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

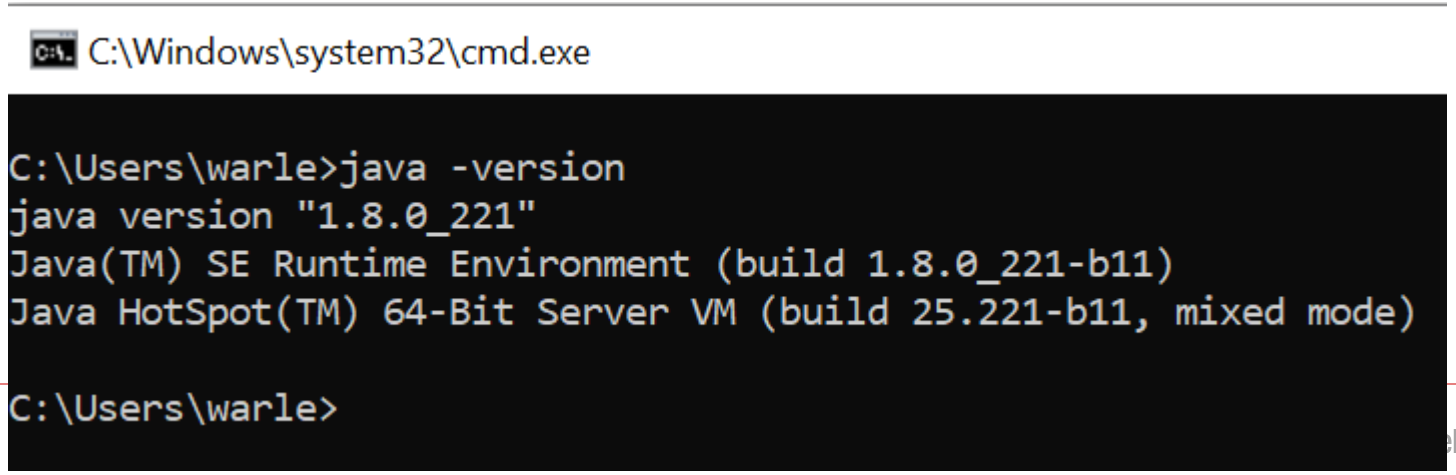
# Agenda

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

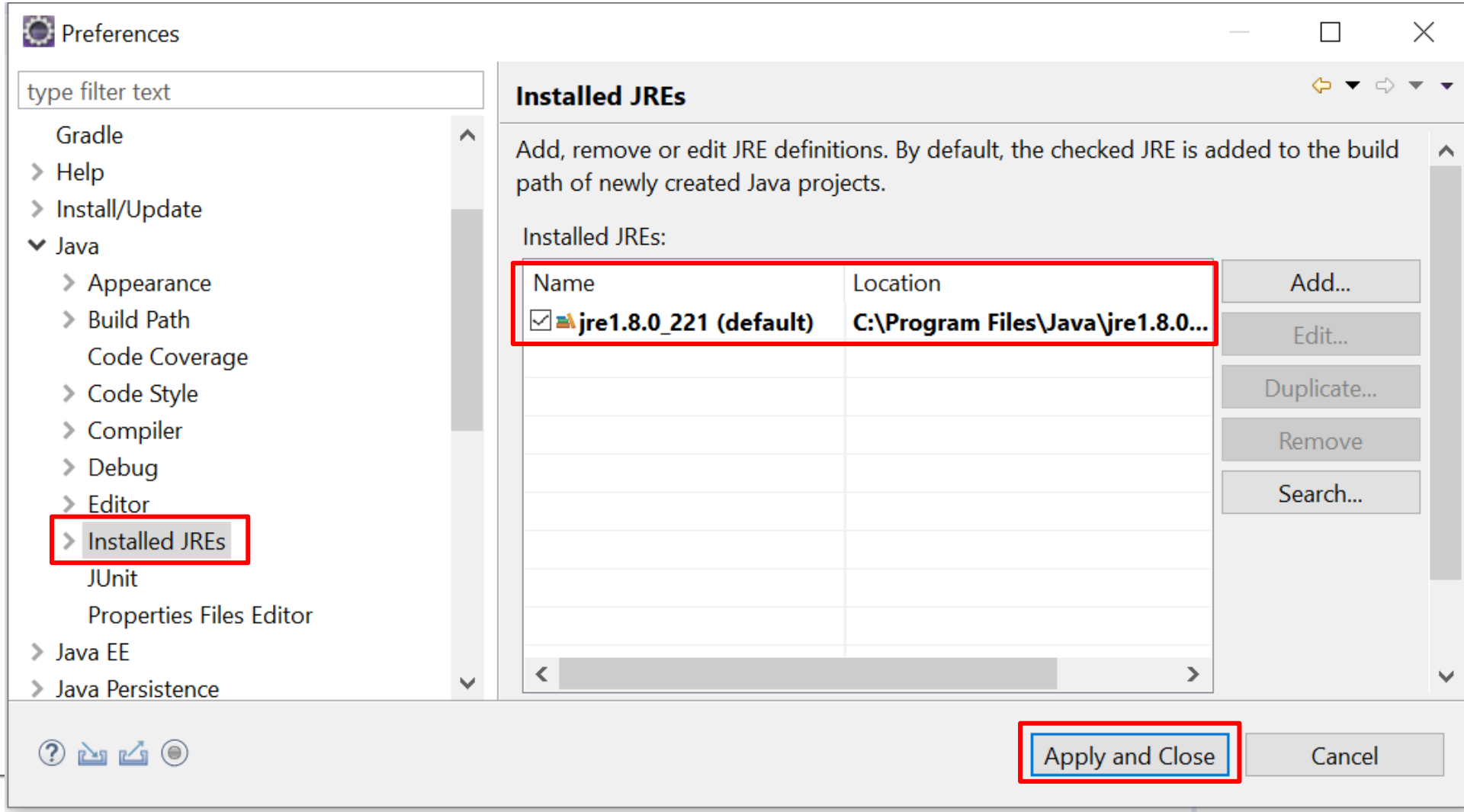
# Verificação da versão do Java 8

- Entre na página web do Apache CXF:
  - http://cxf.apache.org/

- Verifique a versão do java, pois é necessário ter o JAVA 8:

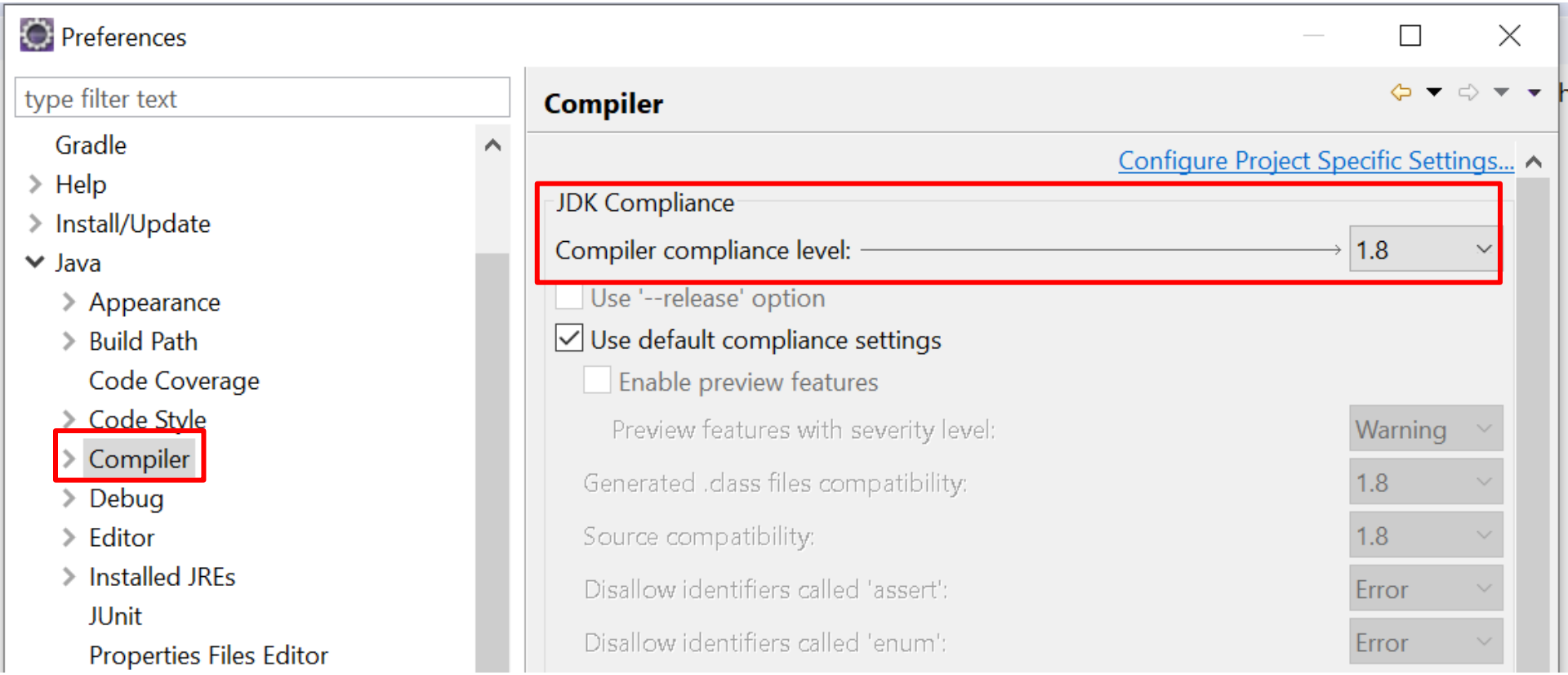


# Criando um Dynamic Web Project no Eclipse



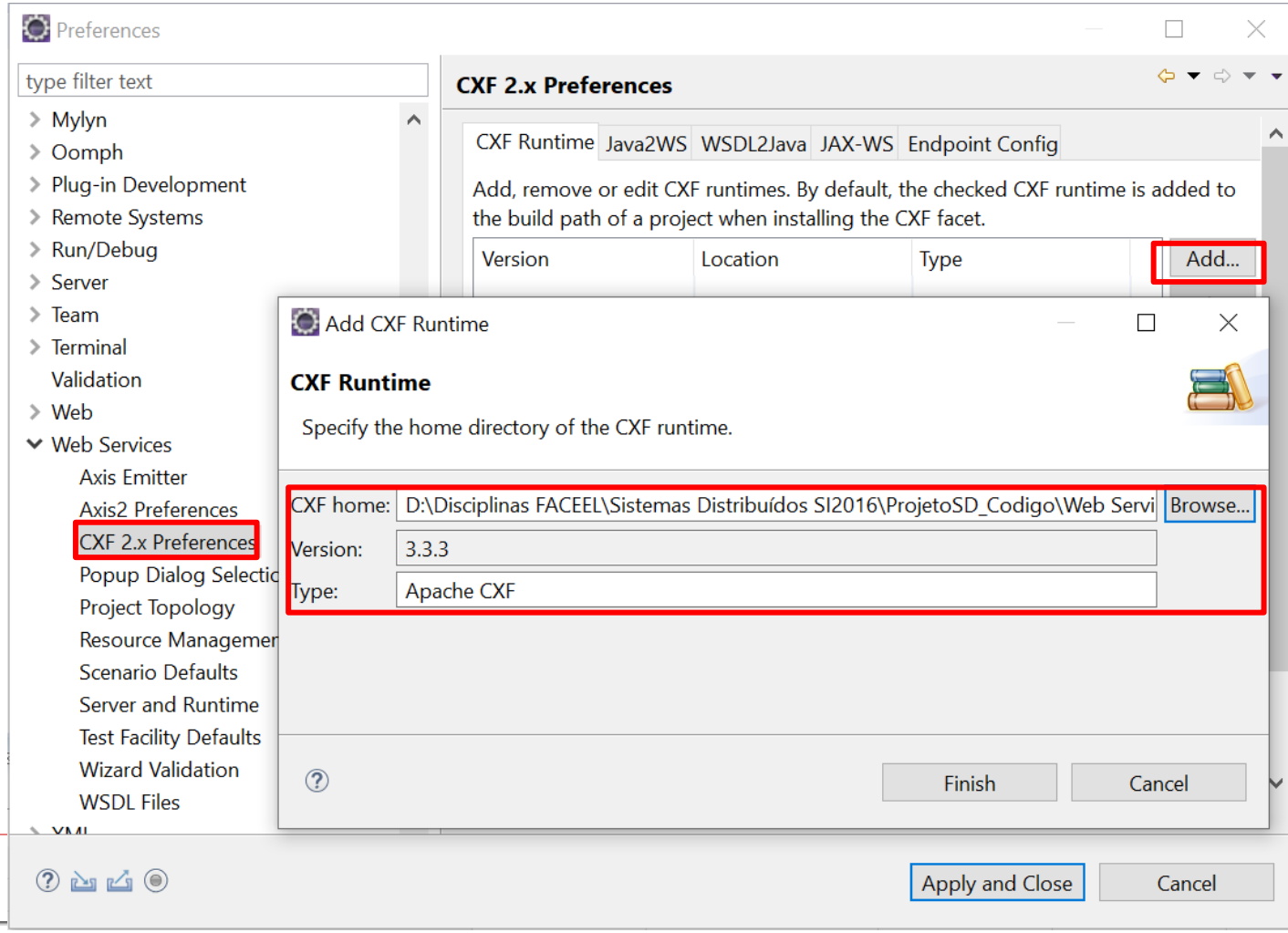
# Criando um Dynamic Web Project no Eclipse

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



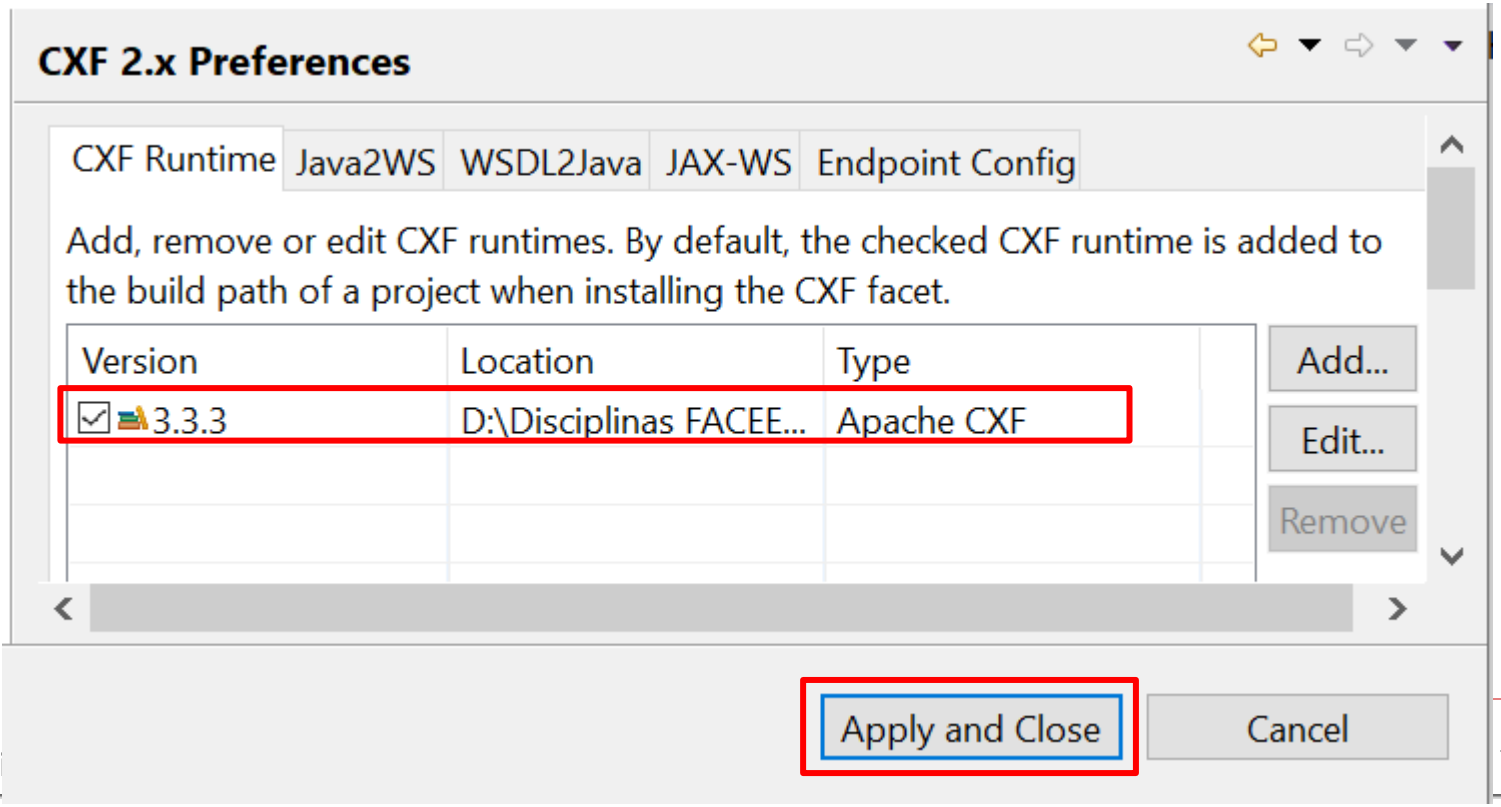
# Criando um Dynamic Web Project no Eclipse

- Adicionar o Apache CXF para integrar com o Eclipse

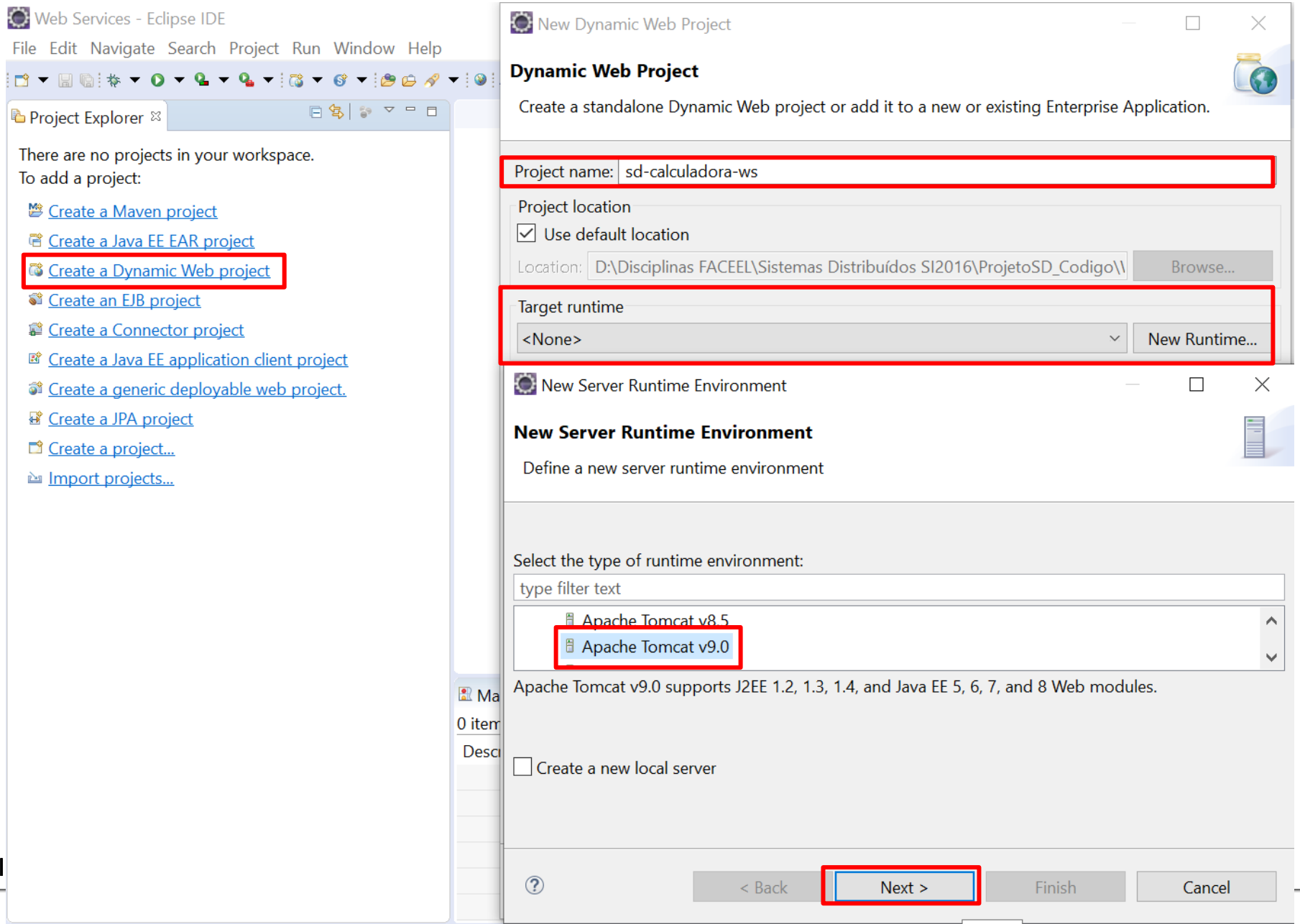


# Criando um Dynamic Web Project no Eclipse

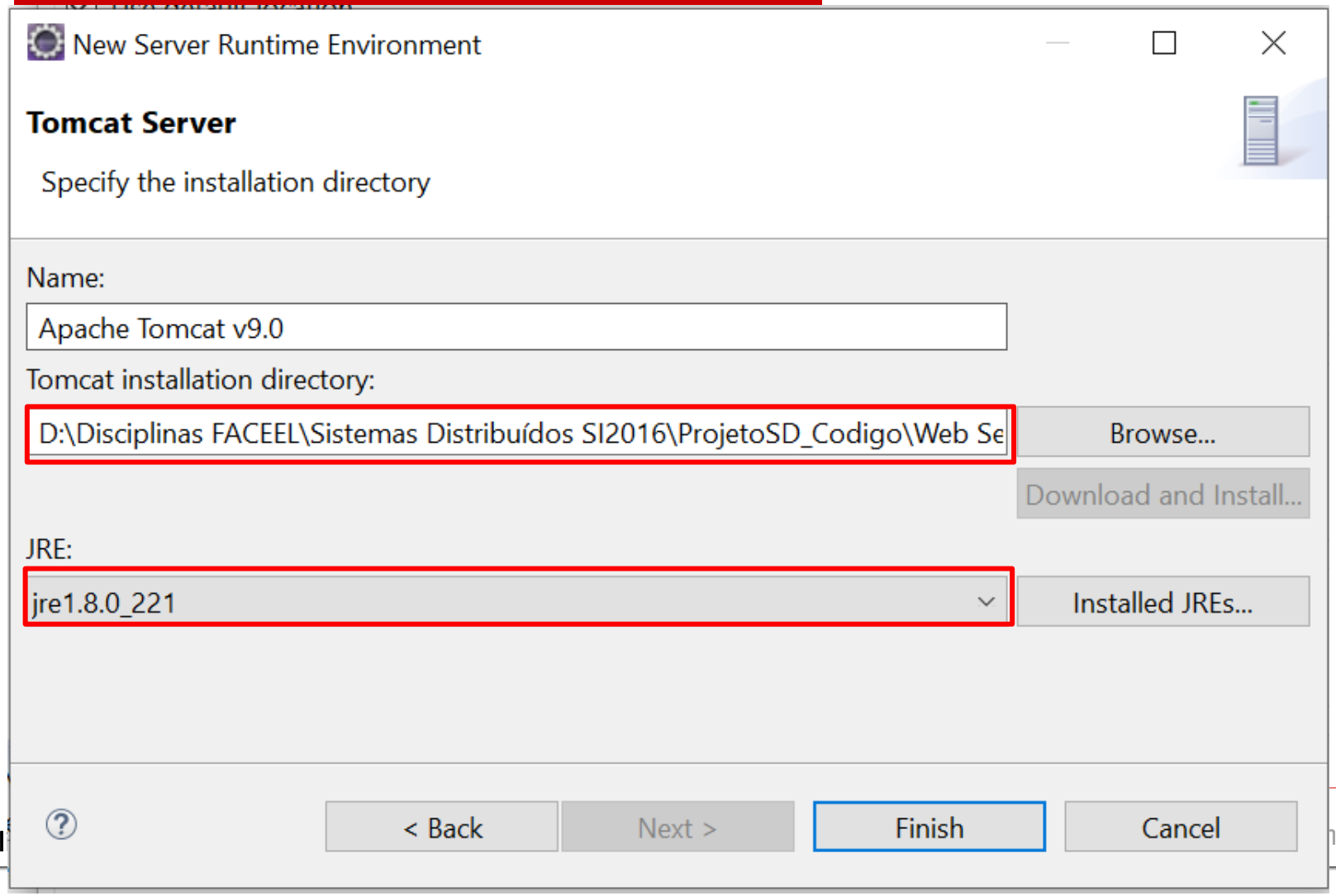
- Selecionar o checkbox do Apache CXF e clicar em "Apply and Close"



# Criar um projeto que use o apache CXF.

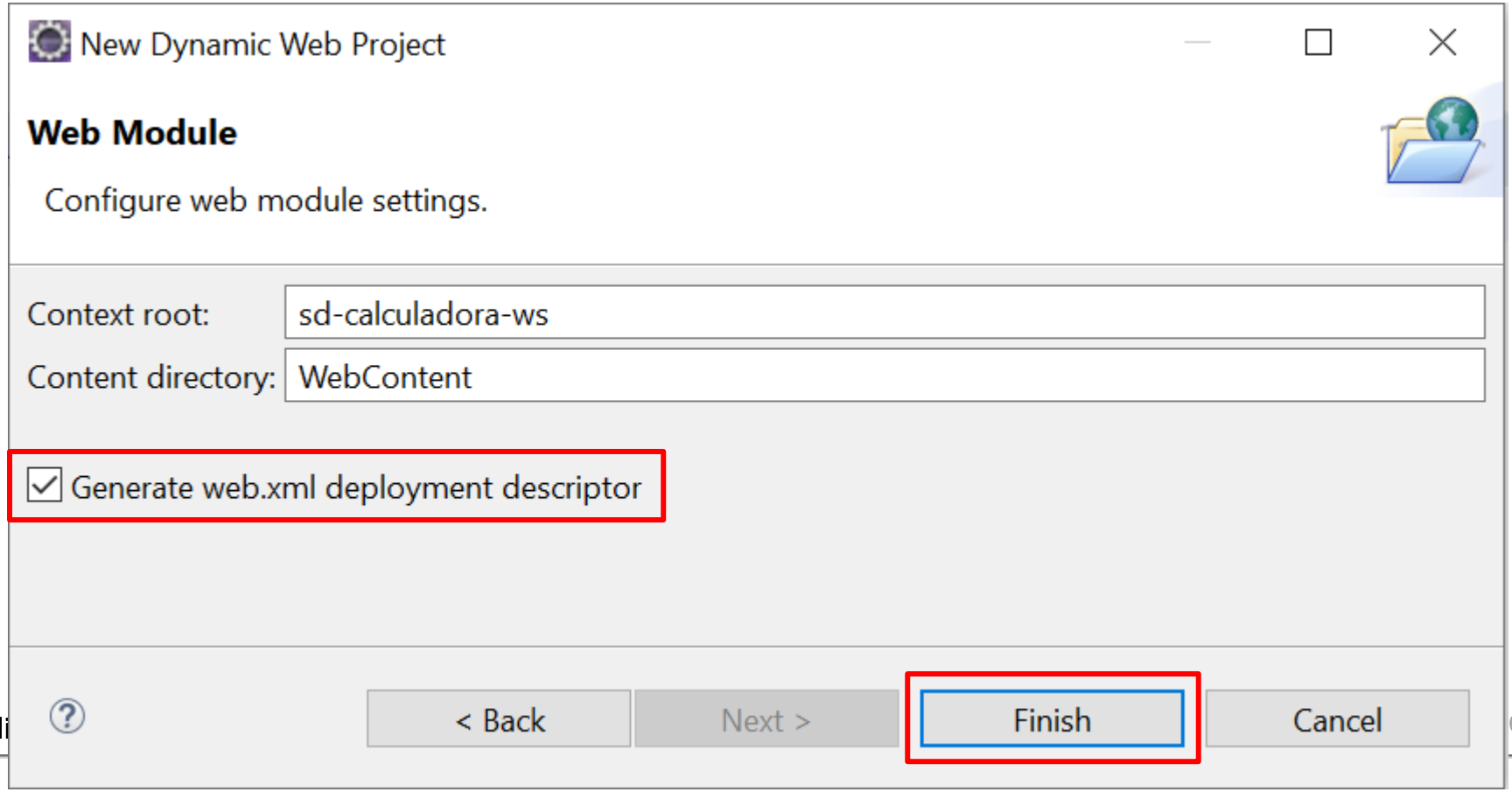


# Criando um Dynamic Web Project no Eclipse



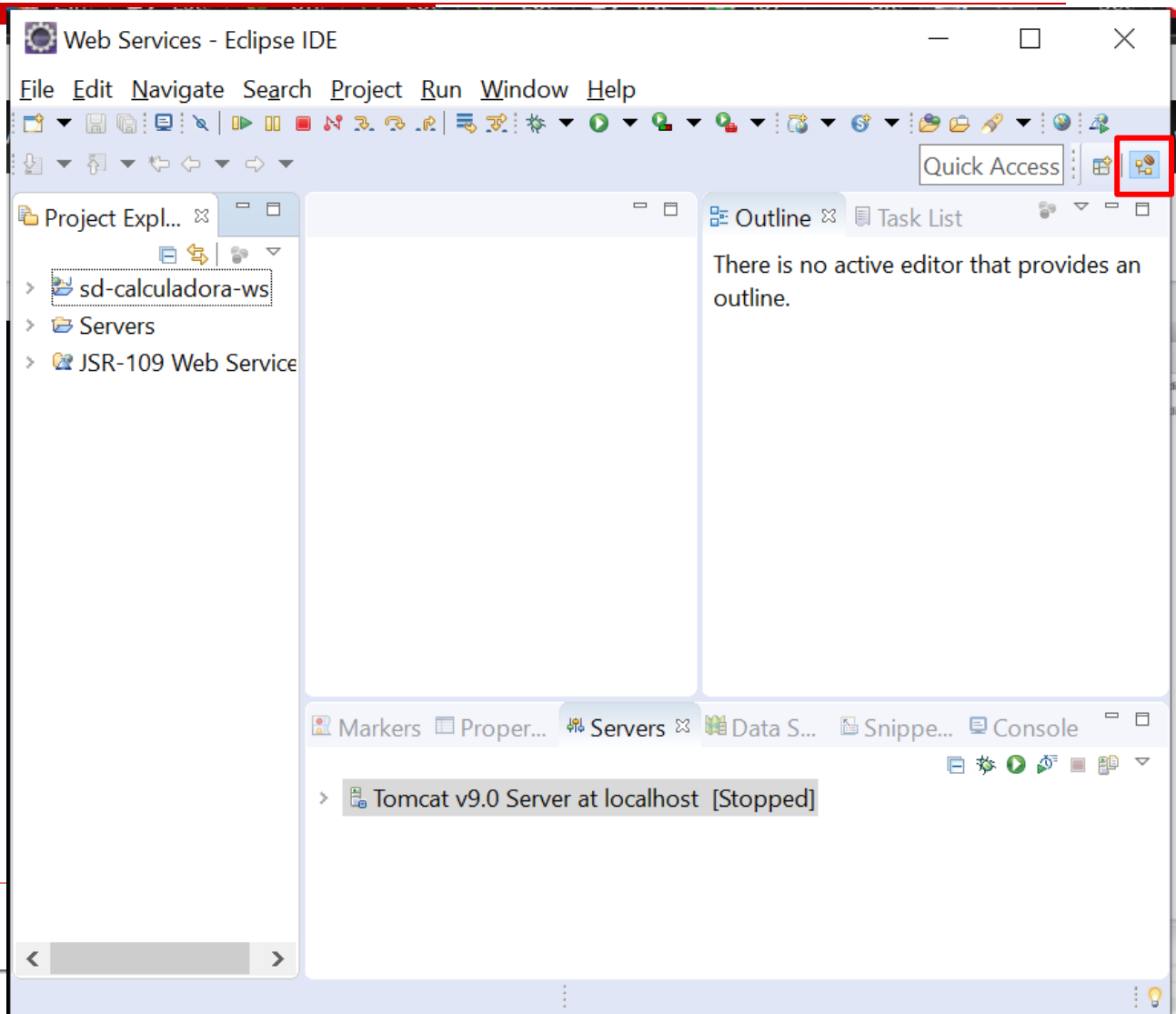
# Criando um Dynamic Web Project no Eclipse

- Clicar no checkbox para gerar o "web.xml"



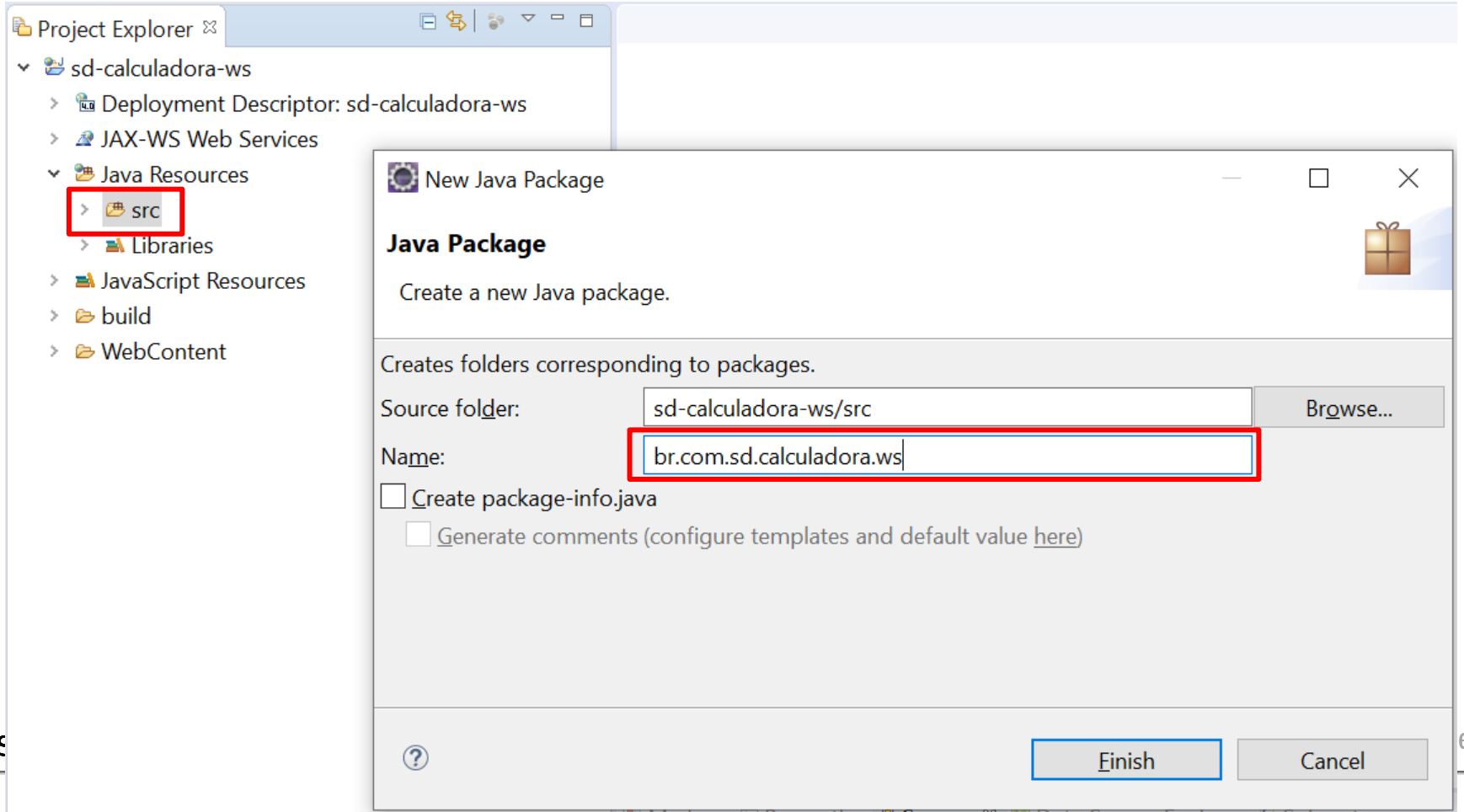
# Criando um Dynamic Web Project no Eclipse

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



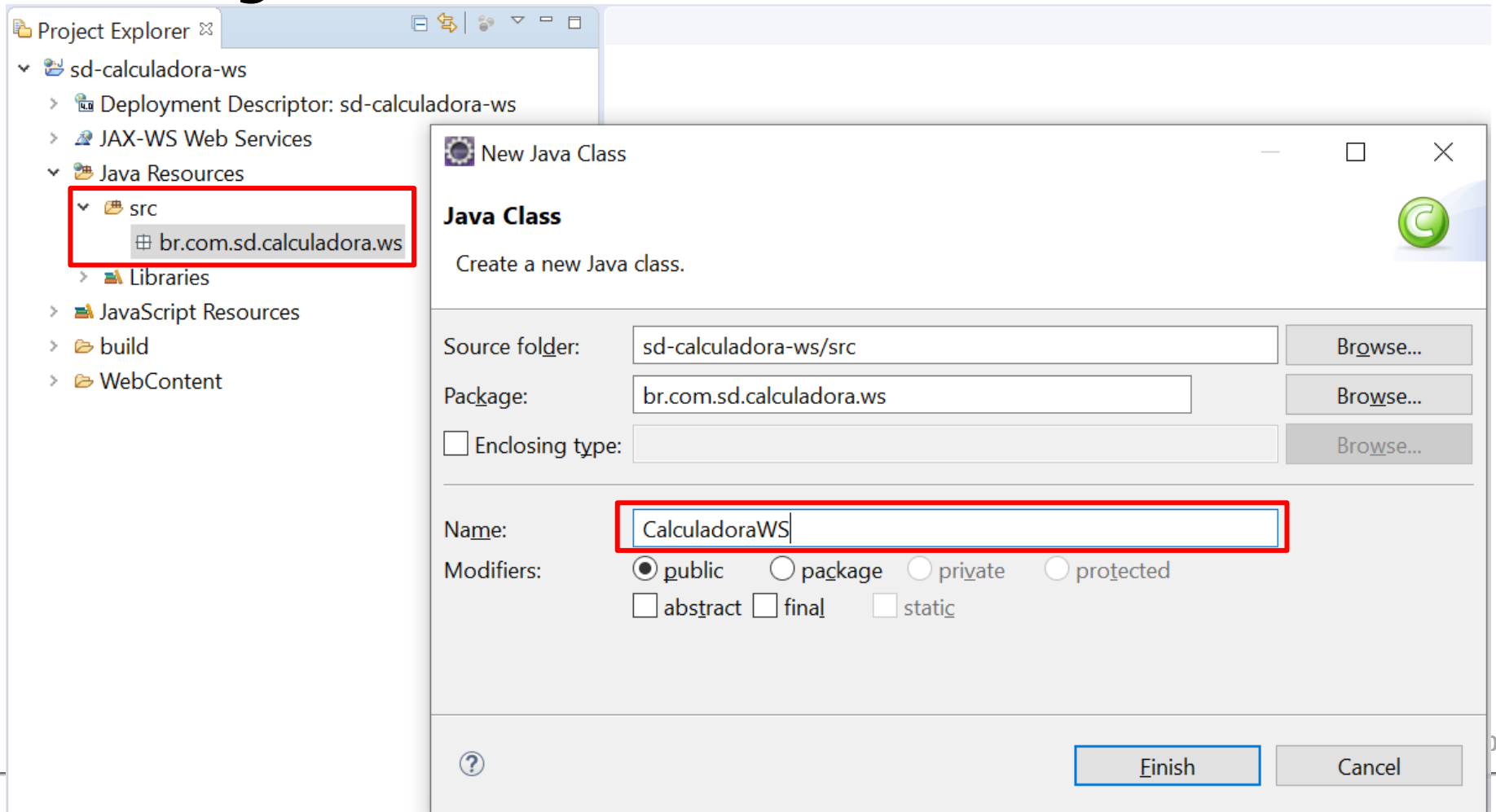
# Implementando as estruturas básicas do Apache CXF

- Primeiro criar um pacote.



# Implementando as estruturas básicas do Apache CXF

- Segundo criar uma classe.

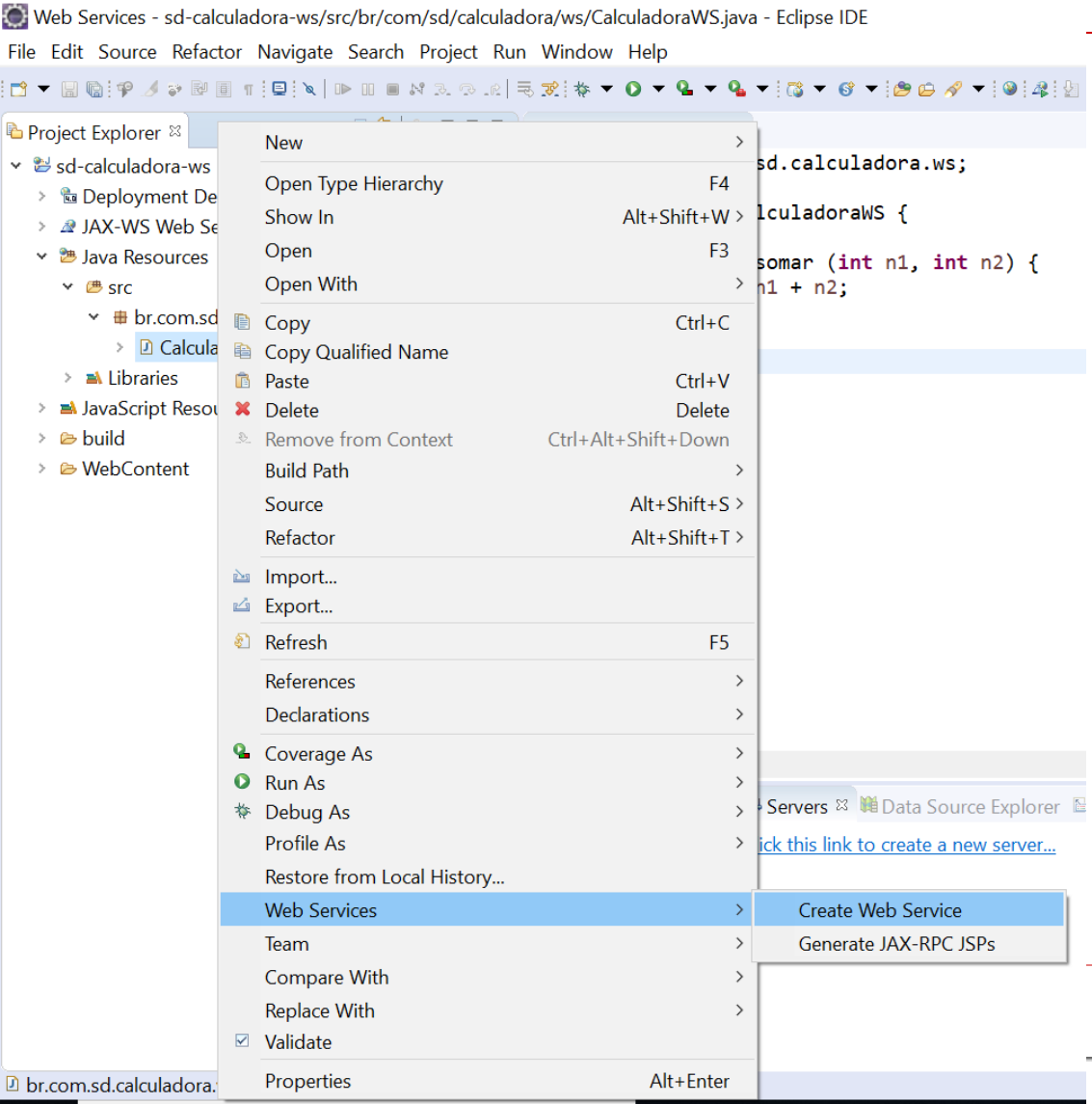


# Implementando as estruturas básicas do Apache CXF

- Criar um método da calculadora chamado “somar”

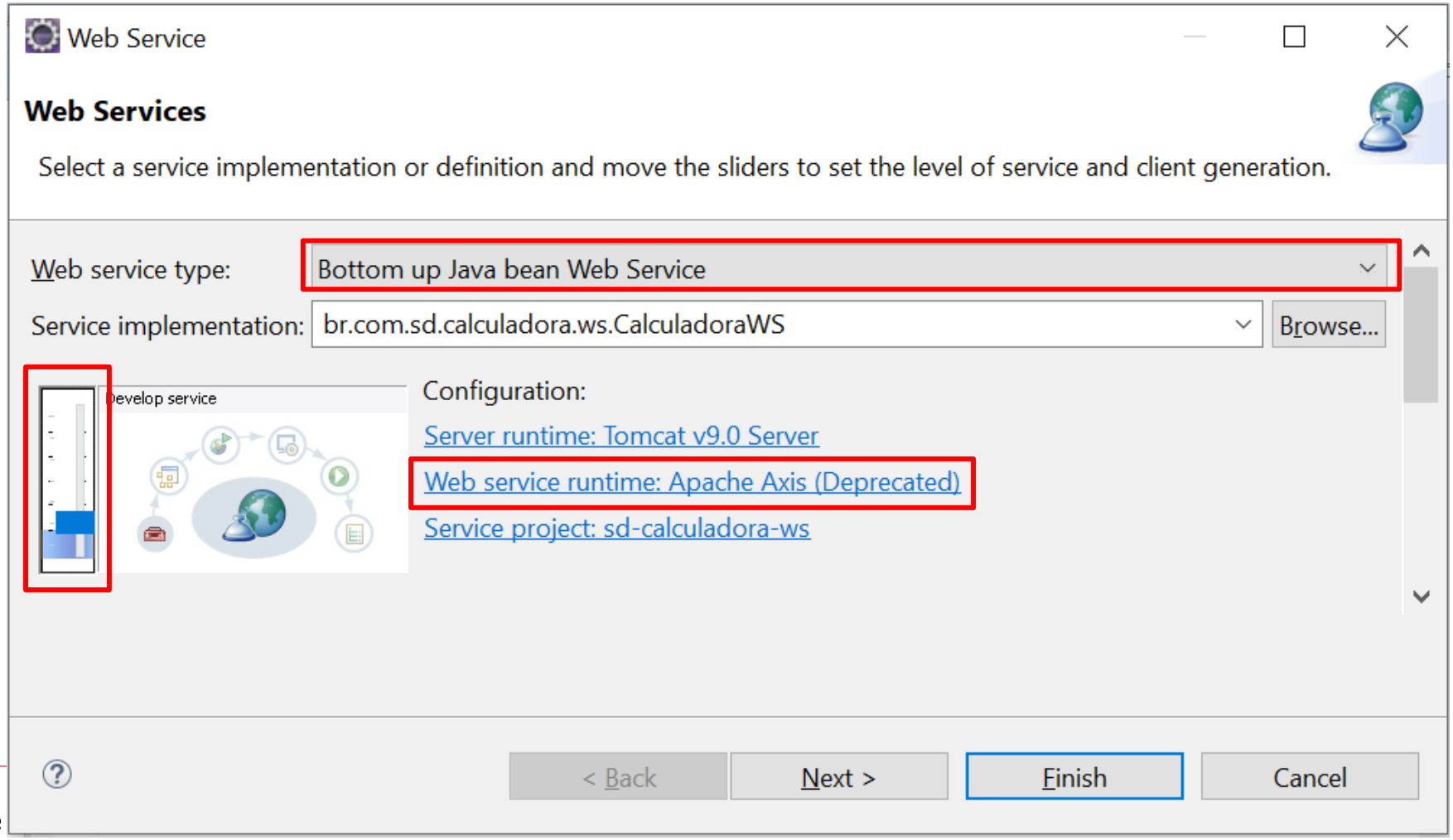
```
package br.com.sd.calculadora.ws;

public class CalculadoraWS {

    public int somar (int n1, int n2) {
        return n1 + n2;
    }
}
```

# Implementando as estruturas básicas do Apache CXF



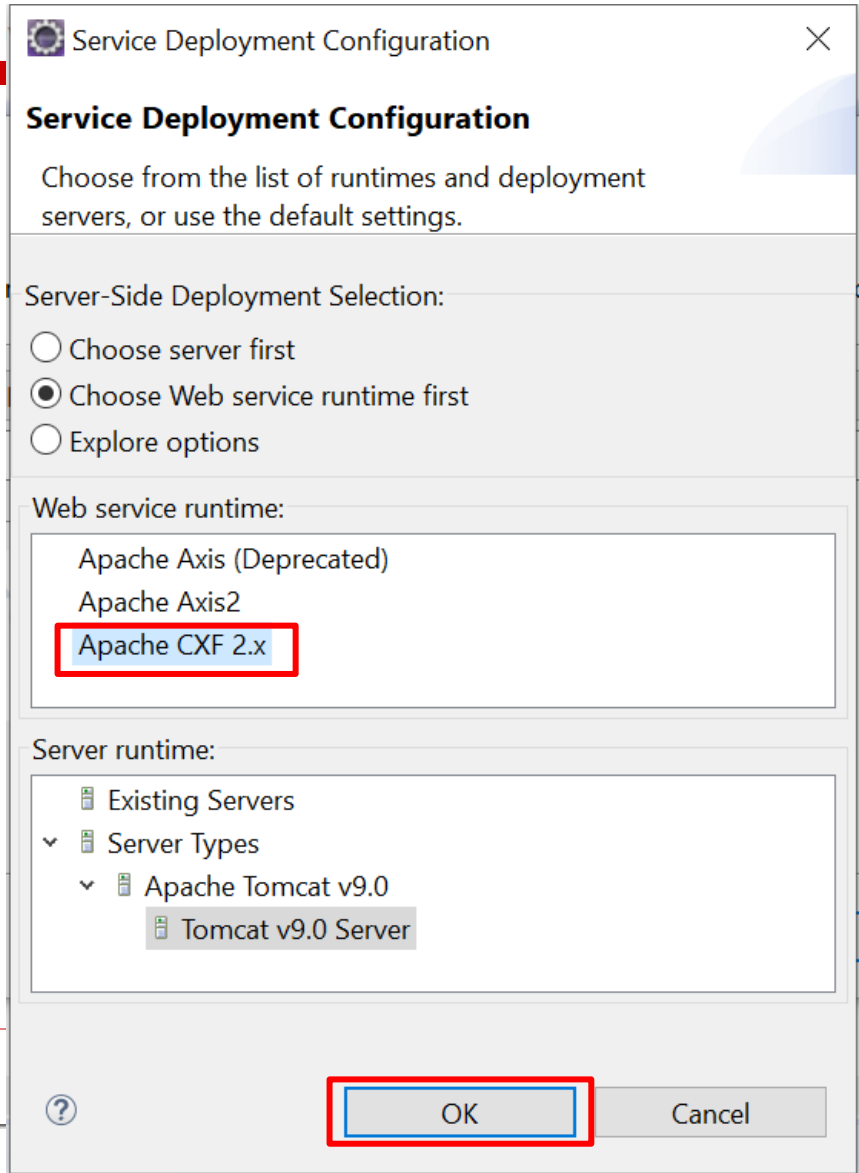

- A partir da classe criada, criar um web service.

# Implementando as estruturas básicas do Apache CXF



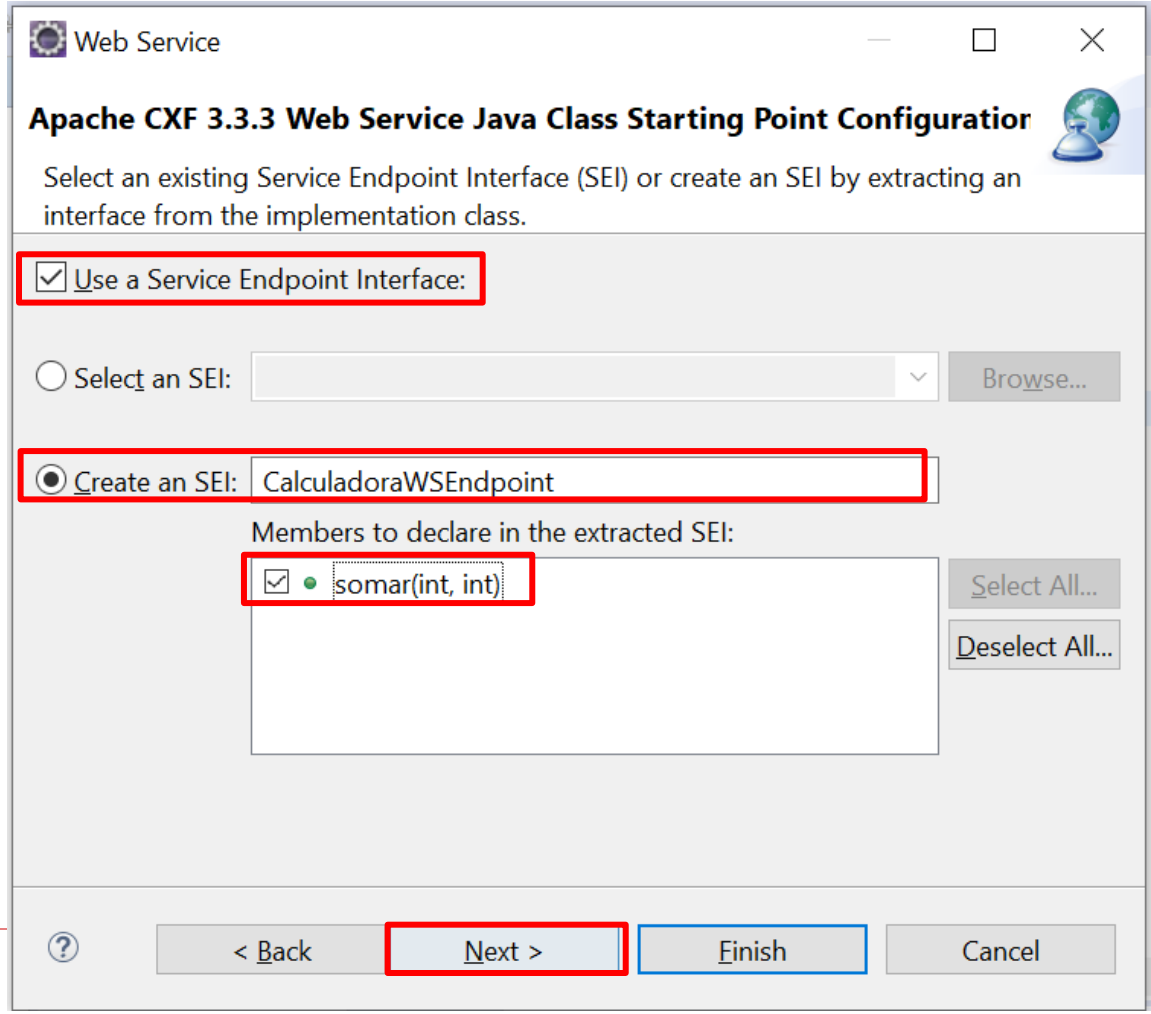
# Implementando as estruturas básicas do Apache CXF

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

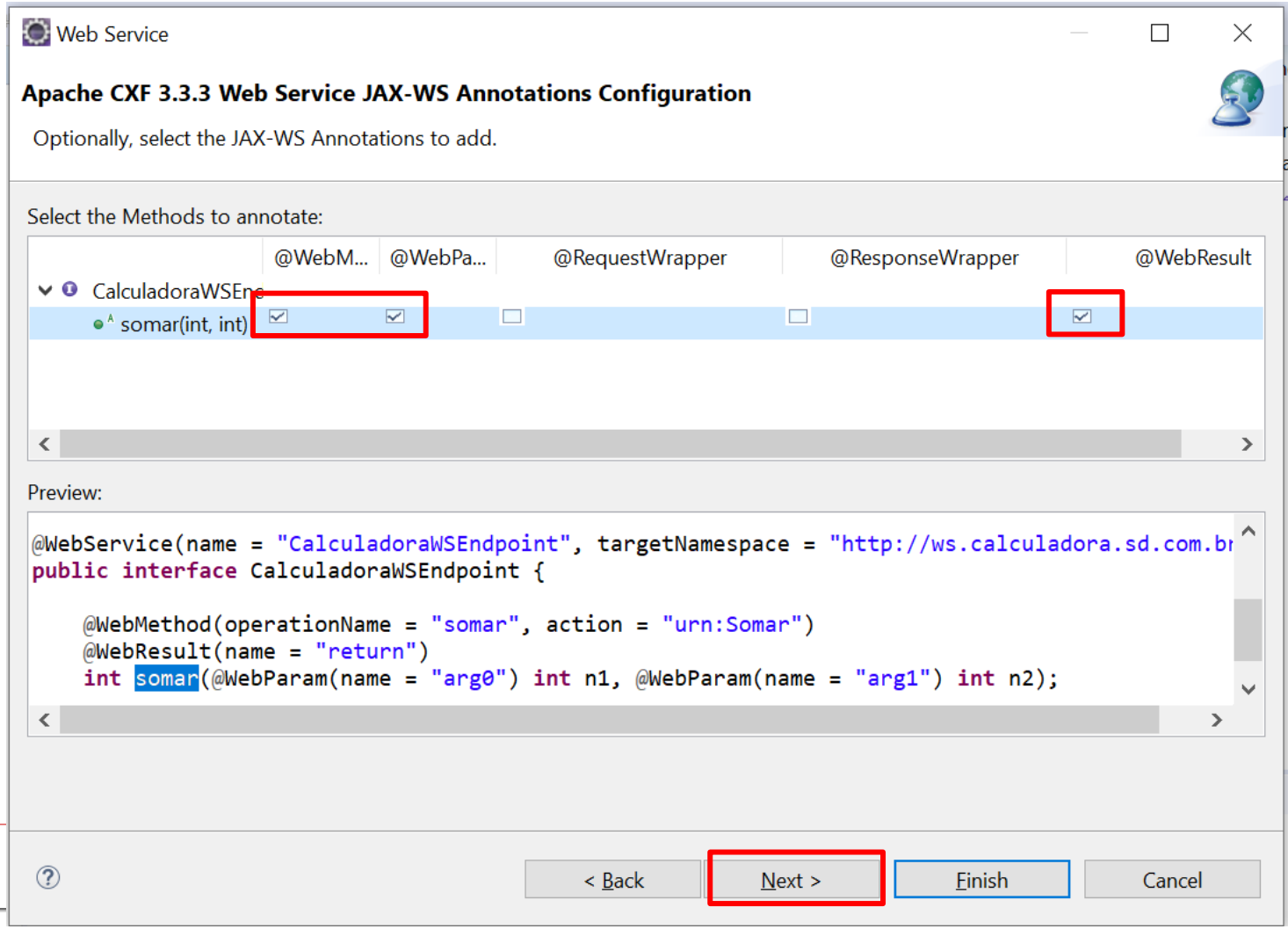


# Implementando as estruturas básicas do Apache CXF

- Criar uma interface de endpoint.

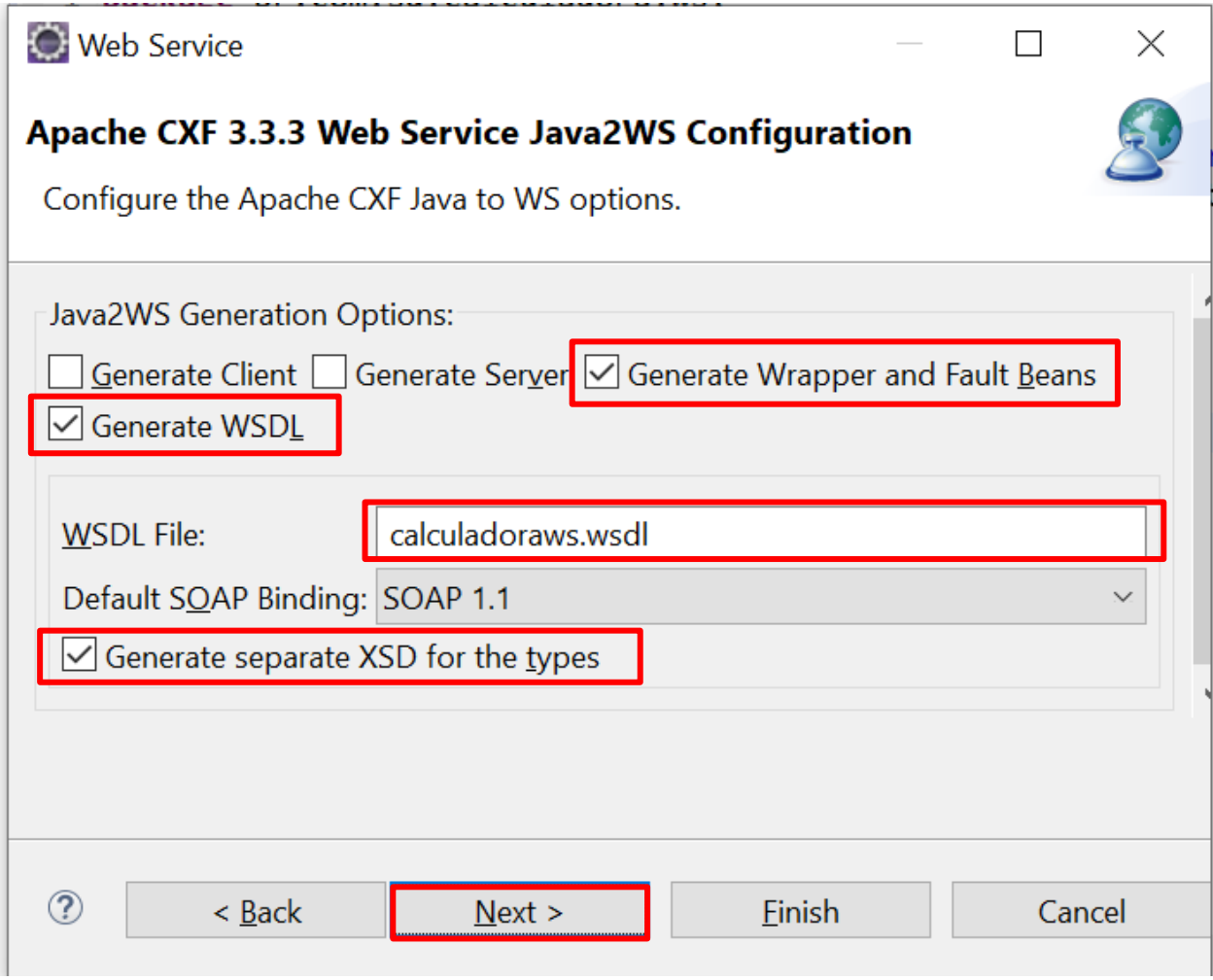


# Implementando as estruturas básicas do Apache CXF



# Implementando as estruturas básicas do Apache CXF

- Selecionar o checkbox “Generate Wrapper and Fault Beans” e “Generate WSDL”.



# Implementando as estruturas básicas do Apache CXF

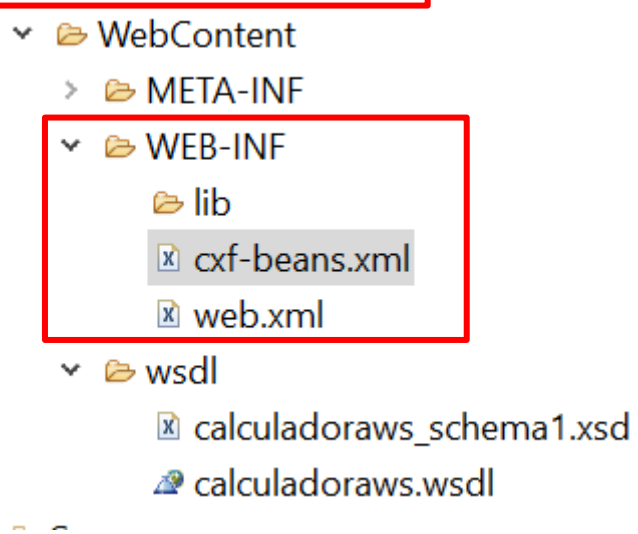
- Na janela “Web Services Publication”, não marcar nada e clicar em “Finish”.

# Implementando as estruturas básicas do Apache CXF

- Comentar o código da linha “7” dentro arquivo “cxf-beans.xml”

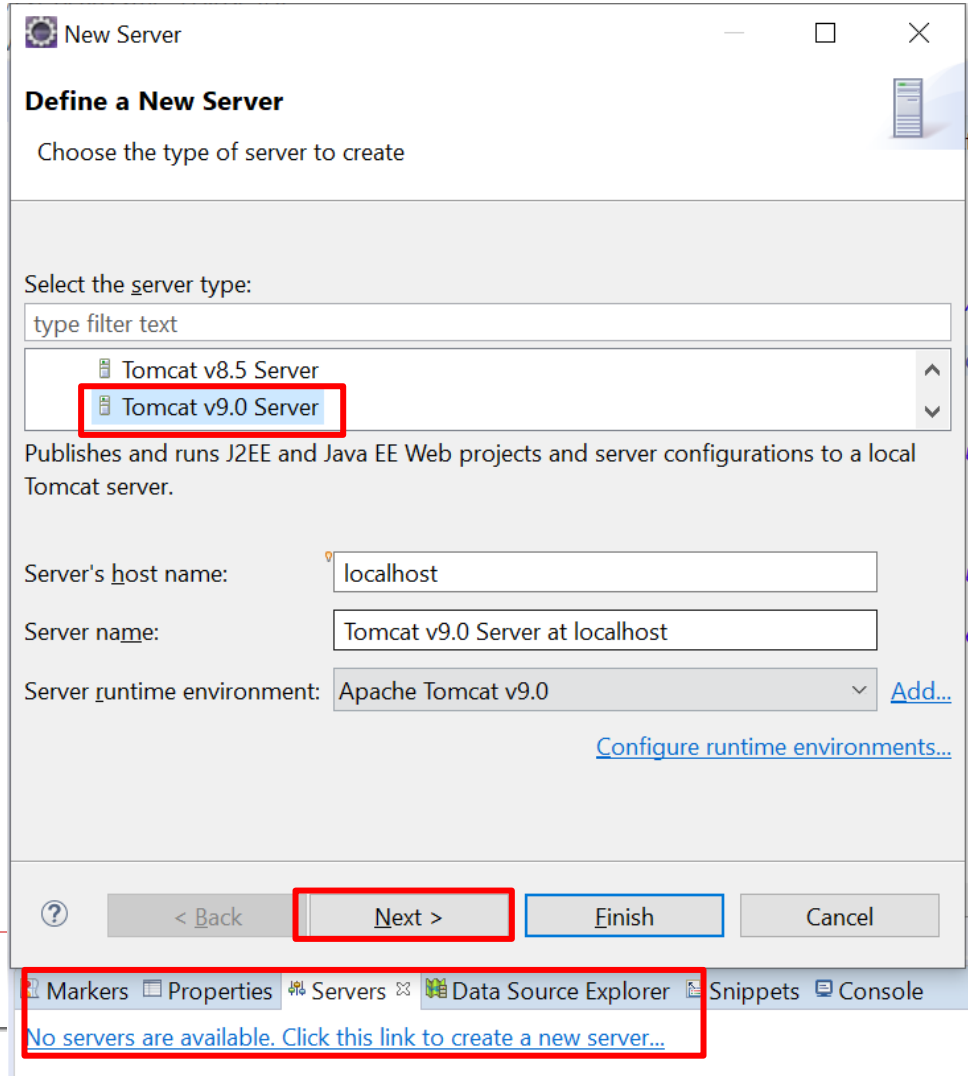
CalculadoraWS.java | CalculadoraWSEndpoint.java | calculadoraws.wsdl | cxf-beans.xml

```
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.springframewor
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="calcu
11         implementor="br.com.sd.calculadora.ws.CalculadoraWS"
12         wsdlLocation="wsdl/calculadoraws.wsdl"
13         endpointName="tns:CalculadoraWSPort"
14         serviceName="tns:CalculadoraWSService" address="/Calcu
15         <jaxws:features>
16             <bean class="org.apache.cxf.feature.LoggingFeature
17         </jaxws:features>
18     </jaxws:endpoint>
19 </beans>
20
```



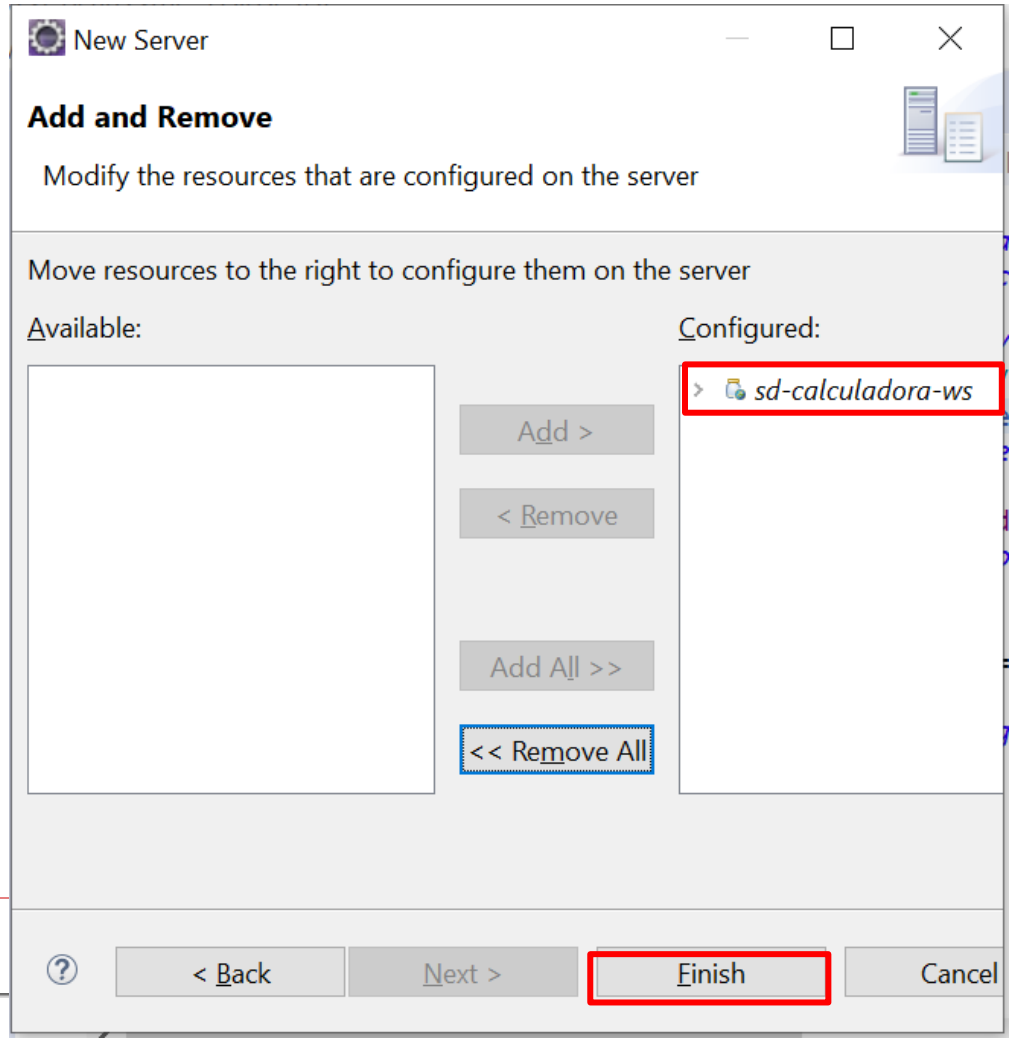
# Implementando as estruturas básicas do Apache CXF

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



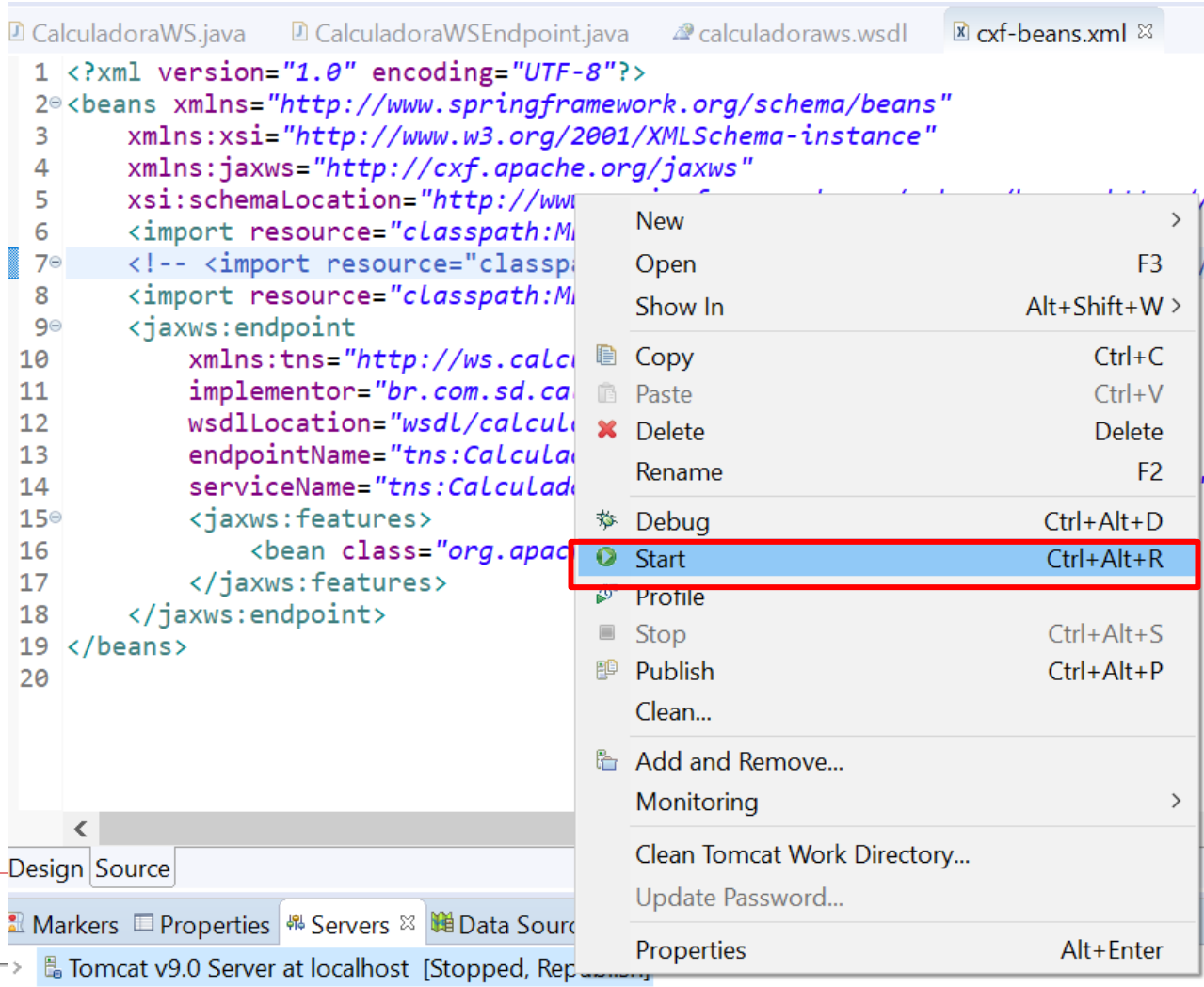
# Implementando as estruturas básicas do Apache CXF

- Jogar a "sd-calculadora-ws" para o meu servidor.



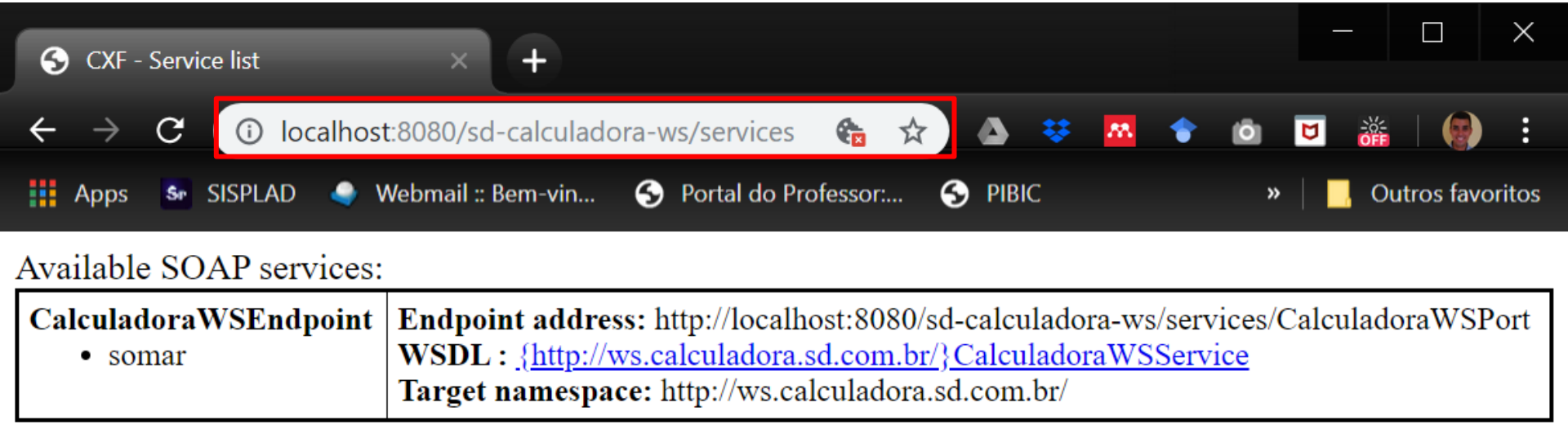
# Implementando as estruturas básicas do Apache CXF

- Iniciar o sevidor.

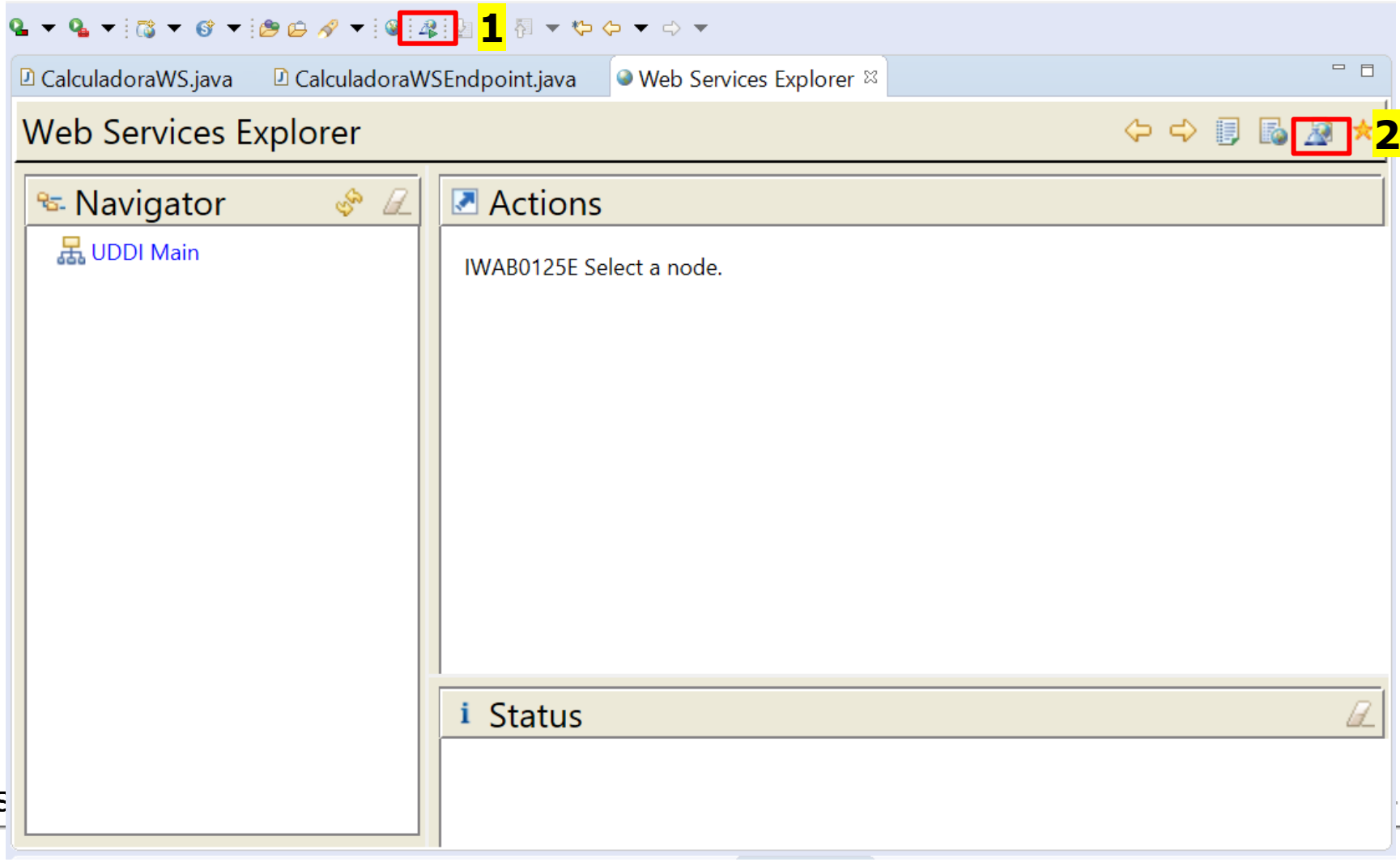


# Implementando as estruturas básicas do Apache CXF

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

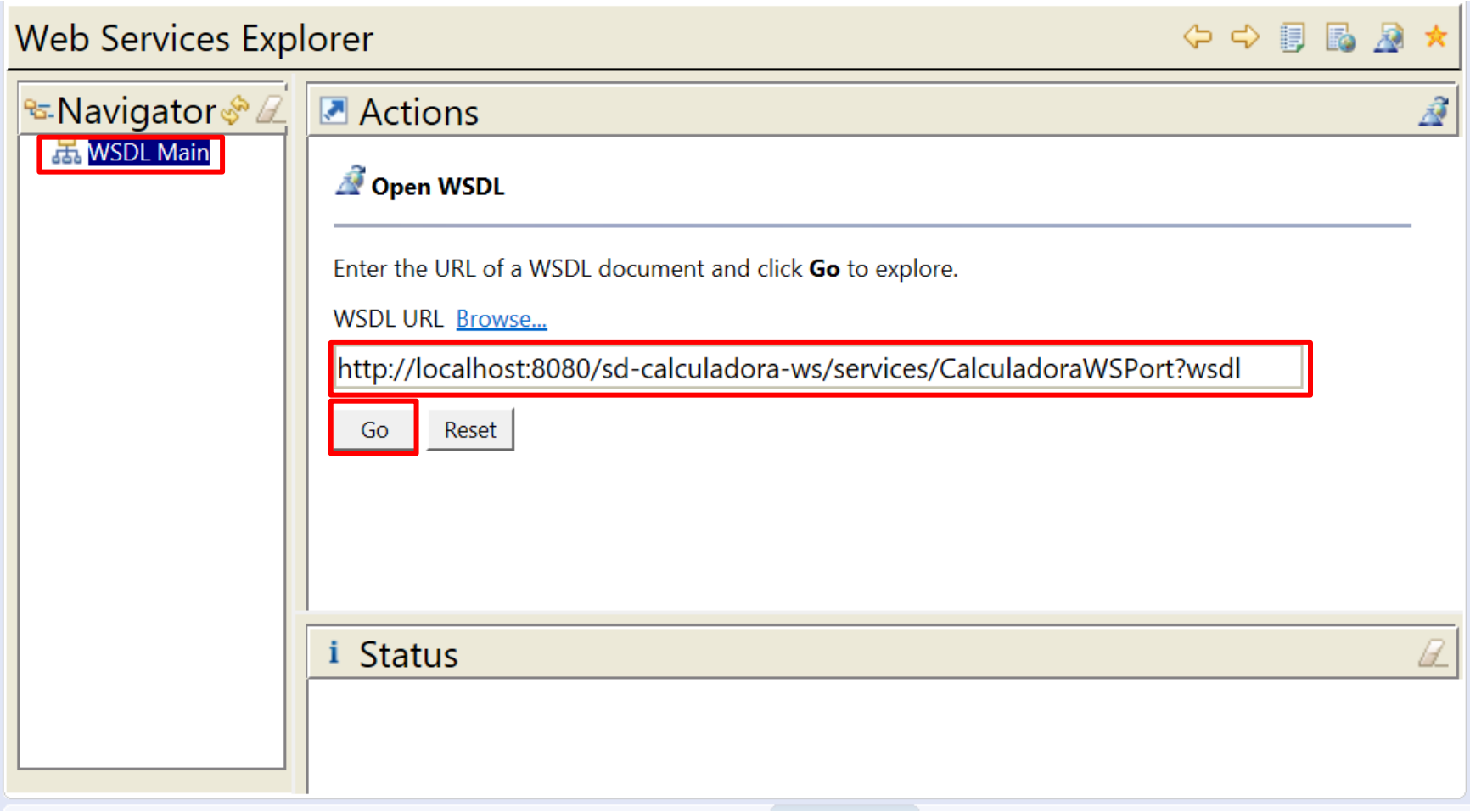


# Implementando as estruturas básicas do Apache CXF



# Implementando as estruturas básicas do Apache CXF

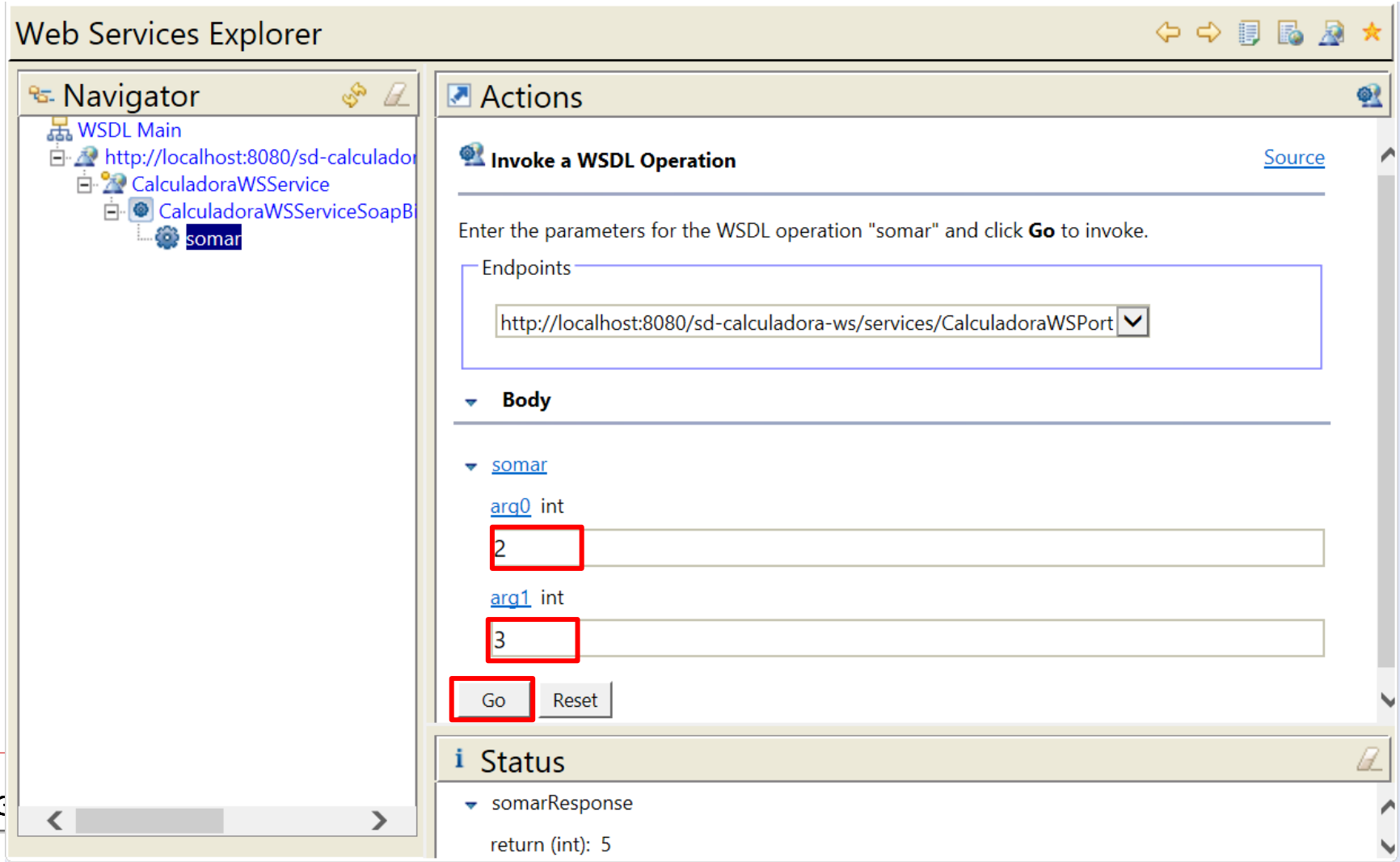
- Incluir a URL do serviço, em clicar em "GO"



# Implementando as estruturas básicas do Apache CXF

- Clicar em “SOMAR” da Aba Operations.

# Implementando as estruturas básicas do Apache CXF



# 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.

CalculadoraWS.java | calculadoraws_schema1.xsd

```
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.

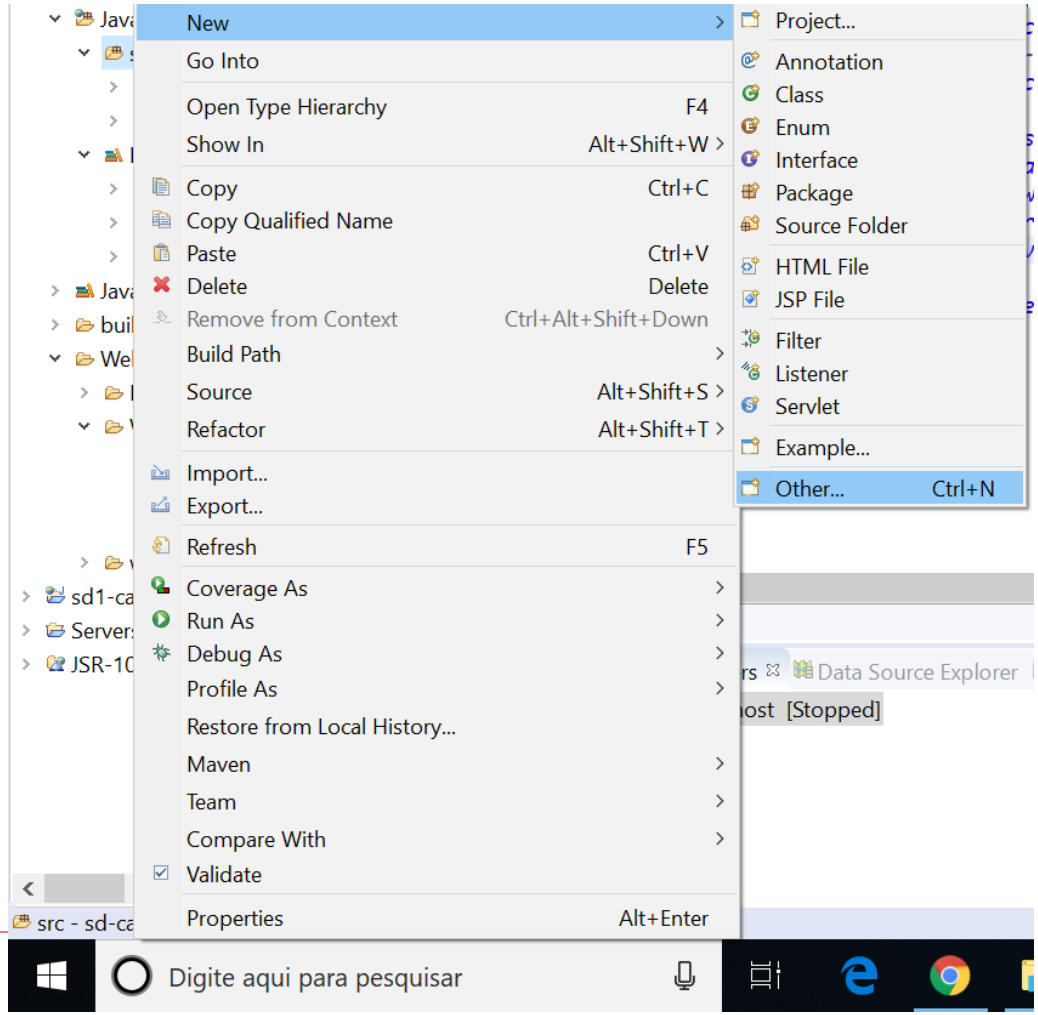
CalculadoraWS.java | calculadoraws_schema1.xsd | CalculadoraWSEndpoint.java | calculadoraws.wsdl

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <wsdl:definitions name="CalculadoraWSService" targetNamespace="http://ws.calculadora.sd.com.br/" xmlns:wsdl="http://schem
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"/>
```

Design | Source

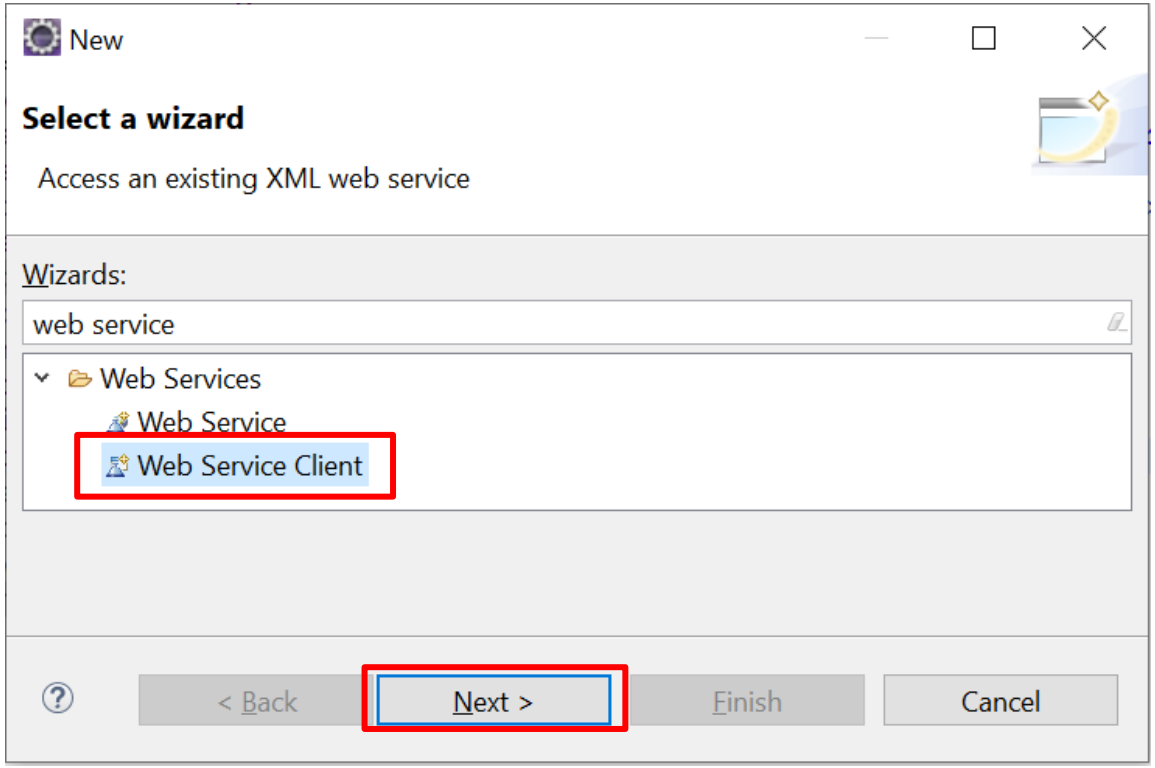
# Gerando um cliente para um serviço SOAP com Apache CXF

- Clique com o botão direto na pasta “src”, selecione “New”, em seguida “Other”.



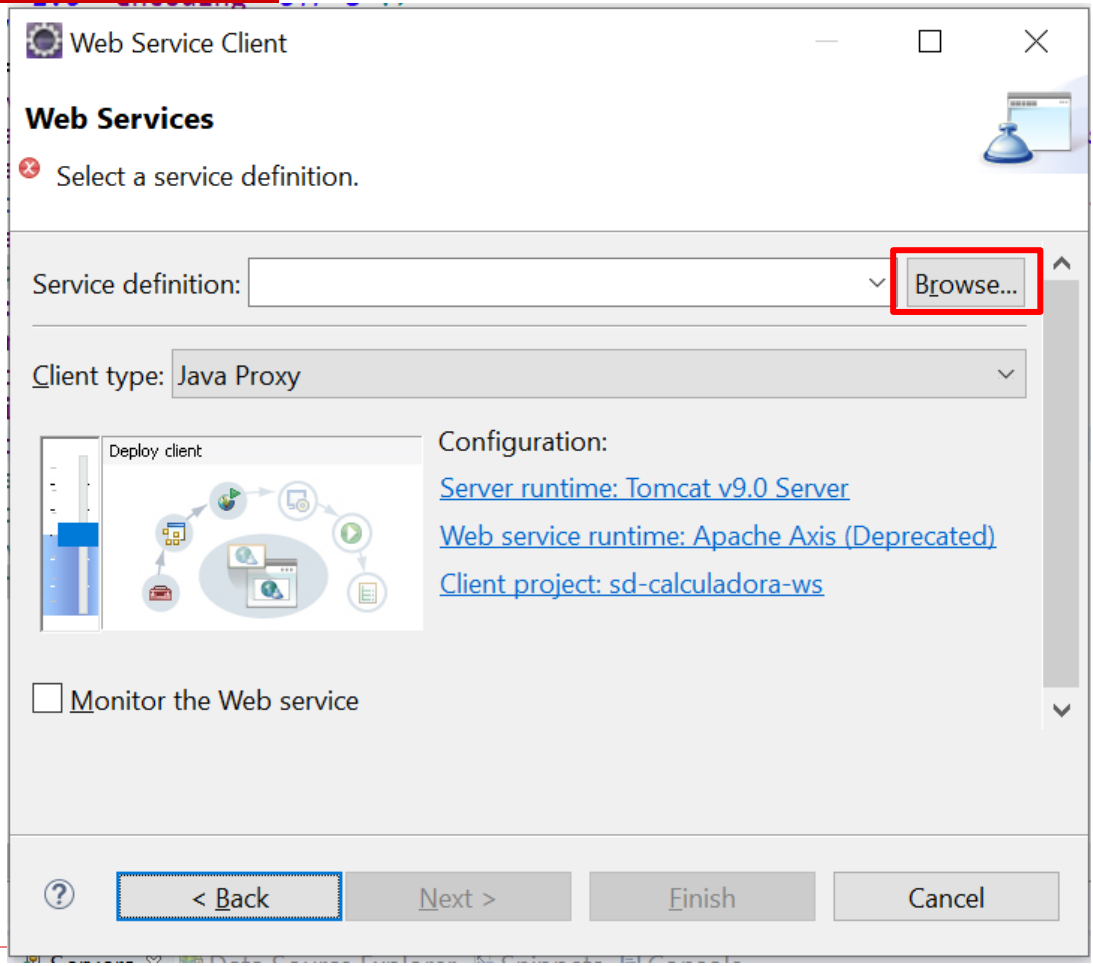
# Gerando um cliente para um serviço SOAP com Apache CXF

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

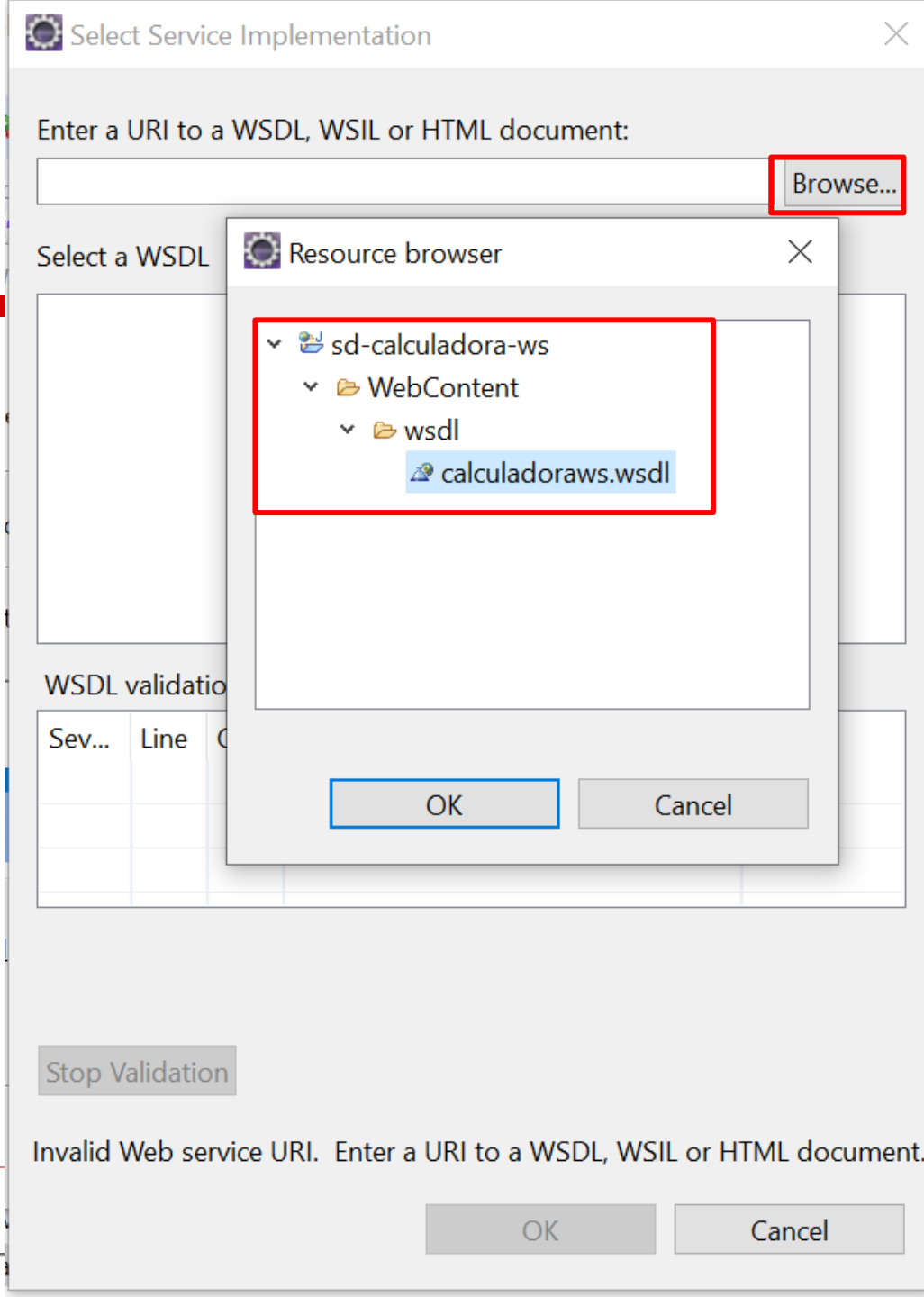


# Gerando um cliente para um serviço SOAP com Apache CXF

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

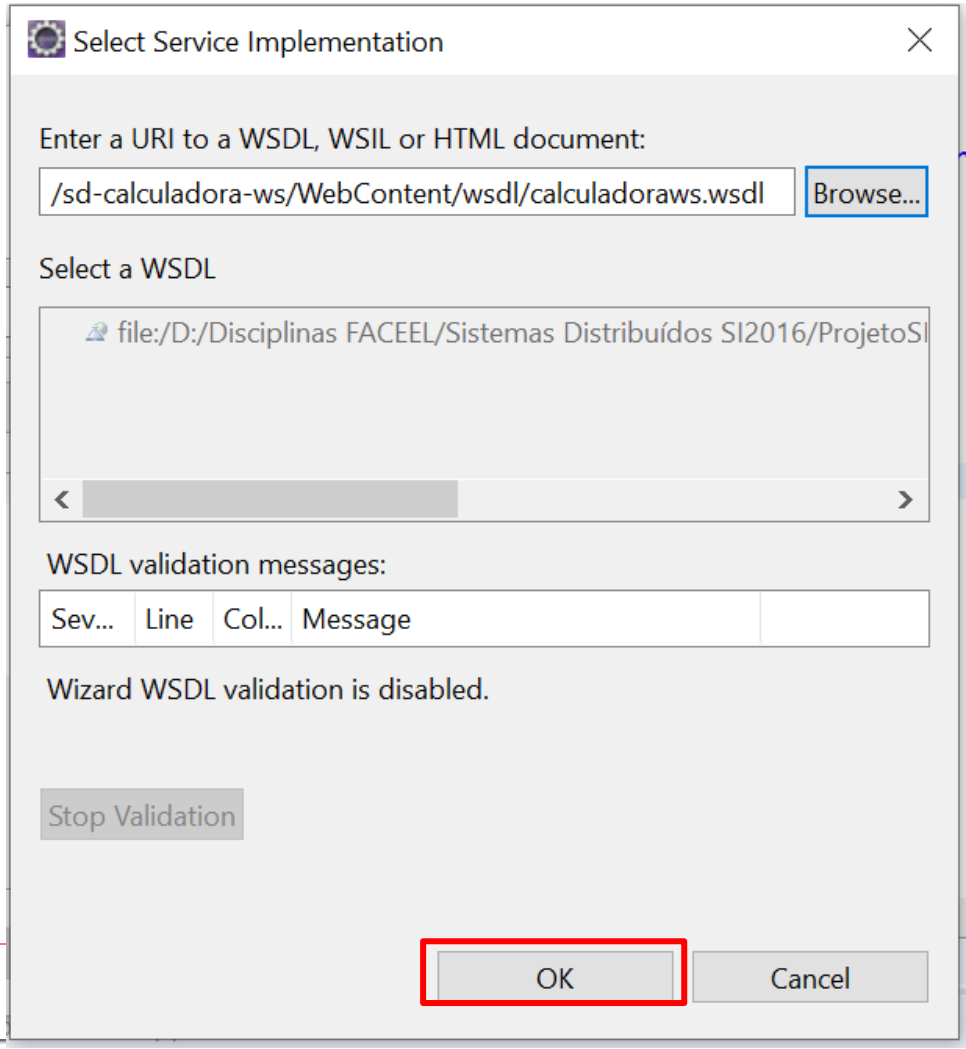


- Selecione o arquivo “calculadora.wsdl” e clique em “OK”



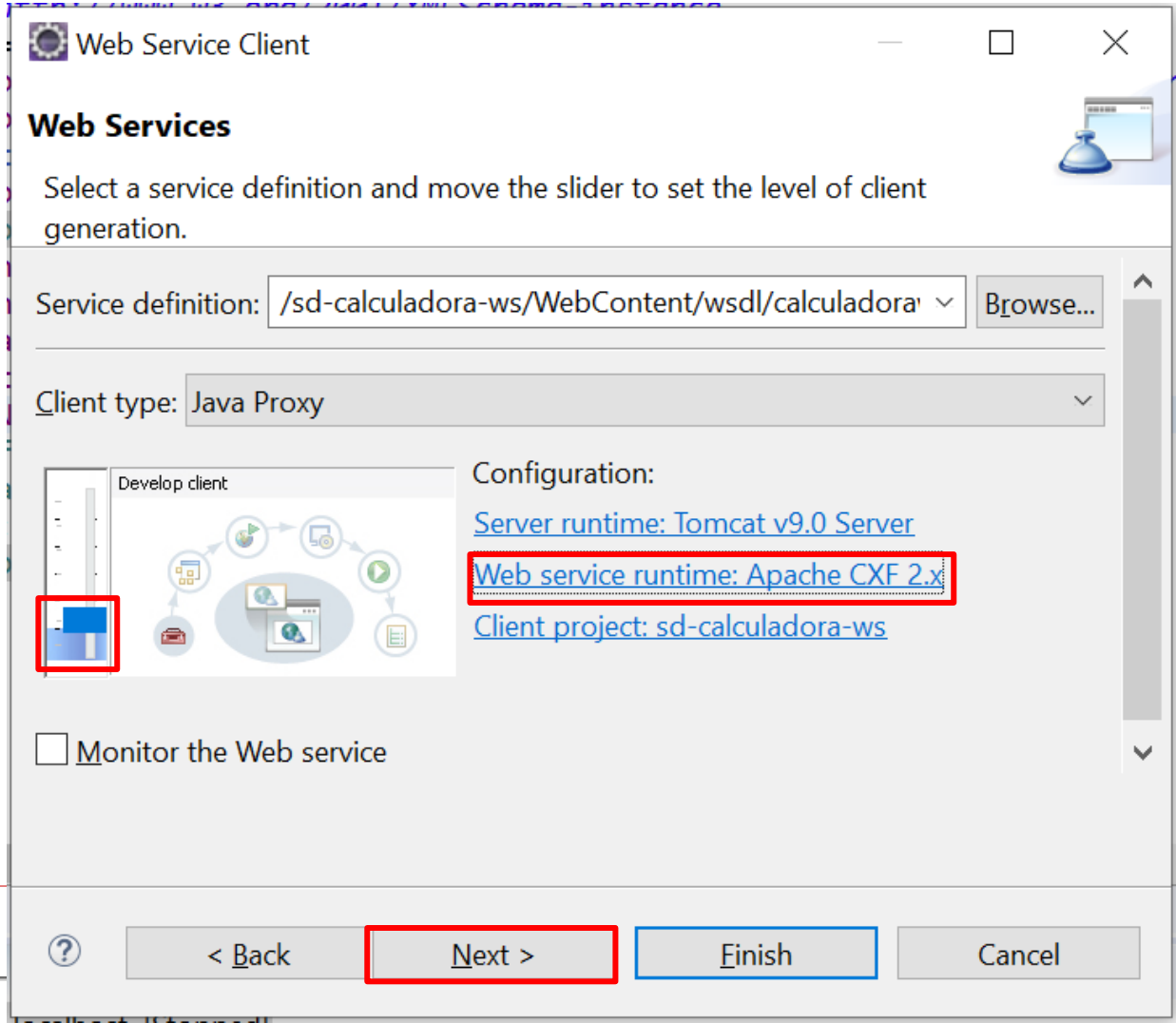
# Gerando um cliente para um serviço SOAP com Apache CXF

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



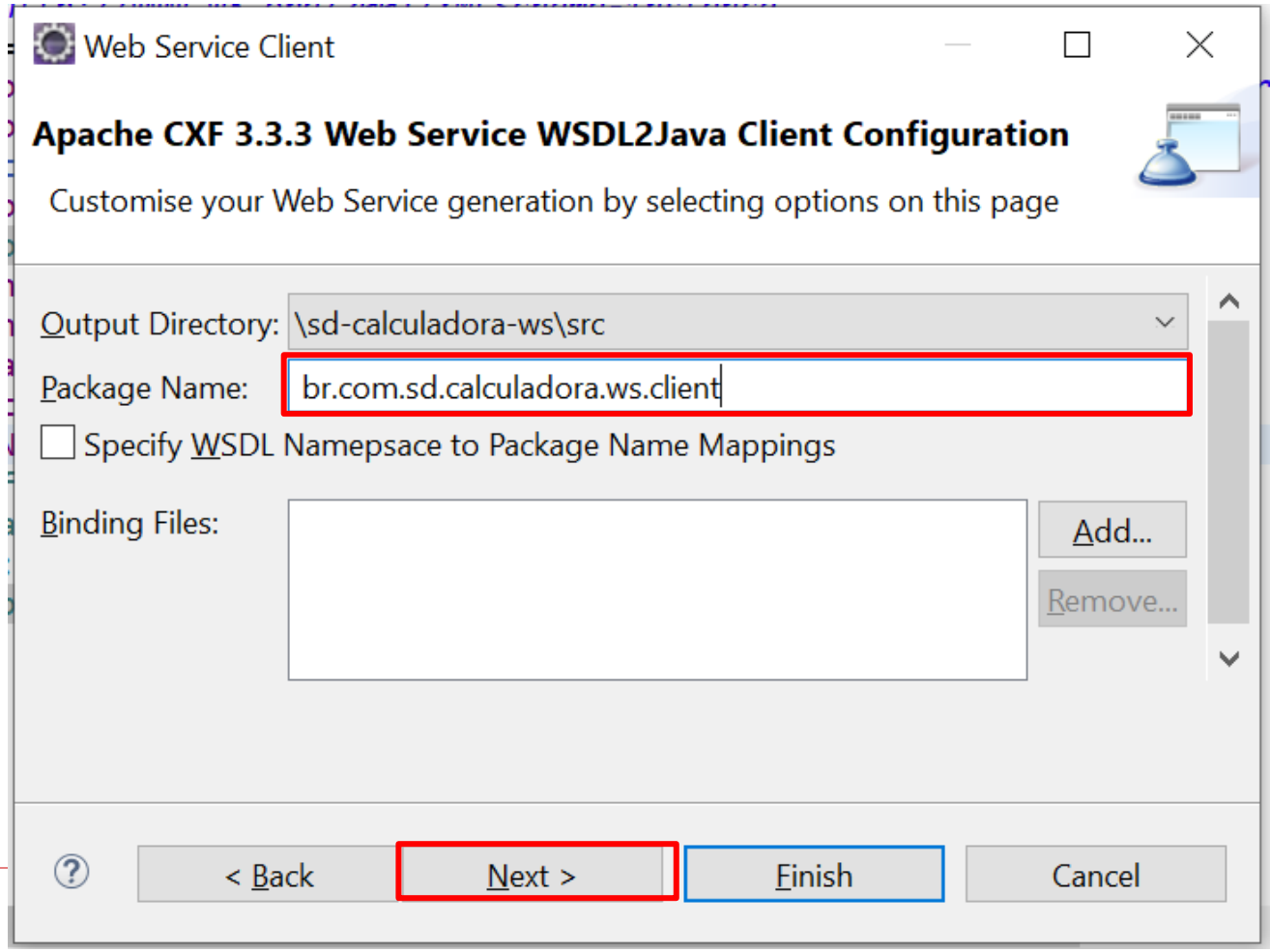
# Gerando um cliente para um serviço SOAP com Apache CXF

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



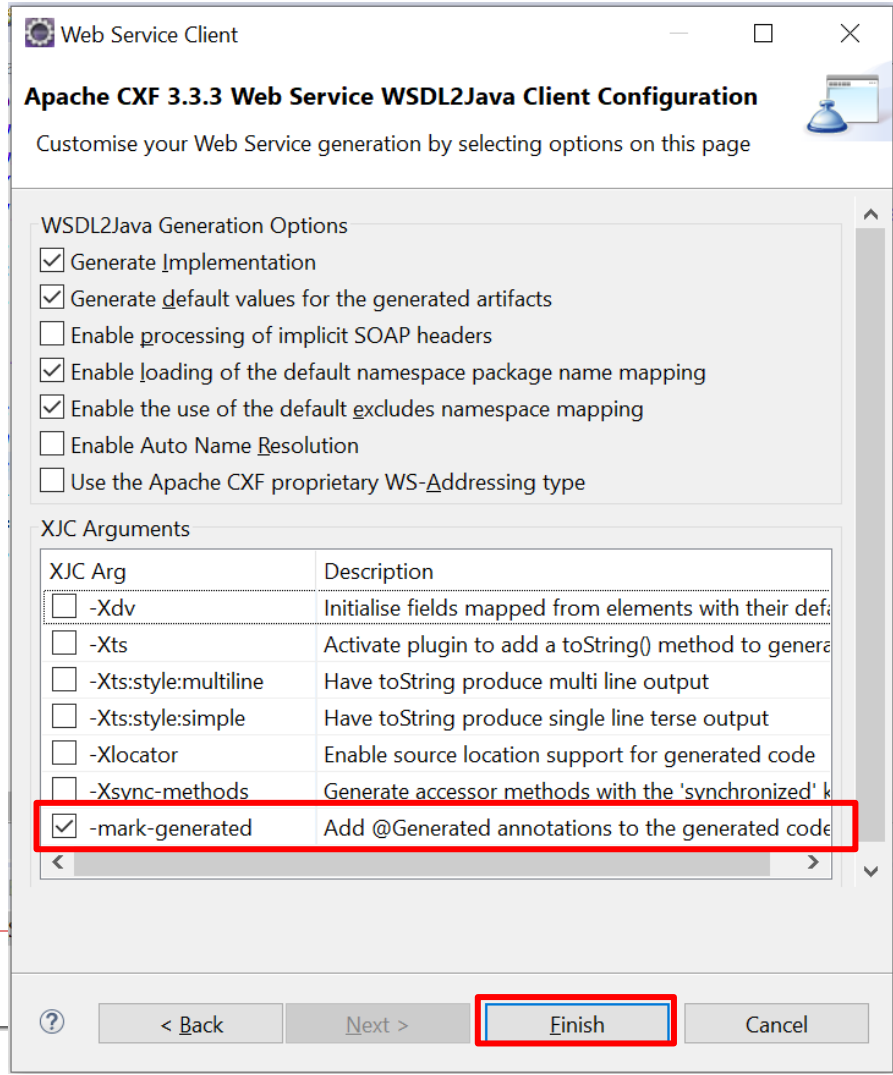
# Gerando um cliente para um serviço SOAP com Apache CXF

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



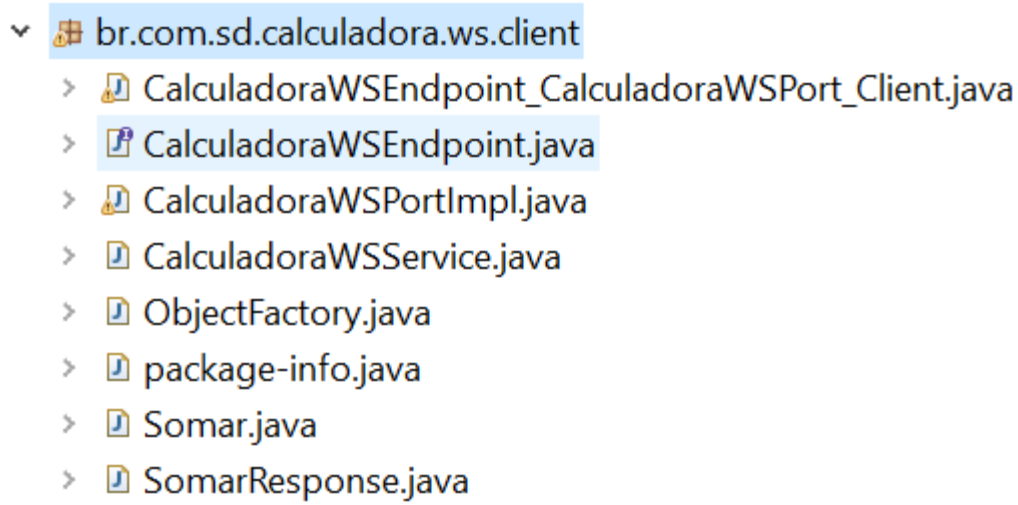
# Gerando um cliente para um serviço SOAP com Apache CXF

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



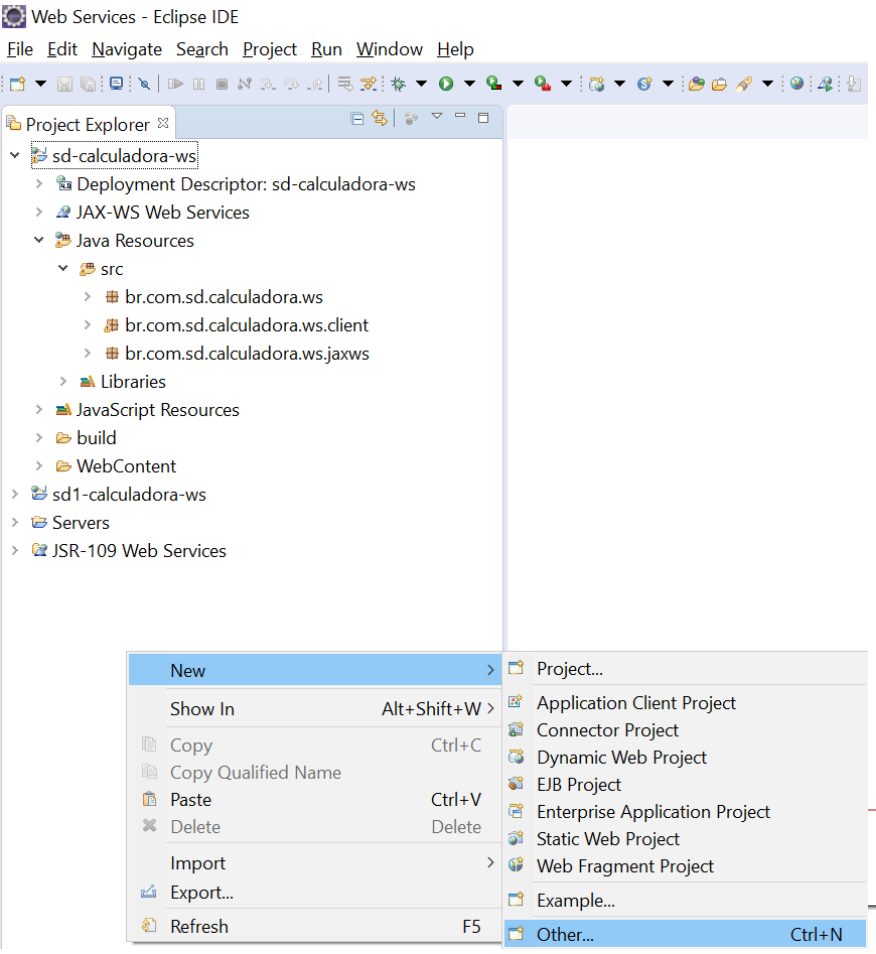
# Gerando um cliente para um serviço SOAP com Apache CXF

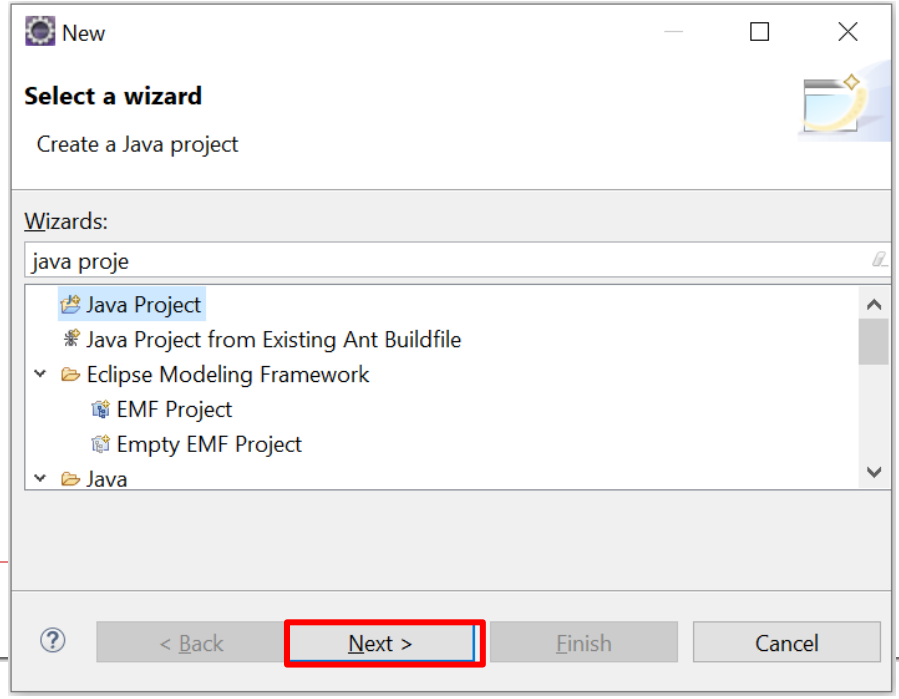
- Todas as classes listadas no pacote, são necessárias para acessar o web service.



# Gerando um cliente para um serviço SOAP com Apache CXF

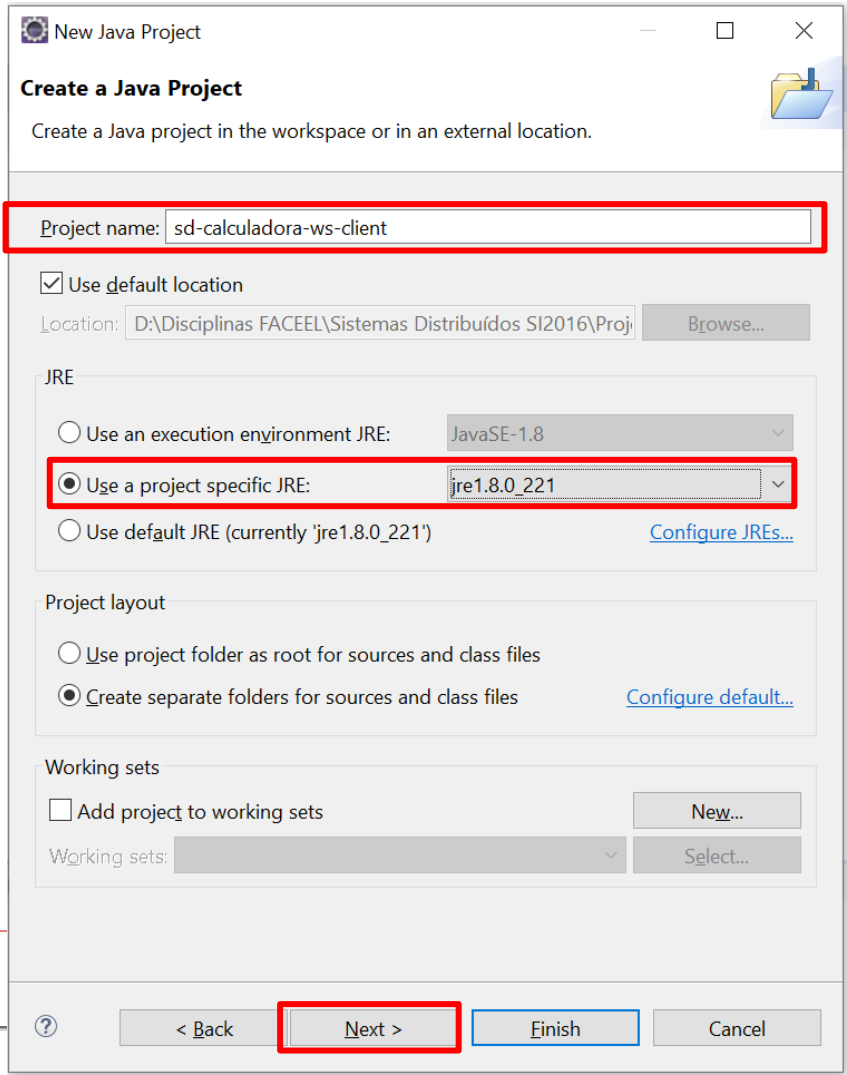
- Criar um novo projeto java.



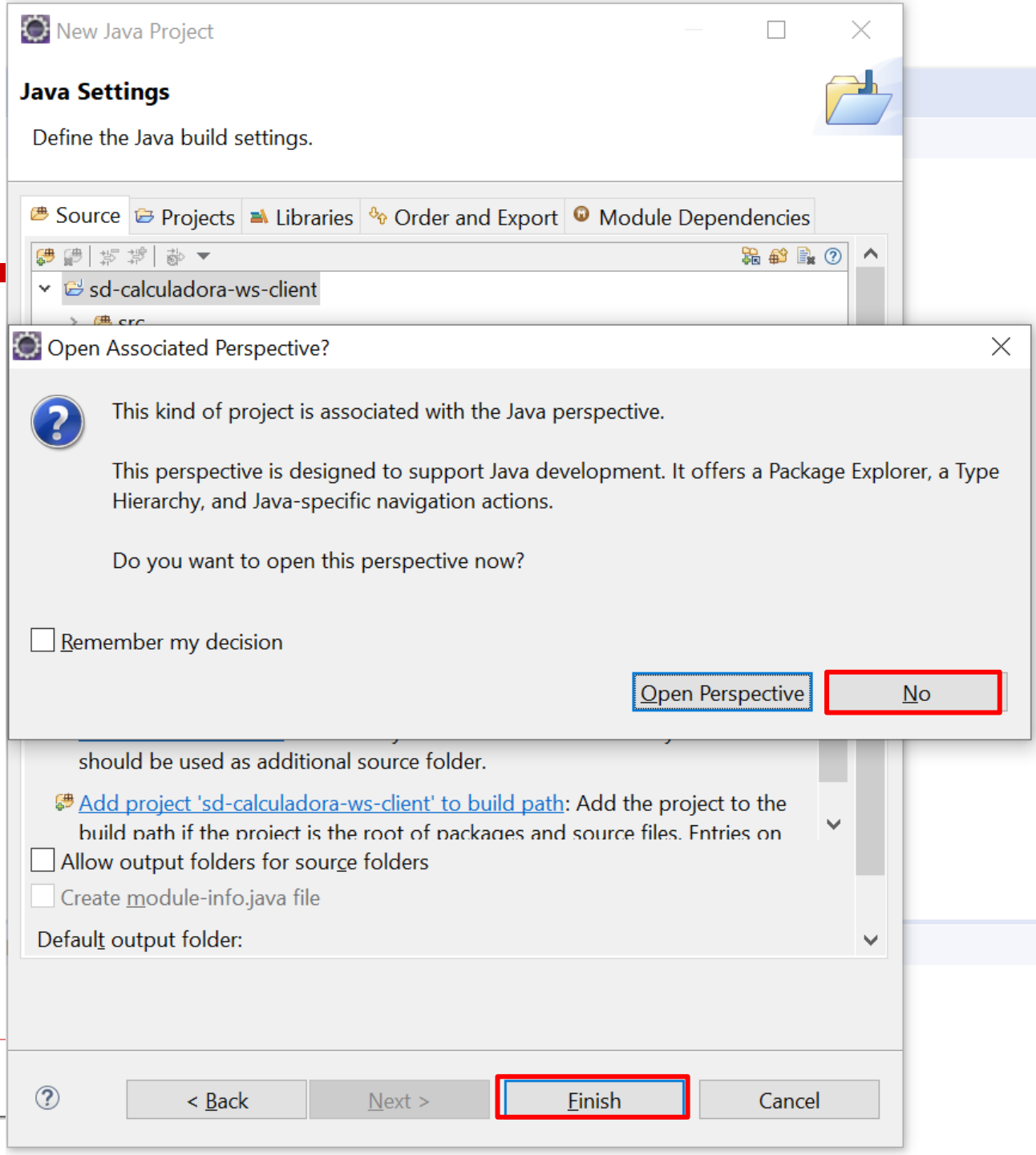



# Gerando um cliente para um serviço SOAP com Apache CXF

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

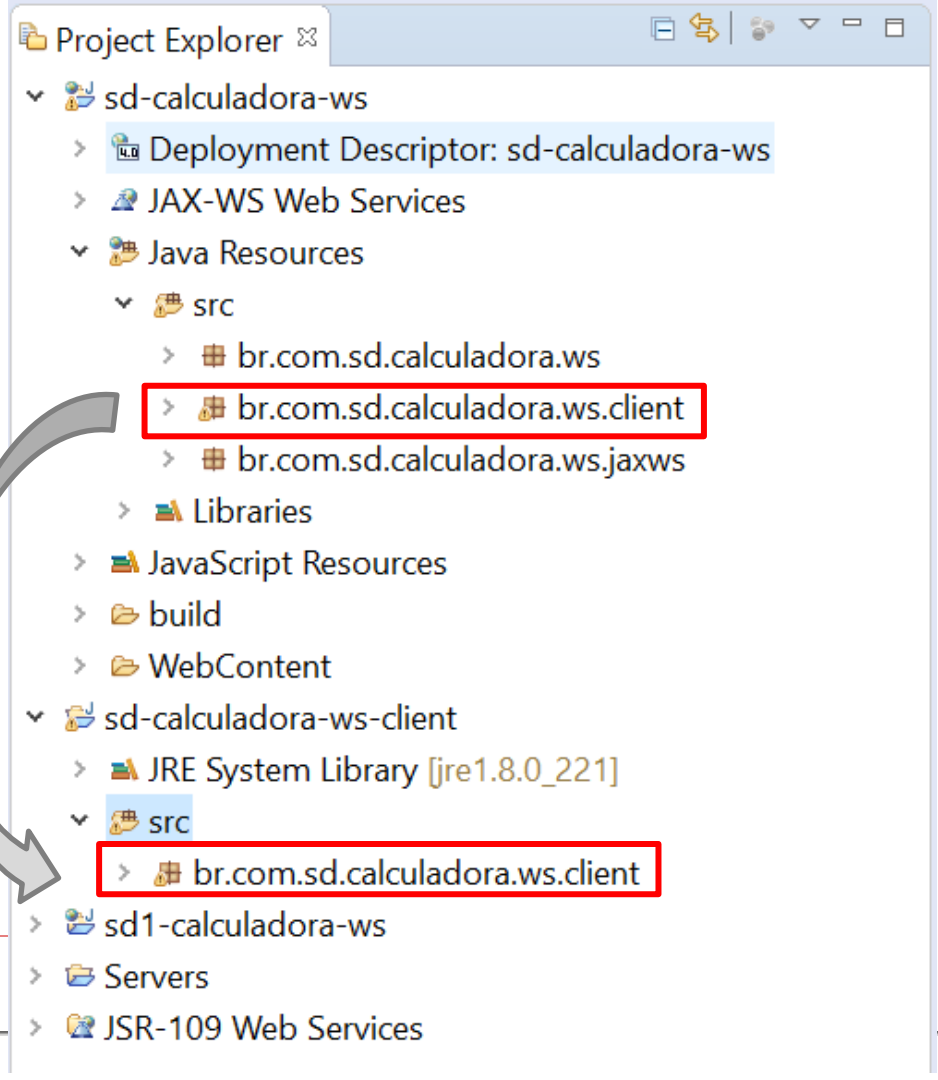


- Clique em "Finish" e em seguida, clique em "No" para manter a perspectiva Java.



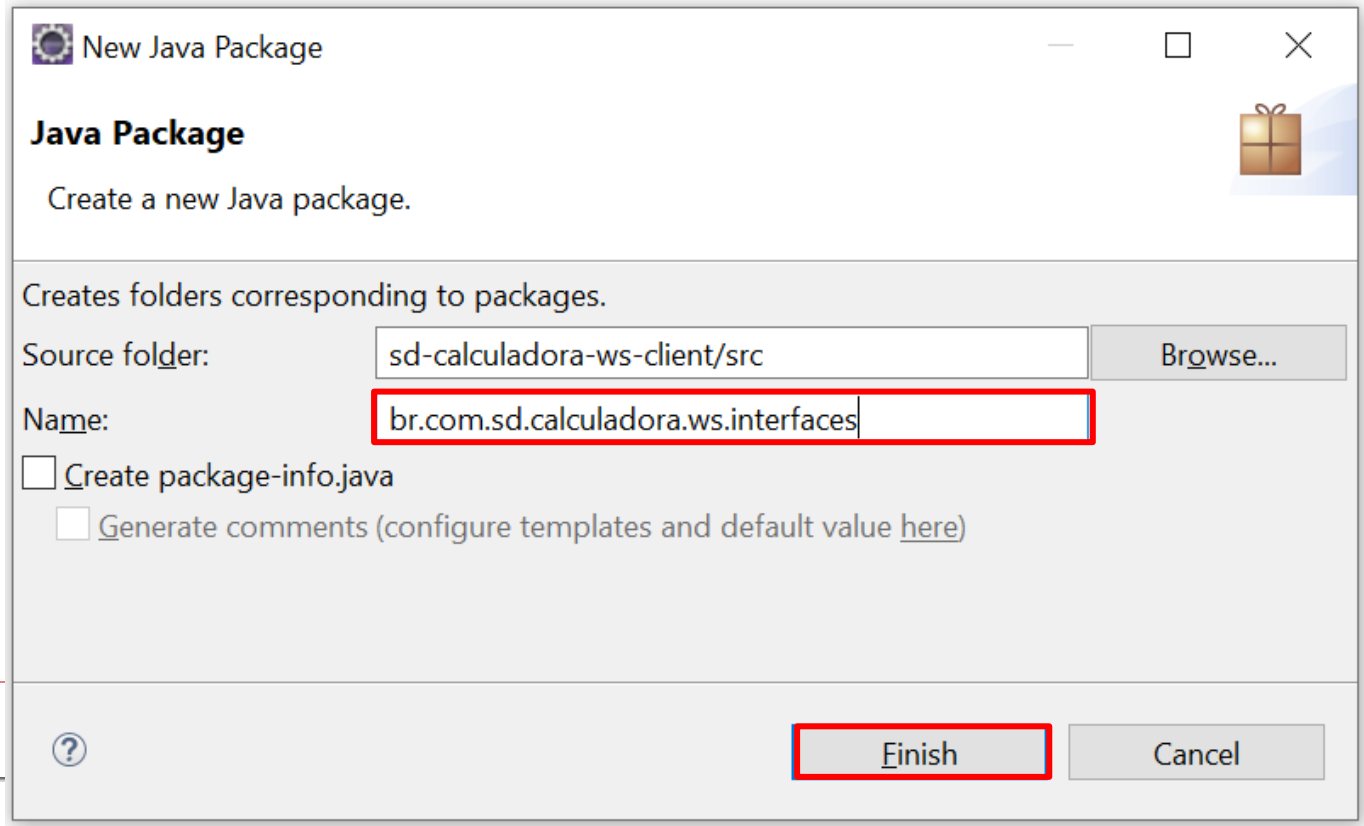
# Gerando um cliente para um serviço SOAP com Apache CXF

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

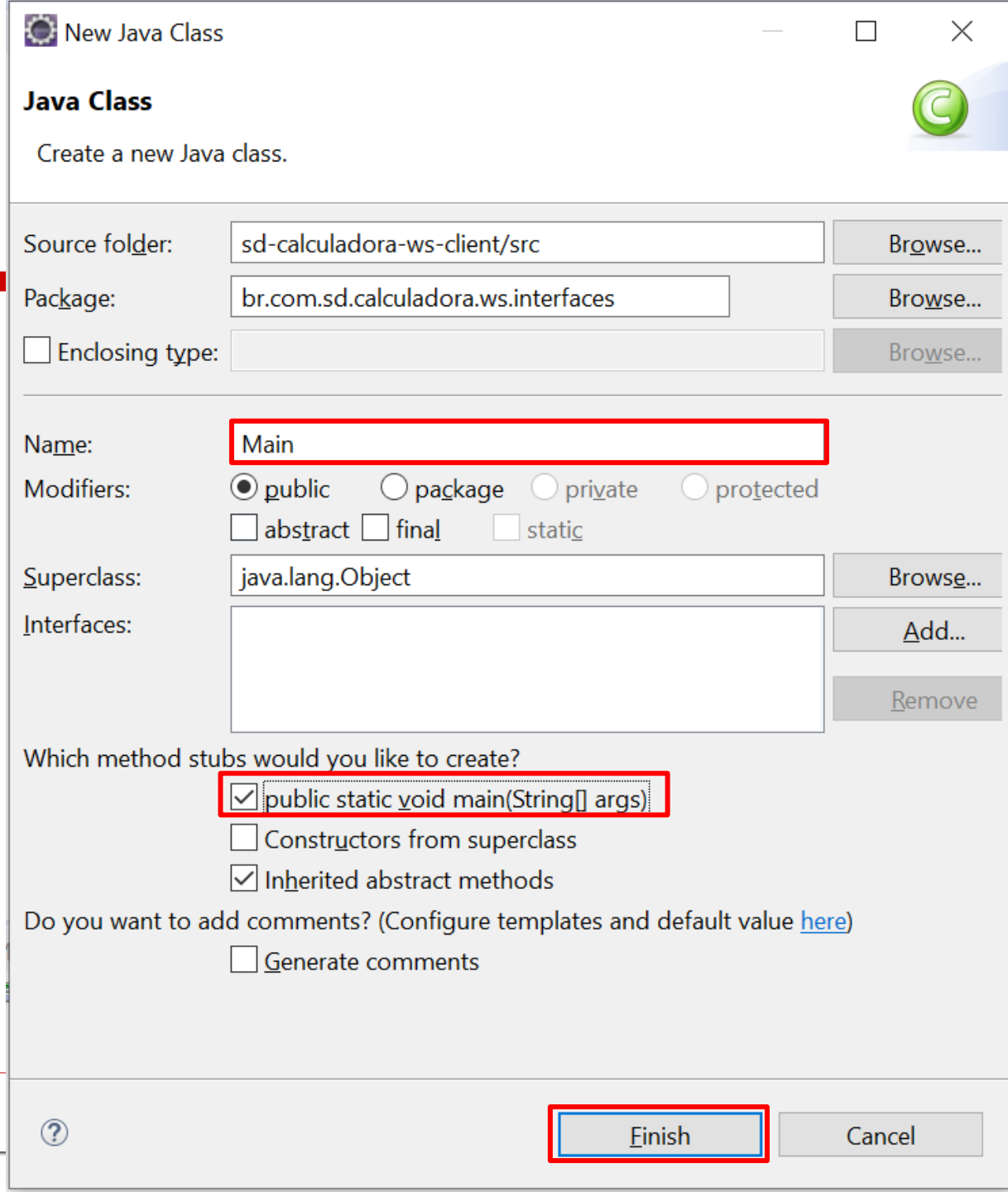


# Gerando um cliente para um serviço SOAP com Apache CXF

- A partir do projeto "sd-calculadora-ws-client", crie um novo pacote.



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



# 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
```