

**CENTRO ESTADUAL DE EDUCAÇÃO  
TECNOLÓGICA PAULA SOUZA**

**FACULDADE DE TECNOLOGIA DE LINS PROF. ANTONIO SEABRA  
CURSO SUPERIOR DE TECNOLOGIA EM SISTEMAS PARA  
INTERNET**

**RITA DE CÁSSIA SILVEIRA SALVAJOLI LEITE**

**CONSUMINDO UM WEB SERVICE:  
Calculadora**

**LINS/SP  
2º SEMESTRE/2021**

# **CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA SOUZA**

**FACULDADE DE TECNOLOGIA DE LINS PROF. ANTONIO  
SEABRA  
CURSO SUPERIOR DE TECNOLOGIA EM SISTEMAS PARA  
INTERNET**

**RITA DE CÁSSIA SILVEIRA SALVAJOLI LEITE**

## **CONSUMINDO UM WEB SERVICE: Calculadora**

Trabalho apresentado à disciplina de  
Arquitetura Orientada a Serviços, da  
Faculdade de Tecnologia de Lins, sob a  
orientação do Prof. Me. Júlio Fernando  
Liera.

**LINS/SP  
2º SEMESTRE/2021**

# Calculadora Webservice

Para os testes, utilizou-se o software SoapUI 5.6.0, Oracle e JDK 8.

## 1. DADOS DO SERVIÇO:

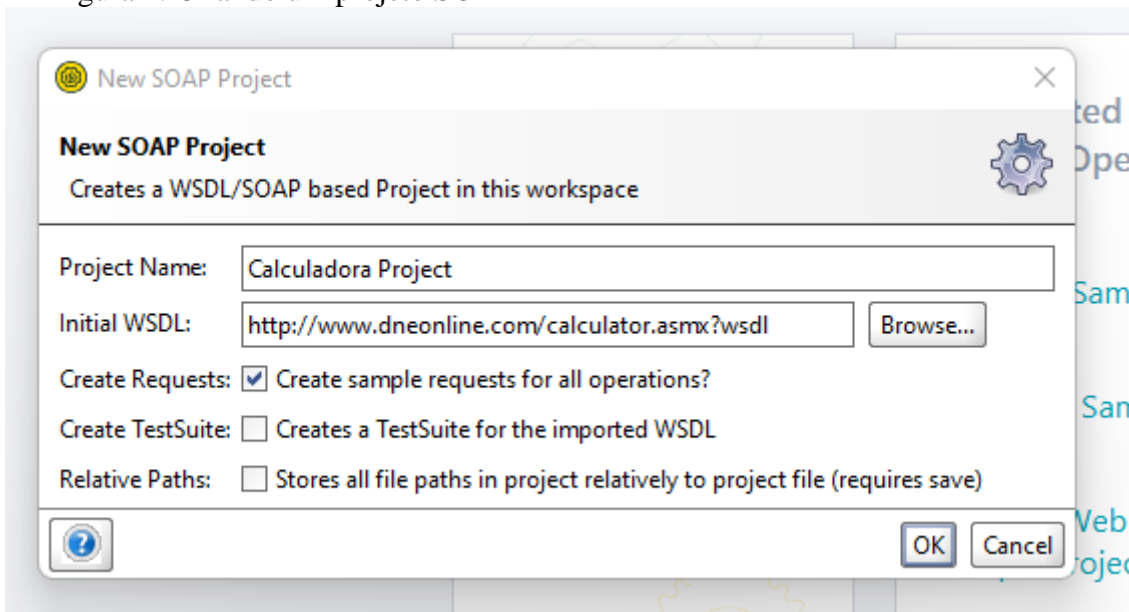
✚ O endereço do descritor do serviço (WSDL) é:

<http://www.dneonline.com/calculator.asmx?wsdl>

## 2. UTILIZAÇÃO:

Primeiramente, cria-se um projeto no SoapUI, utilizando o endereço descritor (WSDL), como mostra a imagem:

Figura 1: Criando um projeto SOAP



Fonte: Elaborado pela autora, 2021.

Segundamente, realizou-se um teste rápido para verificar se o webservice estava ativo e funcionando, assim, decidi fazer uma operação de soma, ao lado esquerdo na aba “Add”, foi possível ter a confirmação como demonstra a imagem:

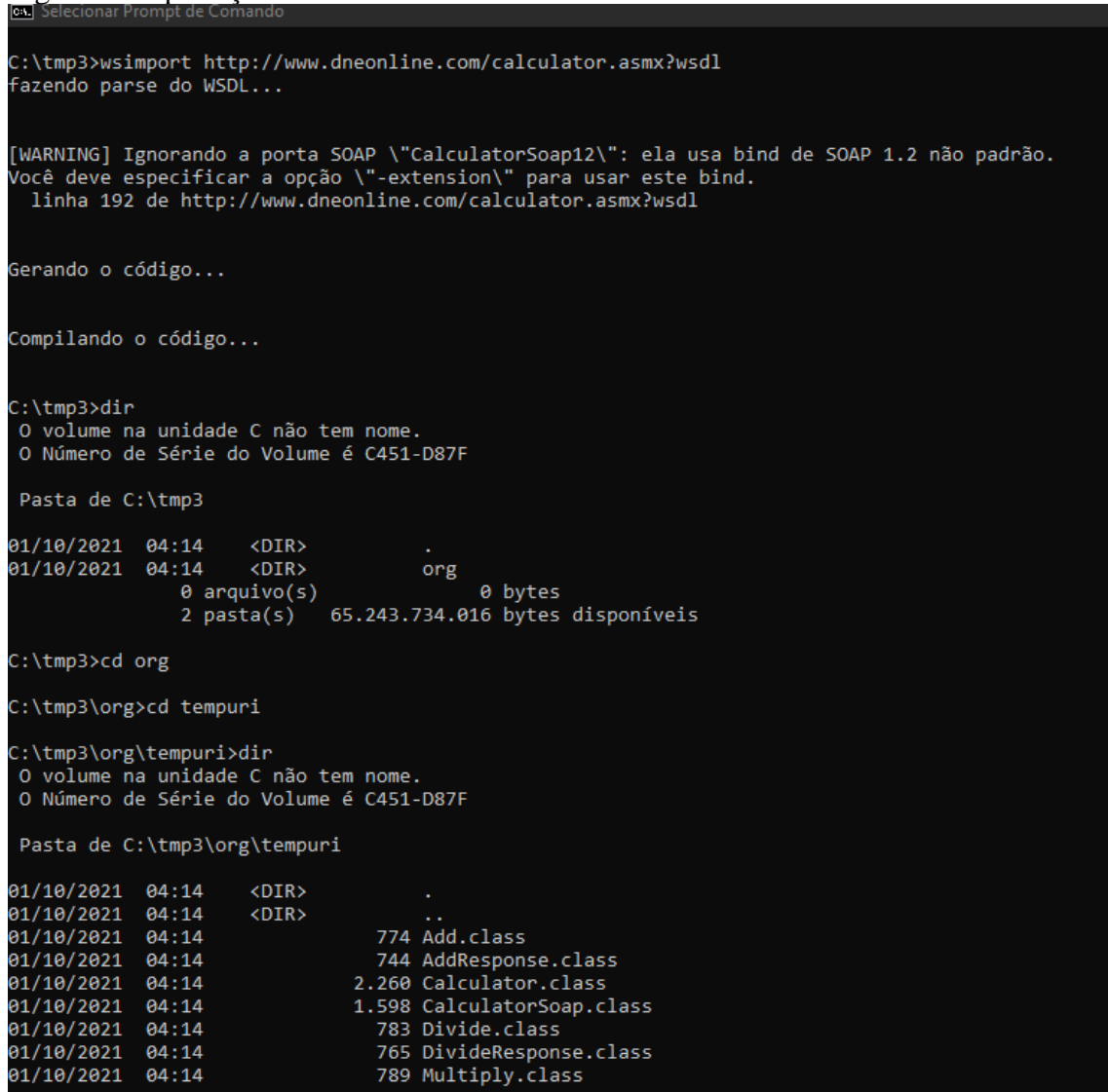
Figura 2: Realizando uma Soma



Fonte: Elaborado pela autora, 2021.

Logo após, agora no terminal (cmd), realizei a importação das classes através do comando “wsimport”.

Figura 3: Importação das Classes

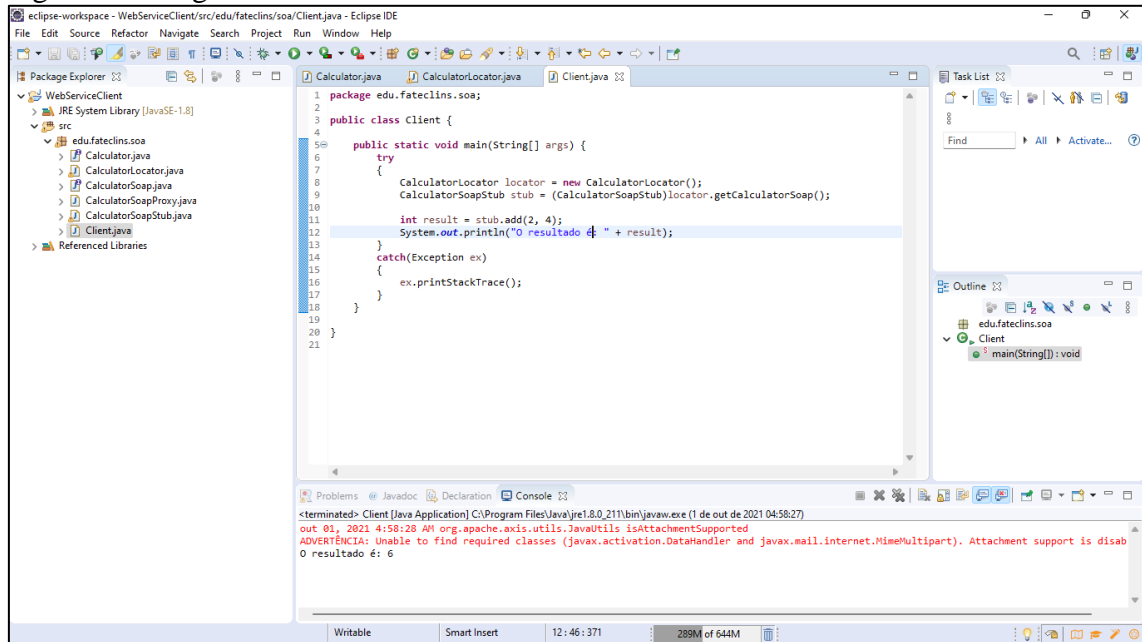


Fonte: Elaborado pela autora, 2021.

Após a importação das classes pelo *wsimport*, foi a vez de utilizarmos o Eclipse. A primeira coisa a se fazer é criar um projeto java comum, o qual nomeei de “WebServiceClient”, depois, clicar com o botão direito no projeto, ir à opção “Outros”, “Web Services”, “Web service client”; colocar o descritor do serviço, no local indicado, e finalizar, assim as classes já serão importadas.

Por fim, para testar se deu tudo certo, realizei uma operação de adição, com o resultado sendo mostrado no console.

Figura 4: Código final do Cliente



```
1 package edu.fateclins.soa;
2
3 public class Client {
4
5     public static void main(String[] args) {
6         try
7         {
8             CalculatorLocator locator = new CalculatorLocator();
9             CalculatorSoapStub stub = (CalculatorSoapStub)locator.getCalculatorSoap();
10
11             int result = stub.add(2, 4);
12             System.out.println("O resultado é: " + result);
13         }
14         catch(Exception ex)
15         {
16             ex.printStackTrace();
17         }
18     }
19 }
20 }
21 }
```

Console Output:

```
<terminated> Client [Java Application] C:\Program Files\Java\jre1.8.0_211\bin\javaw.exe [1 de out de 2021 04:58:27]
out 01, 2021 4:58:28 AM org.apache.axis.utils.JavaUtils isAttachmentSupported
ADVERTÊNCIA: Unable to find required classes (javax.activation.DataHandler and javax.mail.internet.MimeMultipart). Attachment support is disabled
O resultado é: 6
```

Fonte: Elaborado pela autora, 2021.