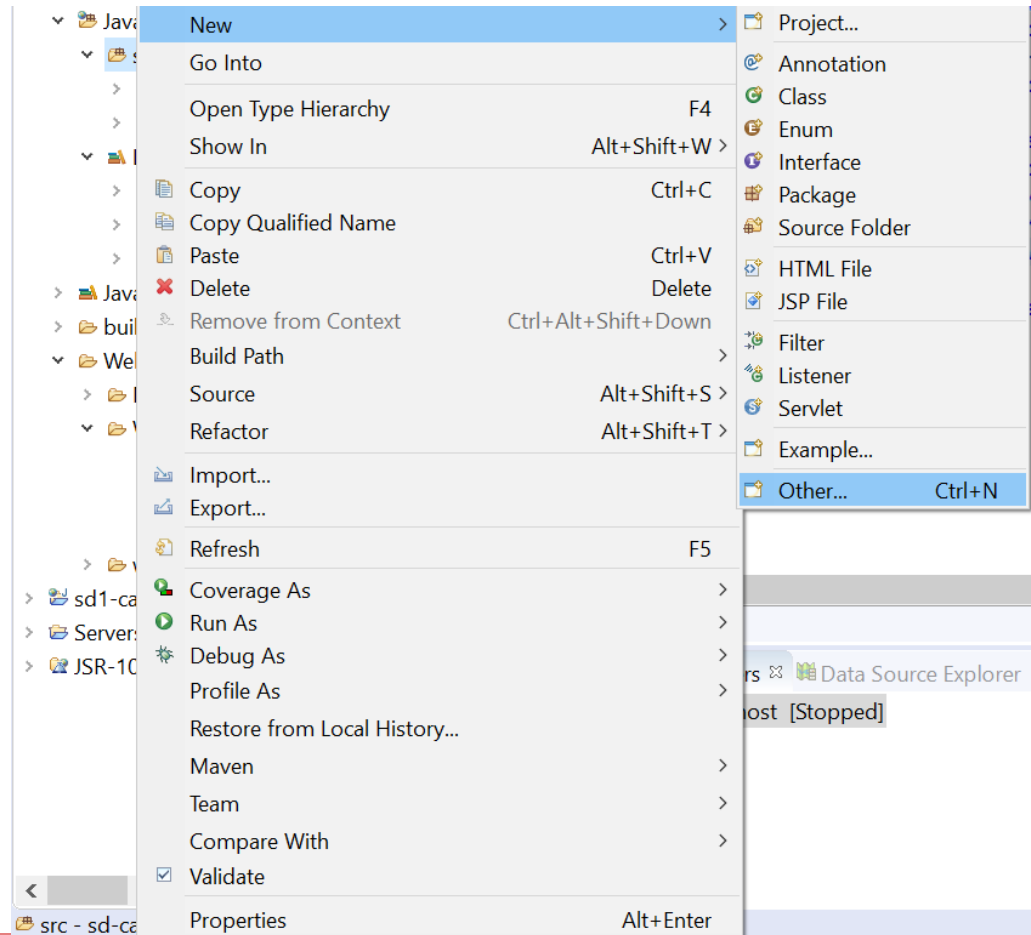
A screenshot of an IDE window showing a WSDL file named 'calculadoraws.wsdl'. The window has several tabs at the top: 'CalculadoraWS.java', 'calculadoraws\_schema1.xsd', 'CalculadoraWSEndpoint.java', and 'calculadoraws.wsdl'. The 'calculadoraws.wsdl' tab is active, displaying XML code. The code defines a service named 'CalculadoraWSService' with a target namespace of 'http://ws.calculadora.sd.com.br/'. It includes a schema import for 'http://www.w3.org/2001/XMLSchema'. The service has two messages: 'somarResponse' and 'somar'. The 'somar' message has a single part named 'parameters' of type 'tns:somar'. The service has a port named 'CalculadoraWSEndpoint' with an operation named 'somar'. The operation has an input named 'somar' of type 'tns:somar' and an output named 'somarResponse' of type 'tns:somarResponse'. The service is bound to a SOAP binding named 'CalculadoraWSServiceSoapBinding' with a type of 'tns:CalculadoraWSEndpoint' and a transport of 'http://schemas.xmlsoap.org/soap/http'.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <wsdl:definitions name="CalculadoraWSService" targetNamespace="http://ws.calculadora.sd.com.br/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
3   <wsdl:types>
4     <schema xmlns="http://www.w3.org/2001/XMLSchema"
5       <import namespace="http://ws.calculadora.sd.com.br/" schemaLocation="calculadoraws_schema1.xsd"/>
6   </schema>
7   </wsdl:types>
8   <wsdl:message name="somarResponse">
9     <wsdl:part name="parameters" element="tns:somarResponse">
10    </wsdl:part>
11  </wsdl:message>
12  <wsdl:message name="somar">
13    <wsdl:part name="parameters" element="tns:somar">
14    </wsdl:part>
15  </wsdl:message>
16  <wsdl:portType name="CalculadoraWSEndpoint">
17    <wsdl:operation name="somar">
18      <wsdl:input name="somar" message="tns:somar">
19      </wsdl:input>
20      <wsdl:output name="somarResponse" message="tns:somarResponse">
21      </wsdl:output>
22    </wsdl:operation>
23  </wsdl:portType>
24  <wsdl:binding name="CalculadoraWSServiceSoapBinding" type="tns:CalculadoraWSEndpoint">
25    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
```



# Gerando um cliente para um serviço SOAP com Apache CXF

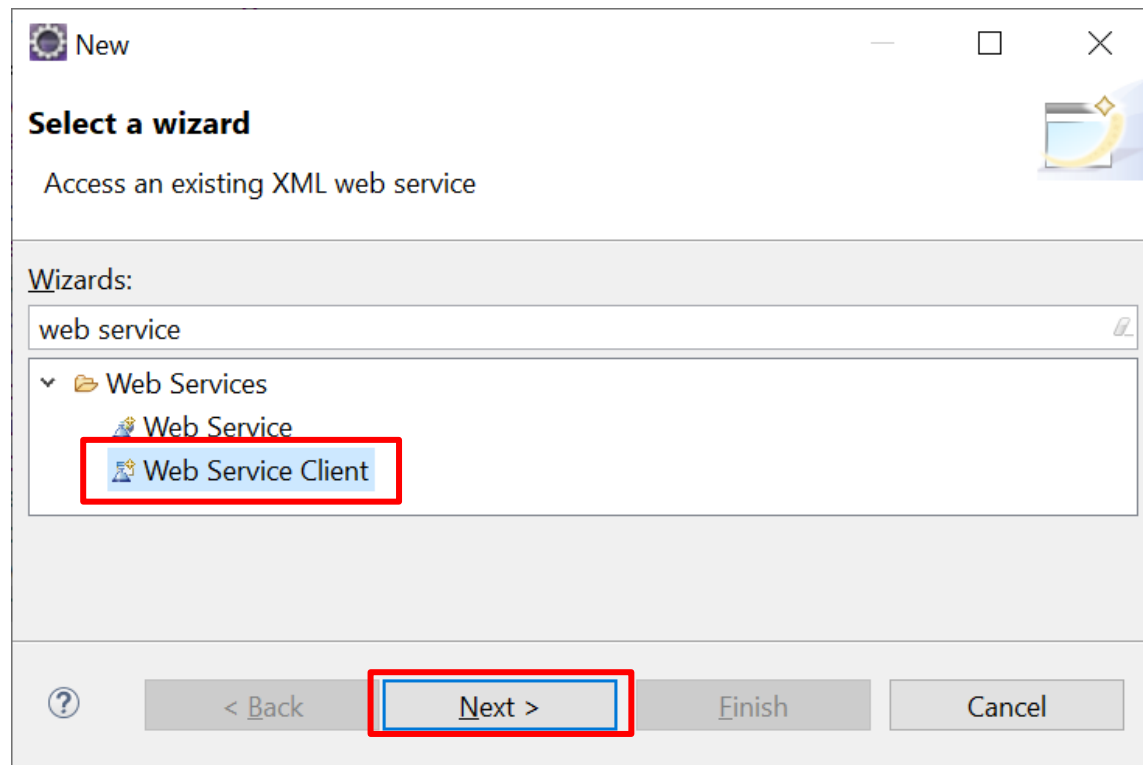
- ❑ Clique com o botão direito na pasta "src", selecione "New", em seguida "Other".



# Gerando um cliente para um serviço SOAP com Apache CXF

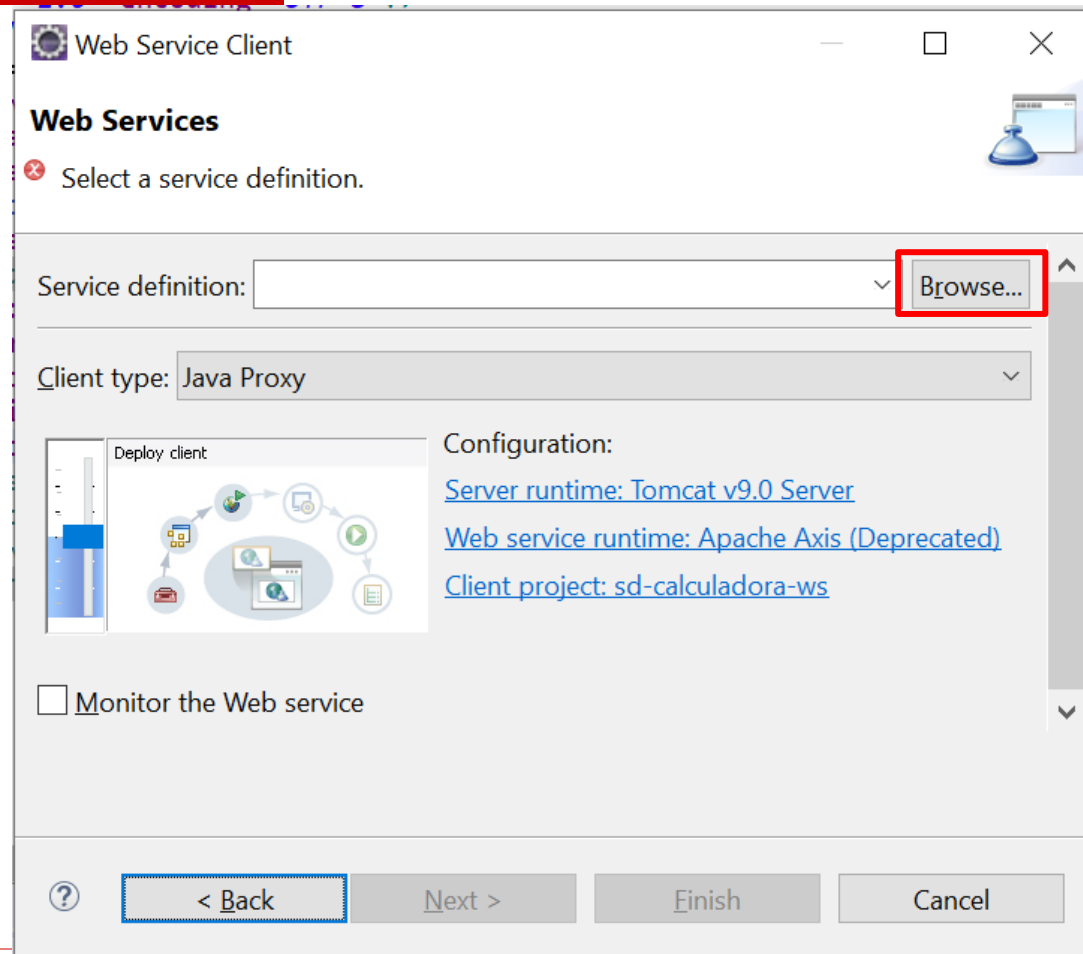
---

- ❑ Na tela de busca, digite “Web Service”.
- ❑ Selecione a opção “Web Service Client”

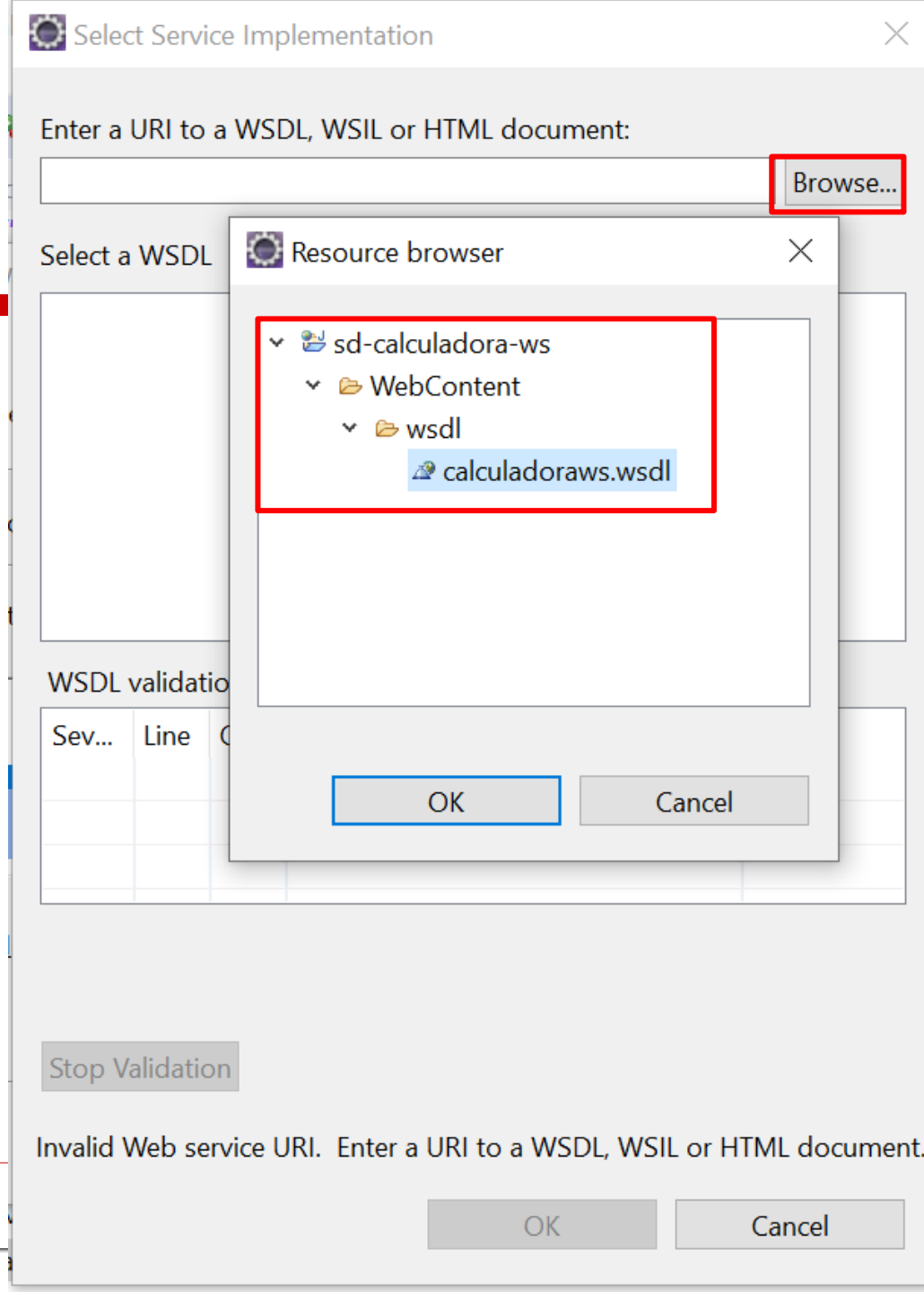


# Gerando um cliente para um serviço SOAP com Apache CXF

- ❑ O “Service Definition” é o WSDL. Através do WSDL, o eclipse vai conseguir ter acesso ao XSD e ao próprio WSDL.

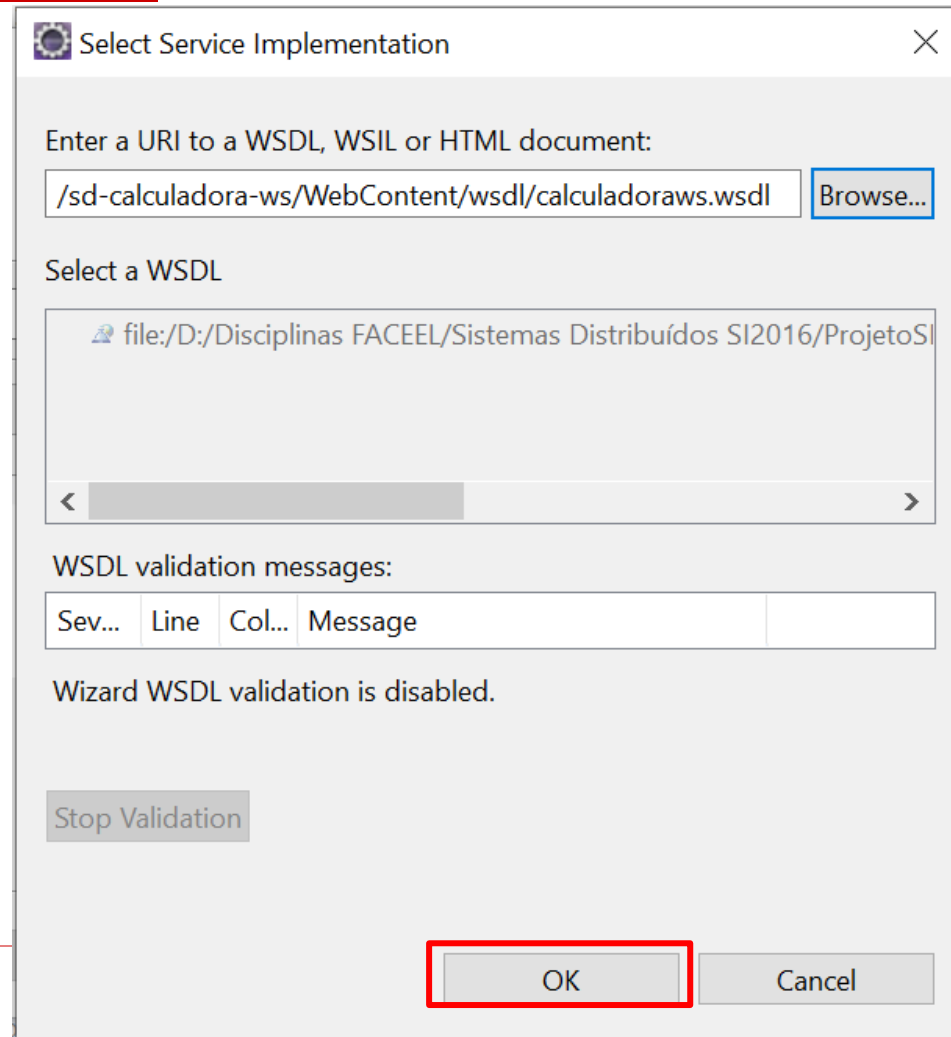


- ❑ Seleccione o arquivo "calculadora.wsdl" e clique em "OK"



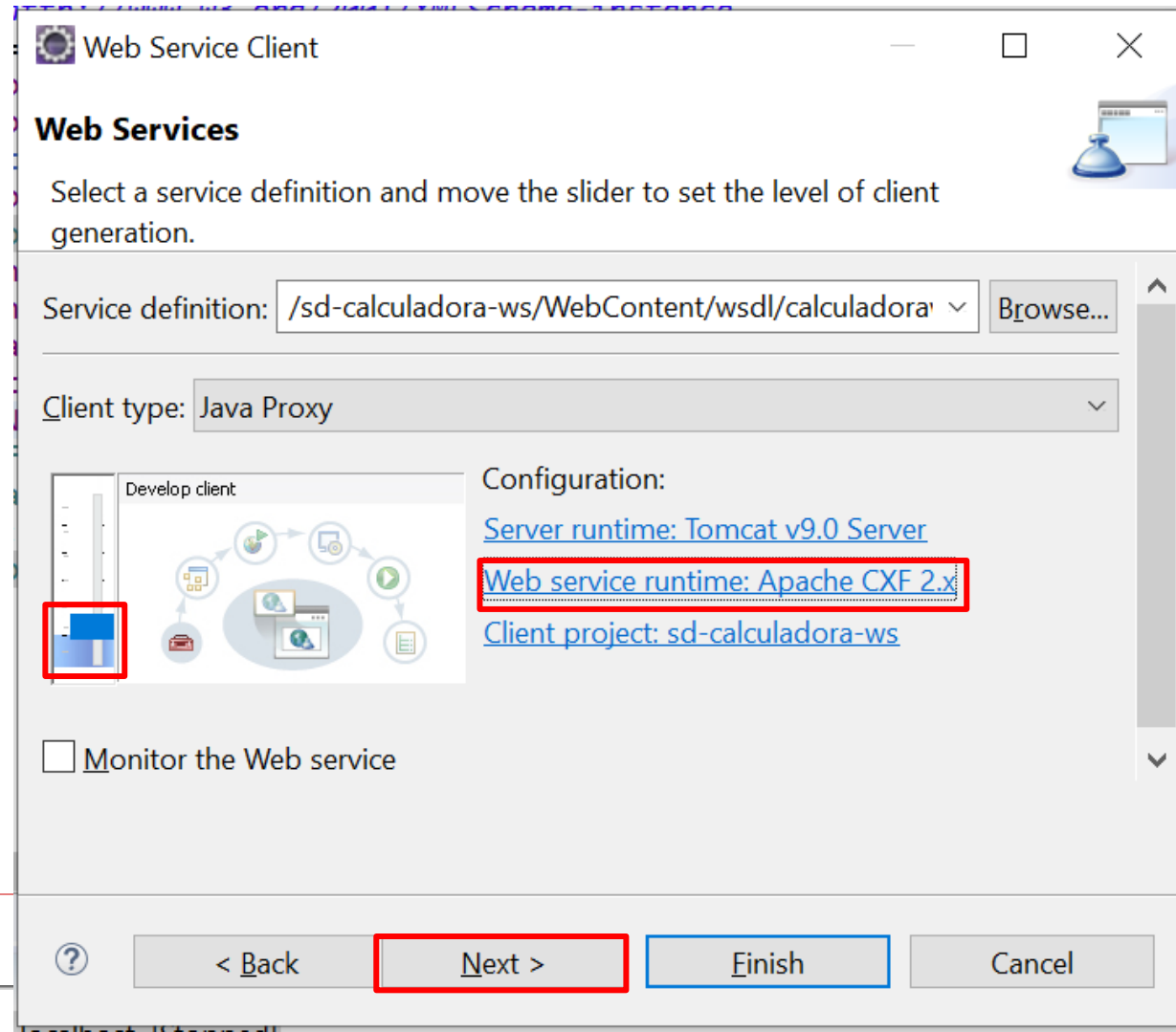
# Gerando um cliente para um serviço SOAP com Apache CXF

- ❑ Confirme a seleção do WSDL no botão "OK".



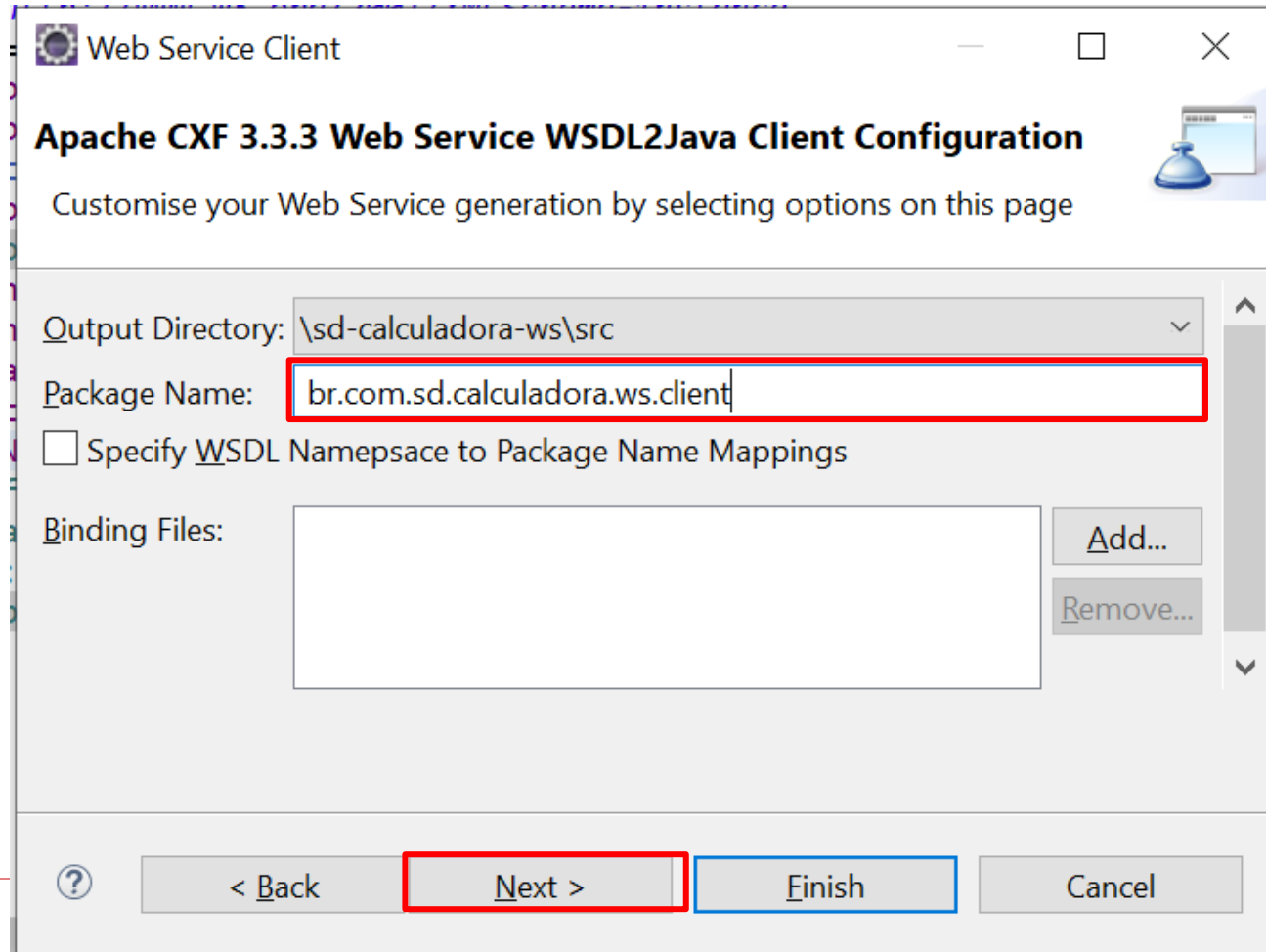
# Gerando um cliente para um serviço SOAP com Apache CXF

- ☐ Desça a barra ao lado para somente gerar o código.
- ☐ Selecione o Apache CXF.



# Gerando um cliente para um serviço SOAP com Apache CXF

- ❑ Atribua um nome para o Pacote.
- ❑ Clique em "NEXT".



Web Service Client

**Apache CXF 3.3.3 Web Service WSDL2Java Client Configuration**

Customise your Web Service generation by selecting options on this page

Output Directory: \\sd-calculadora-ws\\src

Package Name: **br.com.sd.calculadora.ws.client**

☐ Specify WSDL Namespace to Package Name Mappings

Binding Files:

Add...

Remove...

< Back **Next >** Finish Cancel

# Gerando um cliente para um serviço SOAP com Apache CXF

- ☐ Selecione o checkbox "-mark-generated".
- ☐ Clique em "Finish".

Web Service Client

**Apache CXF 3.3.3 Web Service WSDL2Java Client Configuration**

Customise your Web Service generation by selecting options on this page

WSDL2Java Generation Options

- ☒ Generate Implementation
- ☒ Generate default values for the generated artifacts
- ☐ Enable processing of implicit SOAP headers
- ☒ Enable loading of the default namespace package name mapping
- ☒ Enable the use of the default excludes namespace mapping
- ☐ Enable Auto Name Resolution
- ☐ Use the Apache CXF proprietary WS-Addressing type

XJC Arguments

XJC Arg	Description
<input type="checkbox"/> -Xdv	Initialise fields mapped from elements with their default values
<input type="checkbox"/> -Xts	Activate plugin to add a toString() method to generated classes
<input type="checkbox"/> -Xts:style:multiline	Have toString produce multi line output
<input type="checkbox"/> -Xts:style:simple	Have toString produce single line terse output
<input type="checkbox"/> -Xlocator	Enable source location support for generated code
<input type="checkbox"/> -Xsync-methods	Generate accessor methods with the 'synchronized' keyword
<input checked="" type="checkbox"/> -mark-generated	Add @Generated annotations to the generated code

< Back Next > **Finish** Cancel



# Gerando um cliente para um serviço SOAP com Apache CXF

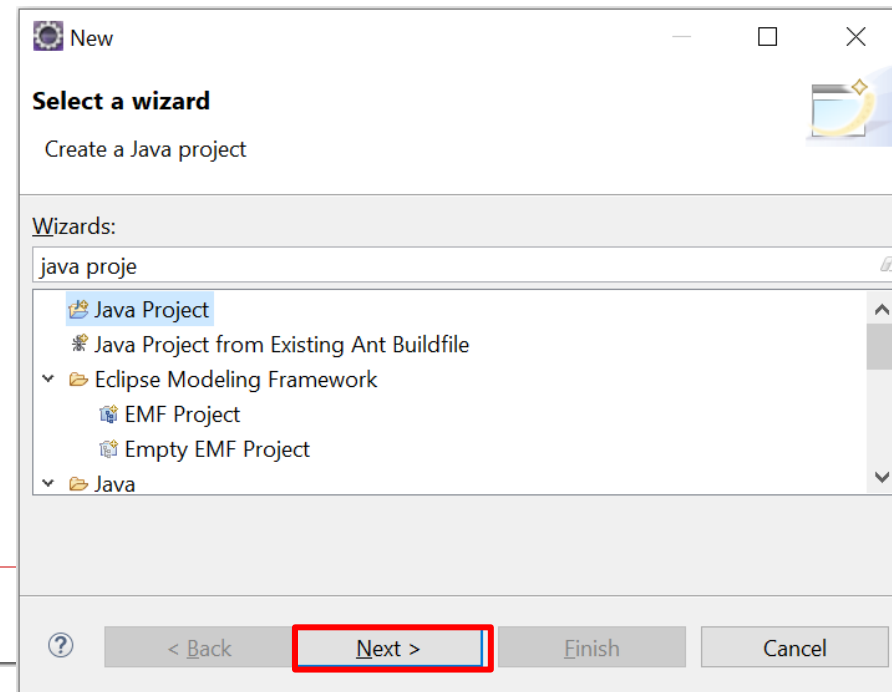
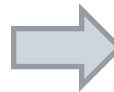
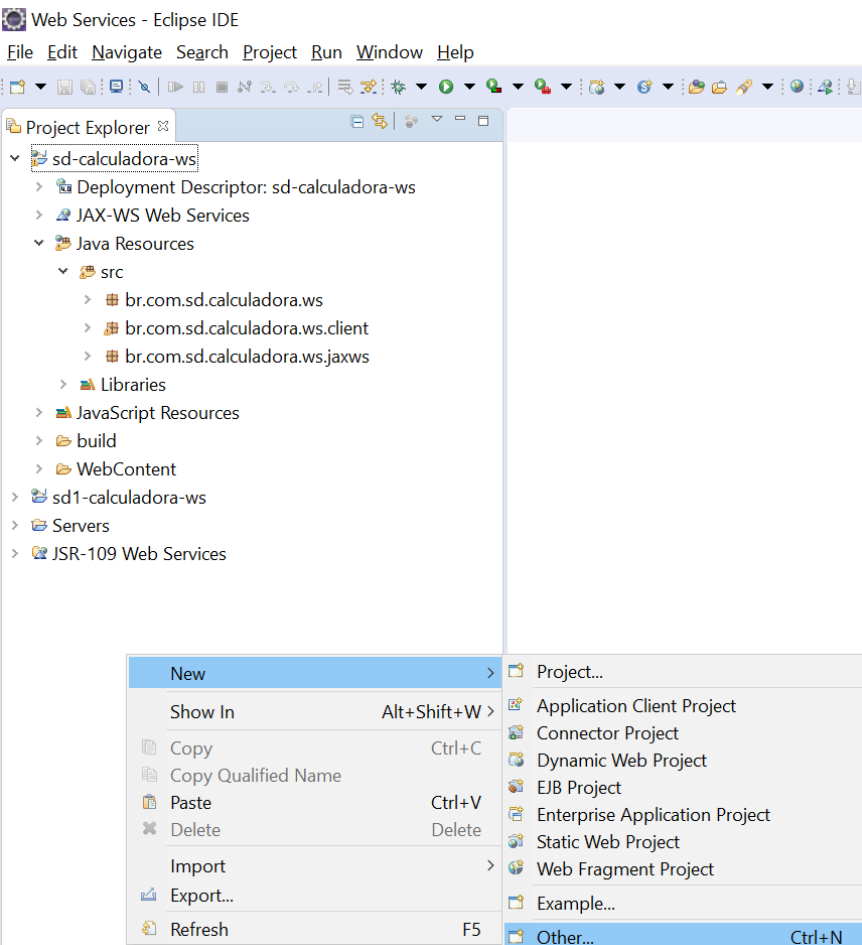
---

- ❑ Todas as classes listadas no pacote, são necessárias para acessar o web service.

```
▼ br.com.sd.calculadora.ws.client
  > CalculadoraWSEndpoint_CalculadoraWSPort_Client.java
  > CalculadoraWSEndpoint.java
  > CalculadoraWSPortImpl.java
  > CalculadoraWSService.java
  > ObjectFactory.java
  > package-info.java
  > Somar.java
  > SomarResponse.java
```

# Gerando um cliente para um serviço SOAP com Apache CXF

❑ Criar um novo projeto java.



# Gerando um cliente para um serviço SOAP com Apache CXF

- ❑ Defina o nome do projeto e selecione o java versão 8.
- ❑ Clique em "next", para prosseguir.

New Java Project

Create a Java Project

Create a Java project in the workspace or in an external location.

Project name: sd-calculadora-ws-client

☒ Use default location

Location: D:\Disciplinas FACEEL\Sistemas Distribuídos SI2016\Proj Browse...

JRE

☐ Use an execution environment JRE: JavaSE-1.8

☒ Use a project specific JRE: jre1.8.0\_221

☐ Use default JRE (currently 'jre1.8.0\_221') [Configure JREs...](#)

Project layout

☐ Use project folder as root for sources and class files

☒ Create separate folders for sources and class files [Configure default...](#)

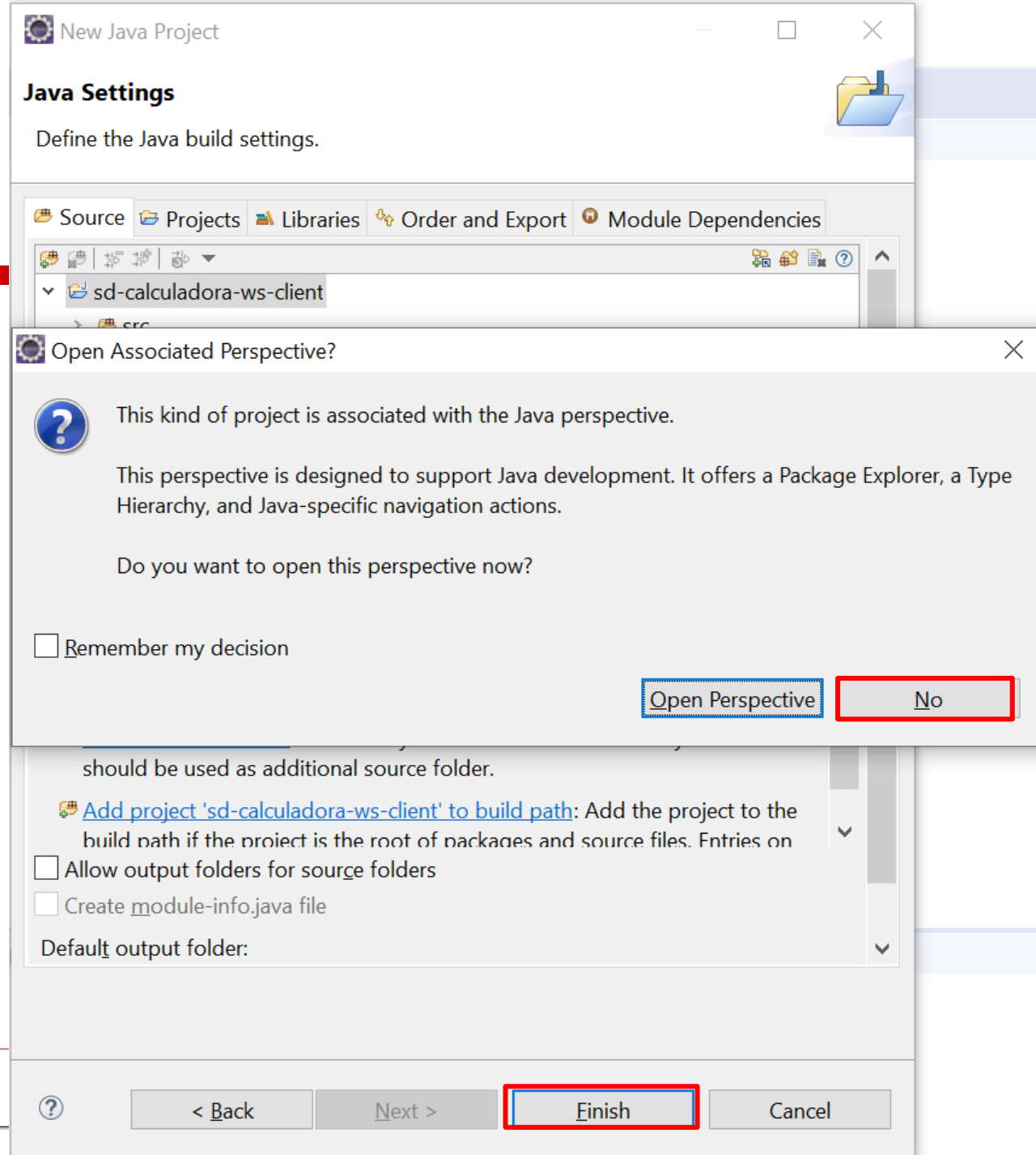
Working sets

☐ Add project to working sets New...

Working sets: Select...

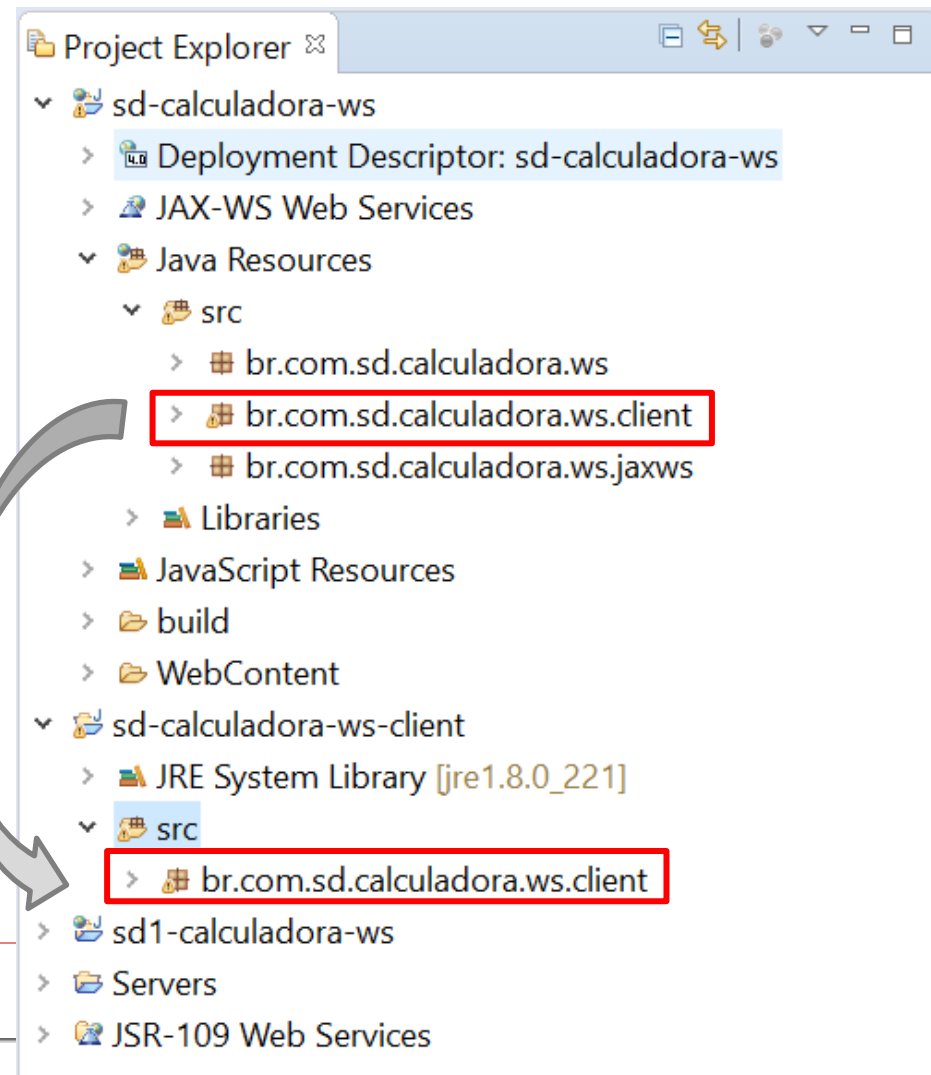
[?](#) < Back **Next >** Finish Cancel

❑ Clique em “Finish” e em seguida, clique em “No” para manter a perspectiva Java.



# Gerando um cliente para um serviço SOAP com Apache CXF

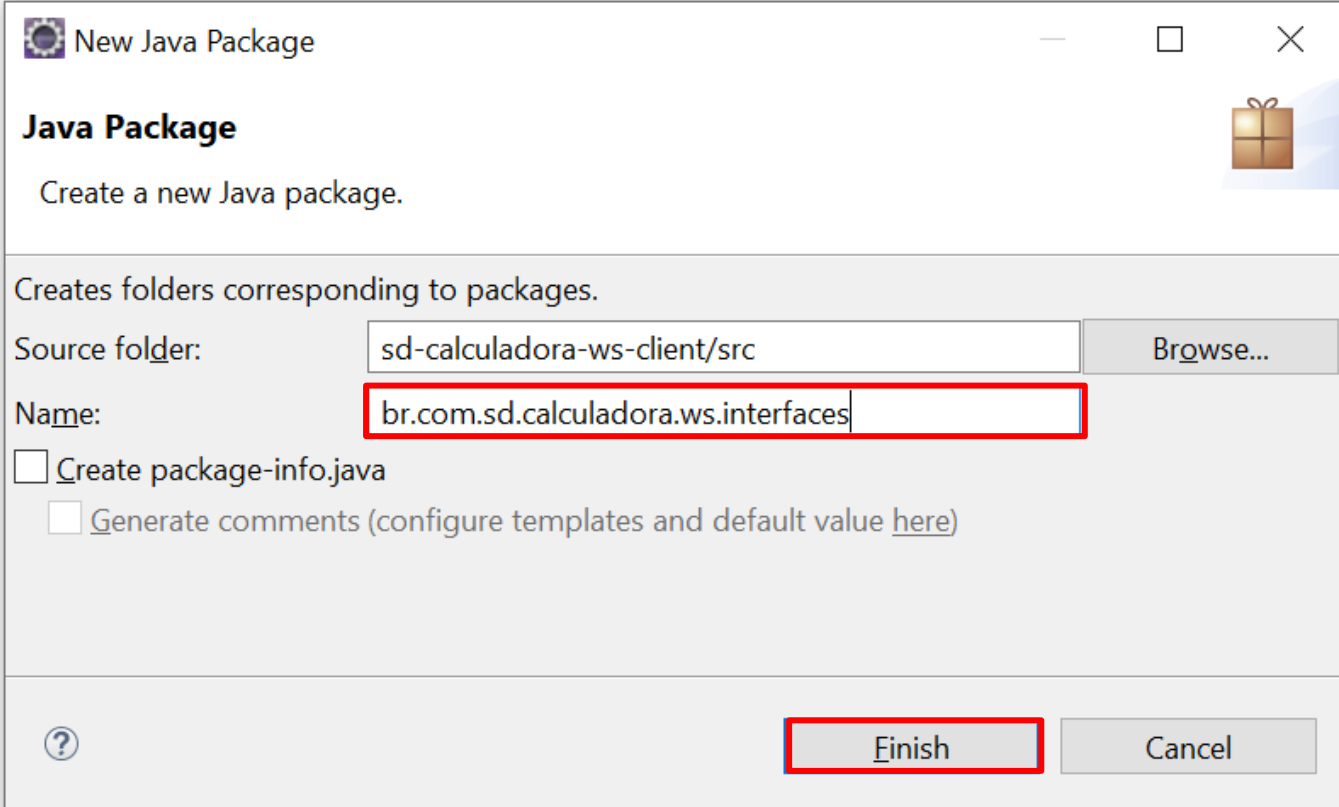
**Copiar o pacote de um projeto para o outro**



# Gerando um cliente para um serviço SOAP com Apache CXF

---

- ❑ A partir do projeto “sd-calculadora-ws-client”, crie um novo pacote.



**New Java Package**

**Java Package**  
Create a new Java package.

Creates folders corresponding to packages.

Source folder:  [Browse...](#)

Name:

☐ Create package-info.java

☐ Generate comments (configure templates and default value [here](#))

[?](#) [Finish](#) [Cancel](#)

□ A partir do pacote criado, crie uma nova classe java.

New Java Class

**Java Class**

Create a new Java class.

Source folder:  [Browse...](#)

Package:  [Browse...](#)

☐ Enclosing type:  [Browse...](#)

Name:

Modifiers: ☒ public ☐ package ☐ private ☐ protected  
☐ abstract ☐ final ☐ static

Superclass:  [Browse...](#)

Interfaces:  [Add...](#) [Remove](#)

Which method stubs would you like to create?

☒ `public static void main(String[] args);`

☐ Constructors from superclass

☒ Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))

☐ Generate comments

[?](#) [Finish](#) [Cancel](#)

# Gerando um cliente para um serviço SOAP com Apache CXF

---

- ❑ Execute o projeto cliente e veja o resultado.

Main.java calculadoraws\_schema1.xsd calculadoraws.wsdl

```
1 package br.com.sd.calculadora.ws.interfaces;
2
3 import java.net.MalformedURLException;
4 import java.net.URL;
5
6 import br.com.sd.calculadora.ws.client.CalculadoraWSService;
7
8 public class Main {
9
10     public static void main(String[] args) throws MalformedURLException {
11         CalculadoraWSService srv = new CalculadoraWSService(new URL("http://localhost:8080/sd-calculadora-ws/services/CalculadoraWSPort"));
12         System.out.println(srv.getCalculadoraWSPort().somar(2, 3));
13     }
14 }
15
```