

SOAP: Simple Object Access Protocol

Douglas Feliciano, Jefferson Souza Rodrigues

Faculdade de Computação - Universidade Federal de Mato Grosso do Sul (UFMS)

Caixa Postal 579 - 79.070-900 - Campo Grande - MS - Brasil

douglasfeliciano10@hotmail.com , jeffersoncacula@gmail.com

Abstract: *This article describes a brief tutorial and definition of the Simple Object Access Protocol (SOAP). This tutorial will be based on RFCs 4227 and 3288 that are related to this application protocol. In addition, it exemplifies a use of the SOAP application protocol.*

Resumo: Neste artigo está descrito um breve tutorial e definição do protocolo SOAP (Simple Object Access Protocol). Esse tutorial terá como base as RFC's 4227 e 3288 que são relacionadas à esse protocolo de aplicação. Além disso, ele exemplifica uma utilização do protocolo de aplicação SOAP.

1. Introdução

Este memorando especifica uma ligação SOAP (Simple Object Access Protocol) para o núcleo do protocolo BEEP (Blocks Extensible Exchange Protocol). Uma ligação SOAP descreve como as mensagens SOAP são transmitidas na rede.

O SOAP é um padrão para especificar protocolo de mensagens baseado em XML (eXtensible Markup Language) usado para implementar uma ampla variedade de modelos distribuídos de mensagens para a aplicação utilizando o protocolo HTTP. Ele define um formato de mensagem e descreve vários padrões de mensagens, incluindo, entre outros, RPC (Remote Procedure Calling), notificação assíncrona de eventos, mensagens não confirmadas e encaminhamento por meio de intermediários SOAP.

O SOAP fornece uma maneira de se comunicar entre aplicativos executados em diferentes sistemas operacionais, com diferentes tecnologias e linguagens de programação. Independente da linguagem utilizada, você precisará de alguma interface para usar a biblioteca do protocolo que é considerada uma aplicação de backend (trabalha ao lado do servidor).

2. SOAP e WSDL

A arquitetura SOAP é baseada na Linguagem de Descrição de Web Services (WSDL - *Web Service Description Language*) e no SOAP. Esses dois padrões são publicados pelo *World Wide Web Consortium* (W3C). WSDL e SOAP arcabouços para especificar e implementar protocolos de aplicação e protocolos de transporte, respectivamente.

WSDL é usada para especificar detalhes específicos da aplicação, como quais operações são admitidas, os formatos dos dados a serem passados para a aplicação ou que são respondidos por essas operações, e se uma operação envolve uma resposta.

O papel do SOAP é facilitar a definição de um protocolo de transporte exatamente com a semântica desejada,, considerando recursos do protocolo tais como confiabilidade e segurança.

Tanto um quanto o outro consiste basicamente em uma linguagem de especificação de protocolos (como dito anteriormente o XML) e isso visa tornar as especificações acessíveis a ferramentas de software como compiladores stub e serviços de diretório.

3. Definindo protocolos de Aplicação

A WSDL escolheu um modelo de operação procedural para protocolos de aplicação. Ela possui vínculos predefinidos para protocolos baseados em HTTP e SOAP, com parâmetros que permitem que o projetista do protocolo faça ajustes finos ao mapeamento para esses protocolos. Existe um arcabouço para definir novos vínculos, mas os protocolos SOAP são dominantes.

Um aspecto decisivo de como a WSDL alivia o problema de especificar grandes quantidades de protocolos é através da reutilização daquilo que são basicamente módulos de especificação. A especificação da WSDL de um Web Service pode ser composta de vários documentos WSDL, e Documentos WSDL individuais também podem ser usados em outras especificações de Web Services. Essa modularidade facilita o desenvolvimento de uma especificação e também garante que, se duas especificações precisarem de alguns elementos que são idênticos, então esses elementos devem ser realmente idênticos.

A modularidade da WSDL deve ser familiar a qualquer um que já tenha desenvolvido algum software razoavelmente grande. Um documento WSDL não precisa ser uma especificação completa.

4. Definindo protocolos de transporte

Embora SOAP, às vezes seja chamado de protocolo, ele é considerado mais como um arcabouço para definição de protocolos. Conforme a especificação do SOAP 1.2 explica, “O SOAP fornece um arcabouço de mensagem simples cuja funcionalidade central se preocupa em oferecer extensibilidade”. O SOAP usa muitas das mensagens estratégicas da SWDL, incluindo mensagens definidas usando XML Schema, vínculos com protocolos básicos, Message Exchange Patterns e elementos reutilizáveis da especificação, identificados usando espaços de nomes XML.

O SOAP é usado para definir protocolos de transporte exatamente com os recursos necessários para dar suporte a um protocolo de aplicação em particular. O SOAP visa tornar viável definir muitos desses protocolos por meio do uso de componentes reutilizáveis. Cada Componente captura a informação de cabeçalho e a lógica que entram na implementação de um recurso em particular. Para definir um protocolo com um conjunto de recursos, basta escrever os componentes correspondentes.

O SOAP 1.2 introduziu uma abstração de recurso, que a especificação descreve desta forma: Um recurso SOAP é uma extensão do arcabouço de mensagens SOAP. Embora o SOAP não imponha restrições sobre o escopo em potencial desses recursos, alguns recursos de exemplo podem incluir “confiabilidade”, “segurança”, “correlação”, “roteamento” e padrões de troca de mensagens (MEP’s), como comunicações de solicitação/resposta, unidirecionais e peer-to-peer. Uma especificação de recurso SOAP precisa incluir:

- Um URI que identifique o recurso
- A informação de estado e processamento, descritos de forma abstrata, que é necessário em cada nó SOAP para implementar o recurso
- A informação a ser repassada para o próximo nó.

- (Se o recurso é um MEP) o ciclo de vida e relacionamentos temporais/causais das mensagens trocadas - por exemplo, respostas vêm após as solicitações e são enviadas a quem originou a solicitação.

5. Regras de Sintaxe

- Uma mensagem SOAP DEVE ser codificada usando XML
- Uma mensagem SOAP DEVE usar o namespace SOAP Envelope
- Uma mensagem SOAP deve usar o namespace de codificação SOAP
- Uma mensagem SOAP NÃO deve conter uma referência DTD
- Uma mensagem SOAP não deve conter instruções de processamento XML

Uma mensagem unidirecional envolve o envio de uma mensagem sem qualquer resposta sendo retornado.

O perfil BEEP para o SOAP consegue isso usando uma troca um-para-muitos, em que o cliente envia uma mensagem "MSG" contendo um envelope, e o servidor envia imediatamente de volta uma mensagem "NULL", antes do processamento o conteúdo do envelope.

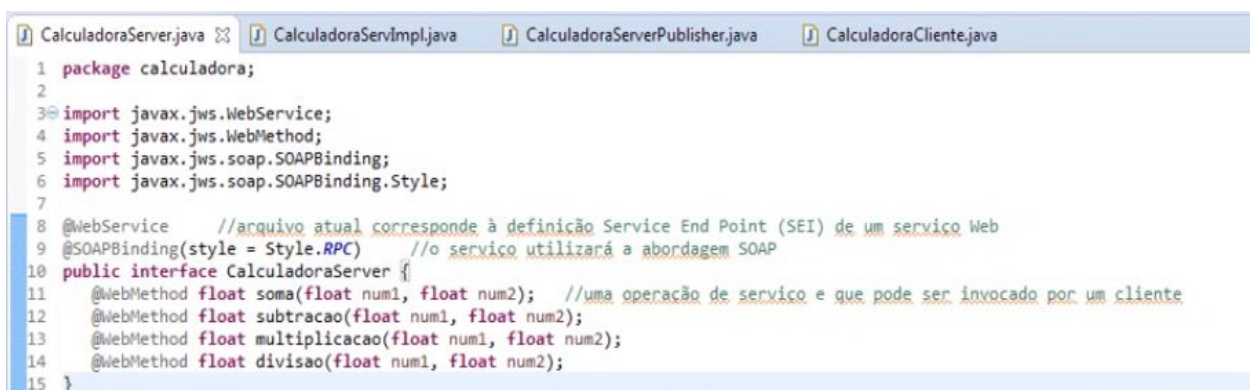
Uma troca de solicitação / resposta envolve o envio de uma solicitação, cujos resultados em uma resposta sendo retornada.

O termo "envelope" refere-se ao elemento de nível superior trocado por remetentes e destinatários SOAP. Dentro do envelope contém o corpo da requisição SOAP com uma mensagem que você deseja executar. Além disso, os termos "peer", "client", "server", "one-to-one" e "one-to-many" são usados no contexto do BEEP.

O perfil BEEP para o SOAP consegue isso usando uma troca de um-para-um, em que o cliente envia uma mensagem "MSG" contendo um envelope, e o servidor envia de volta uma mensagem "RPY" contendo um envelope.

Uma solicitação / troca de respostas N envolve o envio de uma solicitação, que resulta em zero ou mais respostas sendo retornadas.

6. Tutorial Exemplo: Calculadora usando SOAP



```

1 package calculadora;
2
3 import javax.jws.WebService;
4 import javax.jws.WebMethod;
5 import javax.jws.soap.SOAPBinding;
6 import javax.jws.soap.SOAPBinding.Style;
7
8 @WebService //arquivo atual corresponde à definição Service End Point (SEI) de um serviço Web
9 @SOAPBinding(style = Style.RPC) //o serviço utilizará a abordagem SOAP
10 public interface CalculadoraServer {
11     @WebMethod float soma(float num1, float num2); //uma operação de serviço e que pode ser invocado por um cliente
12     @WebMethod float subtracao(float num1, float num2);
13     @WebMethod float multiplicacao(float num1, float num2);
14     @WebMethod float divisao(float num1, float num2);
15 }

```

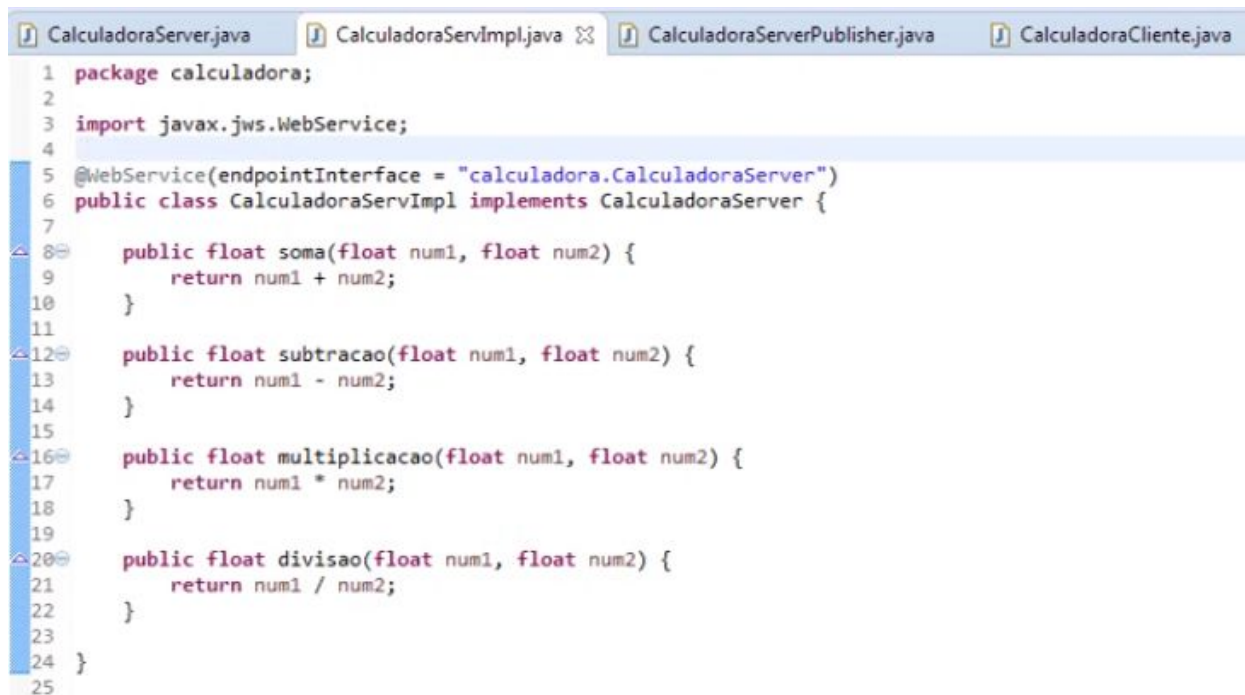
Figura 1.0 - CalculadoraServer.java

Linguagem de programação utilizada: java.

Nesse tutorial será mostrado a implementação de uma simples calculadora usando a abordagem SOAP.

No método CalculadoraServer.java (figura 1.0) a notação @WebService na linha 8 corresponde à definição Service And point de um serviço web. Dentro da interface Calculadora Server é chamado os métodos soma, subtração, multiplicação e divisão.

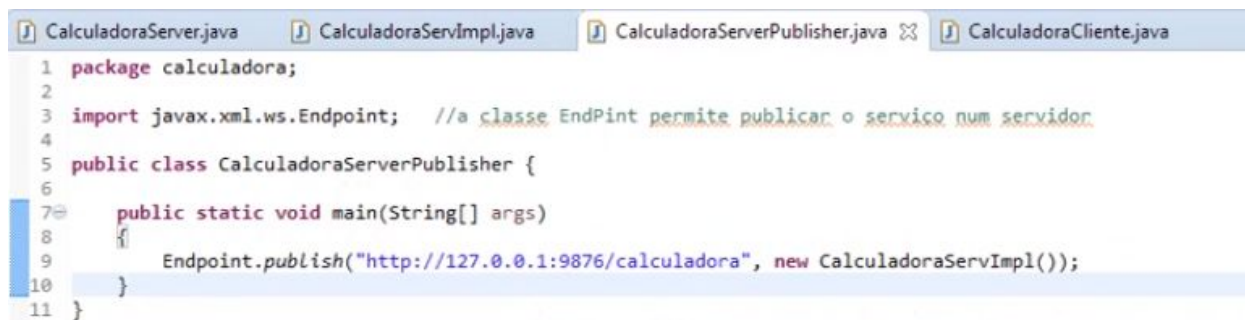
Para que esses serviços e métodos sejam executados, é necessário a implementação desses métodos que é feito pela classe CalculadoraServImpl.java (figura 1.1).



```
1 package calculadora;
2
3 import javax.jws.WebService;
4
5 @WebService(endpointInterface = "calculadora.CalculadoraServer")
6 public class CalculadoraServImpl implements CalculadoraServer {
7
8     public float soma(float num1, float num2) {
9         return num1 + num2;
10    }
11
12    public float subtracao(float num1, float num2) {
13        return num1 - num2;
14    }
15
16    public float multiplicacao(float num1, float num2) {
17        return num1 * num2;
18    }
19
20    public float divisao(float num1, float num2) {
21        return num1 / num2;
22    }
23
24 }
25
```

Figura 1.1 - CalculadoraServImpl.java

Toda essa interface e métodos precisa ser publicado e passado para o servidor. Essa função é da classe CalculadoraServerPublisher.java (figura 1.2) que possui um Endpoint.publish que passa como parâmetro o endereço e cria uma instância da classe CalculadoraServImpl.java.



```
1 package calculadora;
2
3 import javax.xml.ws.Endpoint; //a classe EndPint permite publicar o servico num servidor
4
5 public class CalculadoraServerPublisher {
6
7     public static void main(String[] args)
8     {
9         Endpoint.publish("http://127.0.0.1:9876/calculadora", new CalculadoraServImpl());
10    }
11 }
```

Figura 1.2 - CalculadoraServerPublisher.java

Ao digitarmos no navegador o caminho “127.0.0.1:9876/calculadora?wsdl” (figura 1.3) aparece um documento em XML baseado no protocolo WSDL e ele trás o conjunto de serviços de forma detalhada, descrevendo cada método criado na interface dentro da Calculado, no caso desse exemplo.

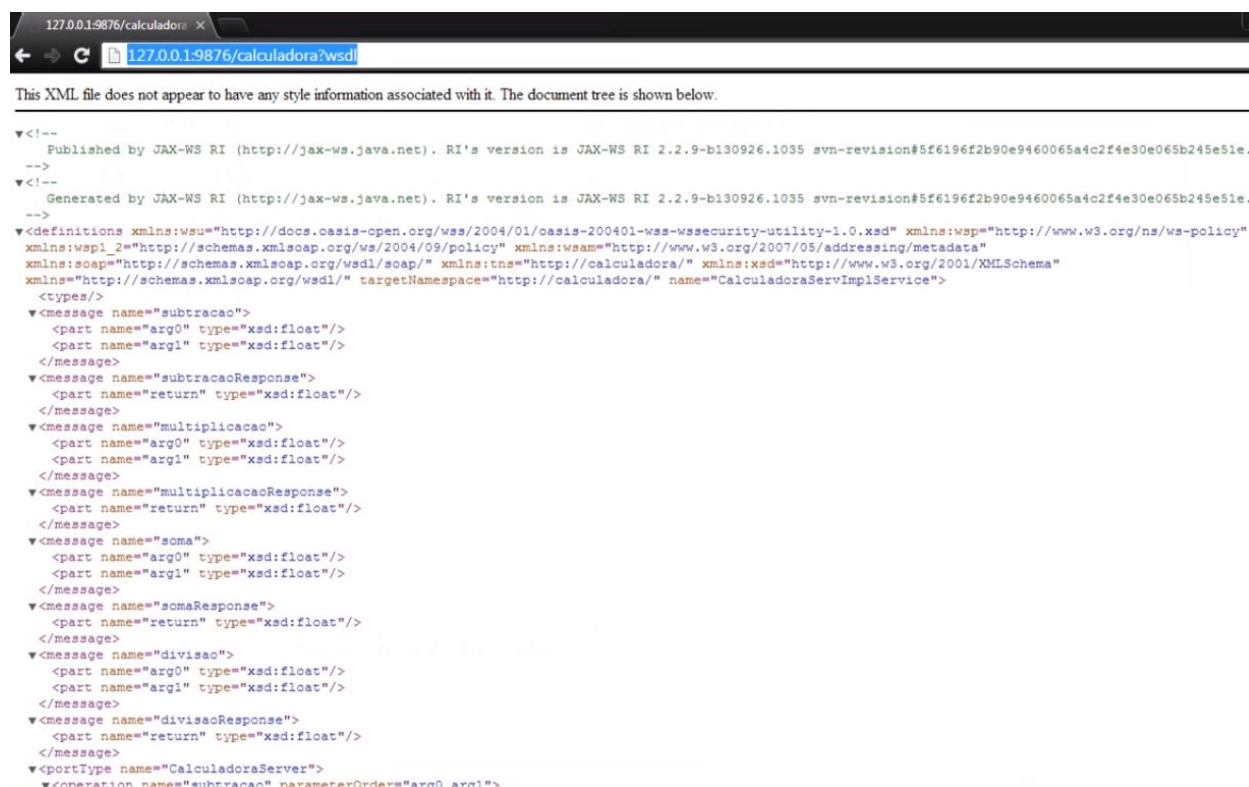


Figura 1.3 - XML do WSDL da Calculadora

Por fim, criamos uma classe cliente chamada CalculadoraCliente.java (figura 1.4) que é responsável em fazer com que as operações requisitadas sejam executadas. Ela cria uma nova calculadora e imprime o resultado de todas as operações que são chamadas e enviadas para o servidor.

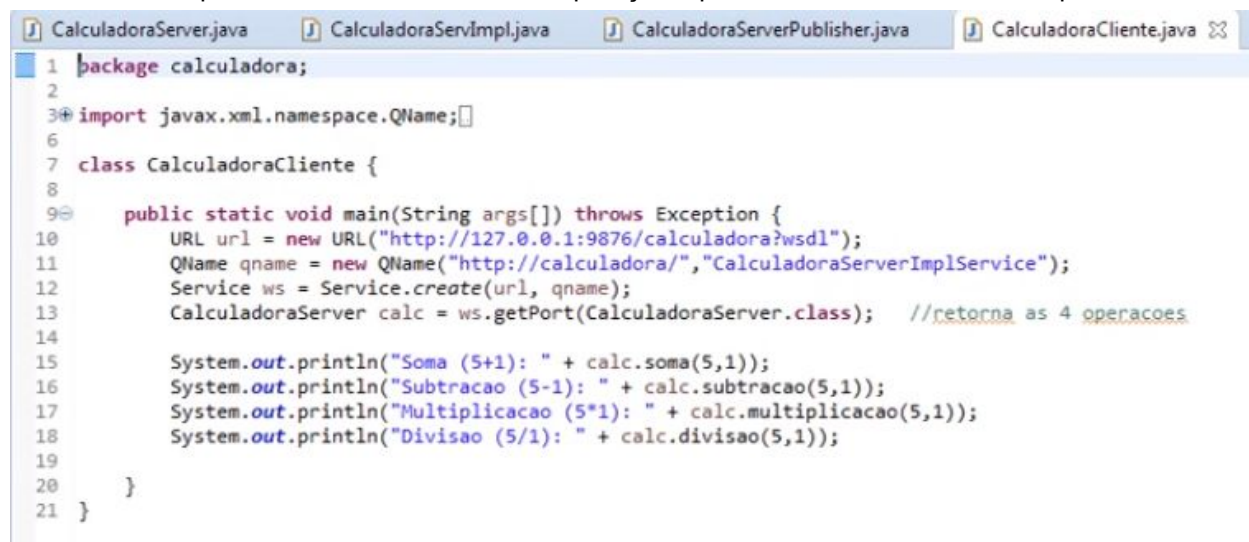
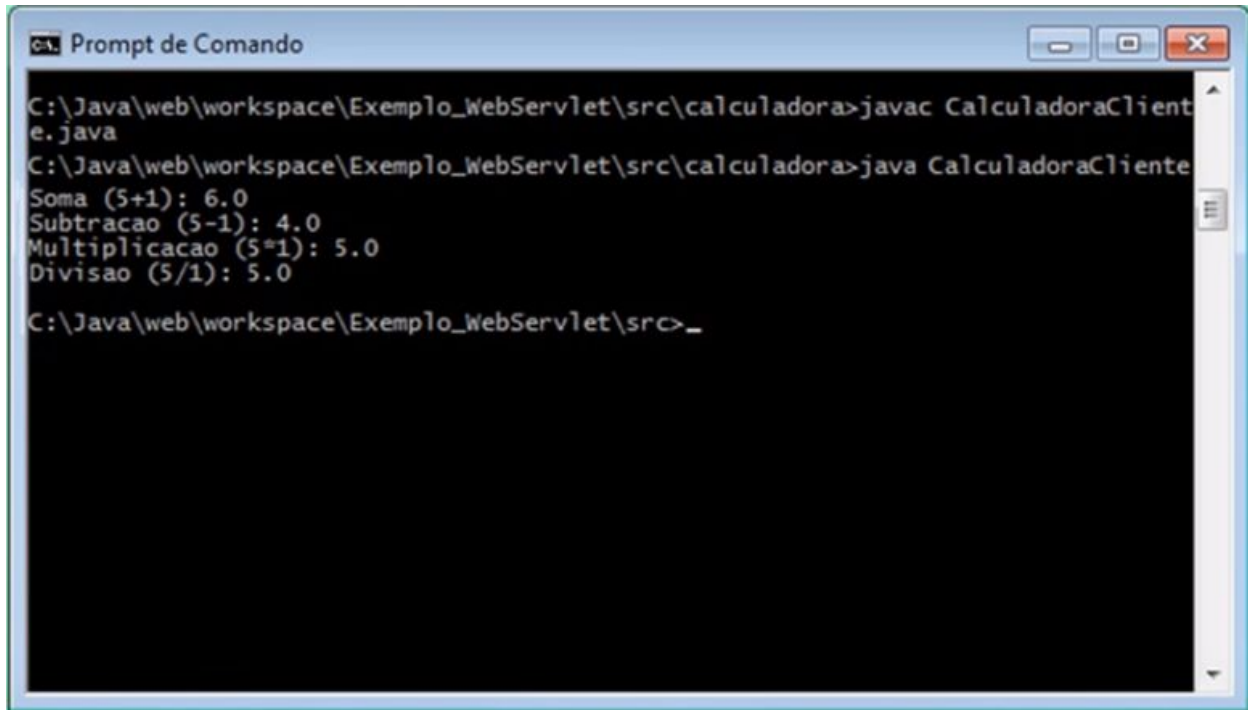


Figura 1.4 - CalculadoraCliente.java

O resultado, quando executado o Cliente em cmd diferente do Servidor é esse:



```
Prompt de Comando

C:\Java\web\workspace\Exemplo_WebServlet\src\calculadora>javac CalculadoraClient
e.java
C:\Java\web\workspace\Exemplo_WebServlet\src\calculadora>java CalculadoraCliente
Soma (5+1): 6.0
Subtracao (5-1): 4.0
Multiplicacao (5*1): 5.0
Divisao (5/1): 5.0
C:\Java\web\workspace\Exemplo_WebServlet\src>_
```

Figura 1.5 - Resultado da Execução da Calculadora usando SOAP

7. Referências Bibliográficas

- O'Tuathail, E. and M. Rose, "Using the Simple Object Access Protocol (SOAP) in Blocks Extensible Exchange Protocol (BEEP)", RFC 4227, Janeiro 2006. <http://www.ietf.org/rfc/rfc4227.txt>
- O'Tuathail, E. and M. Rose, "Using the Simple Object Access Protocol (SOAP) in Blocks Extensible Exchange Protocol (BEEP)", RFC 3288, Junho 2002. <http://www.ietf.org/rfc/rfc3288.txt>
- Computer Networks: A Systems Approach, 5e, Larry L. Peterson and Bruce S. Davie