



**PUC Minas**

# **Desenvolvimento de Aplicações Distribuídas**

**Web Service – Entrega 1 e 2**

**Alunos:** Magnum Dutra de Oliveira

**Data:** 25.11.14

**Professor:** Rommel Vieira Carneiro

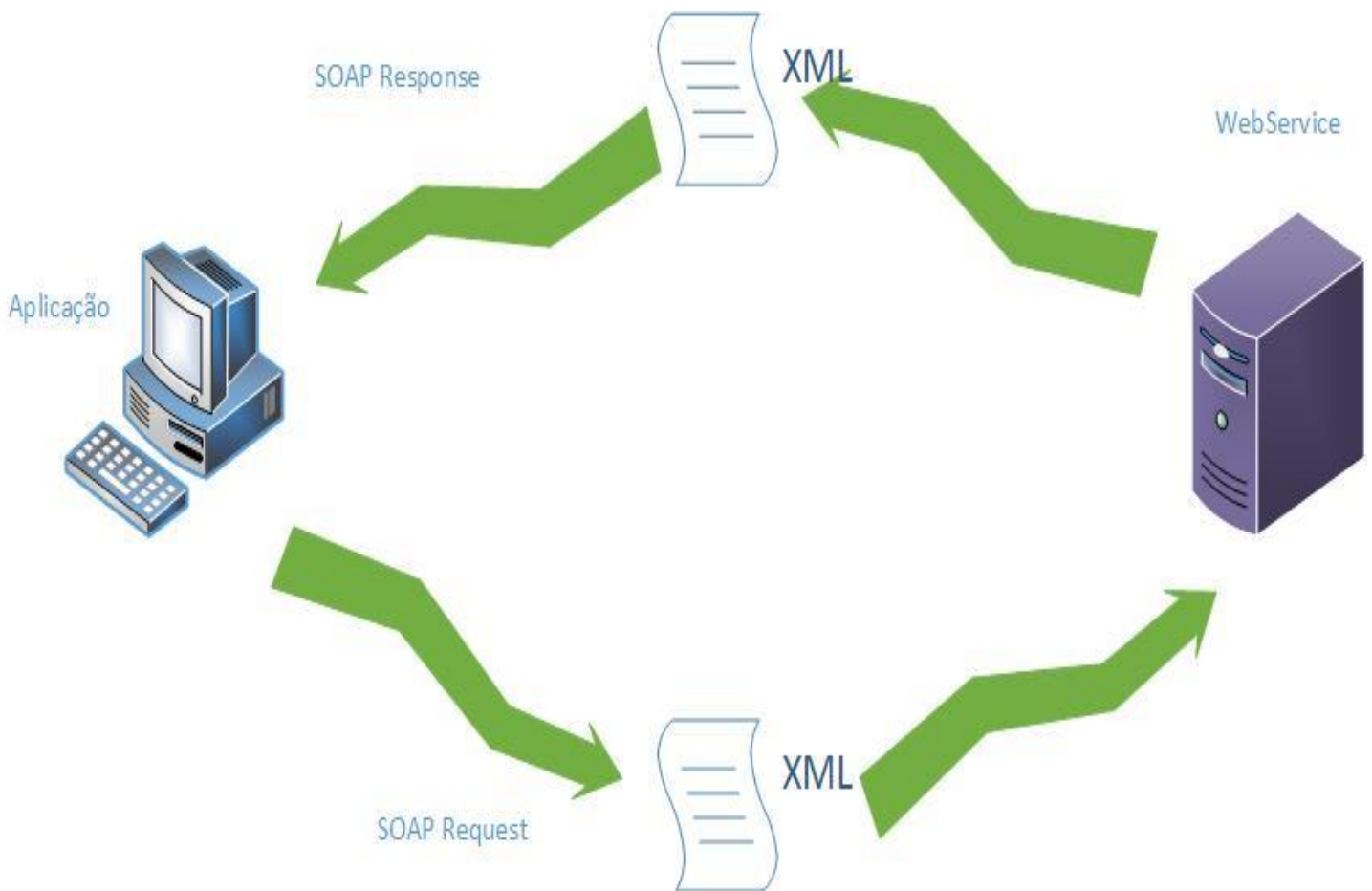
# Trabalho 1

1) *Desenvolva um módulo que recebe uma requisição em JSON ou XML que realize processamento e retorna pacote de dados em JSON ou XML.*

- **Propósito da Aplicação;**

A objeto de estudo, criei uma calculadora que executa os quatro comandos básicos da aritmética.

- **Diagrama da Arquitetura da Aplicação;**



- **Estrutura Física (Componentes de hardware e conectividade/protocolos de rede envolvidos);**

Utilizo o servidor do Webservice, o computador que fara requisição do serviço e o meio de transporte físico, cabeamento.

- **Estrutura lógica (módulos de software que são executados em cada um dos componentes);**

O webService, desenvolvido em C#, executa processamento com base nos parâmetros recebidos através de uma requisição realizada via POST.

Os parâmetros são:

<decimal></decimal> -----→ Valor utilizado na operação

<decimal></decimal> -----→ Valor utilizado na operação

<decimal></decimal> -----→ Operação requerida

Com base nisso é retornado uma resposta em XML com o resultado da operação.

A requisição é executada na aplicação do cliente, o processamento no servidor WebService e a resposta recebida pela aplicação do Cliente.

- **Detalhamento da estrutura dos dados trocados entre cliente e servidor (JSON ou XML);**

Não utilizei muitas Tag's no XML. Como iria retornar o resultado do cálculo somente. Abaixo a estrutura:

#### **Requisição Http, realizada via POST.**

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<decimal xmlns="http://AplicacoesDistribuidas/">3,1,1</decimal>
```

#### **Resposta SOAP XML do WebService**

```
<decimal>7</decimal>
```

- Código fonte do aplicativo servidor;

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Services;

namespace WebService
{
    /// <summary>
    /// Summary description for Service1
    /// </summary>
    [WebService(Namespace = "http://AplicacoesDistribuidas/")]
    [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
    [System.ComponentModel.ToolboxItem(false)]
    // To allow this Web Service to be called from script, using ASP.NET AJAX, uncomment the
    following line.
    // [System.Web.Script.Services.ScriptService]

    public class Service1 : System.Web.Services.WebService
    {
        public enum OperacaoTipo
        {
            Soma = 1,
            Subtracao = 2,
            Multiplicacao = 3,
            Divisao = 4
        }

        [WebMethod]
        public decimal Calcular(decimal Numero1, decimal Numero2, OperacaoTipo operacaoTipo)
        {
            decimal Resultado;
            switch (operacaoTipo)
            {
                case OperacaoTipo.Soma:
                    Resultado = decimal.Add(Numero1, Numero2);
                    break;
                case OperacaoTipo.Subtracao:
                    Resultado = decimal.Subtract(Numero1, Numero2);
                    break;
                case OperacaoTipo.Multiplicacao:
                    Resultado = decimal.Multiply(Numero1, Numero2);
                    break;
                case OperacaoTipo.Divisao:
                    Resultado = decimal.Divide(Numero1, Numero2);
                    break;
                default:
                    Resultado = 0;
                    break;
            }
            return Resultado;
        }
    }
}
```

- Impressões de tela.

## Service1

Clique [aqui](#) para obter uma lista completa das operações.

### Calcular

Testar

Para testar a operação usando o protocolo HTTP POST, clique no botão 'Chamar'.

Parâmetro	Valor
Numero1:	<input type="text" value="2"/>
Numero2:	<input type="text" value="4"/>
operacaoTipo:	<input type="text" value="3"/>

### SOAP 1.1

O exemplo a seguir mostra uma solicitação e uma resposta SOAP 1.1. Os espaços reservados mostrados devem ser substituídos por valores reais.

```
POST /Service1.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://AplicacoesDistribuidas/Calcular"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <Calcular xmlns="http://AplicacoesDistribuidas/">
      <Numero1>decimal</Numero1>
      <Numero2>decimal</Numero2>
      <operacaoTipo>Soma or Subtracao or Multiplicacao or Divisao</operacaoTipo>
    </Calcular>
  </soap:Body>
</soap:Envelope>

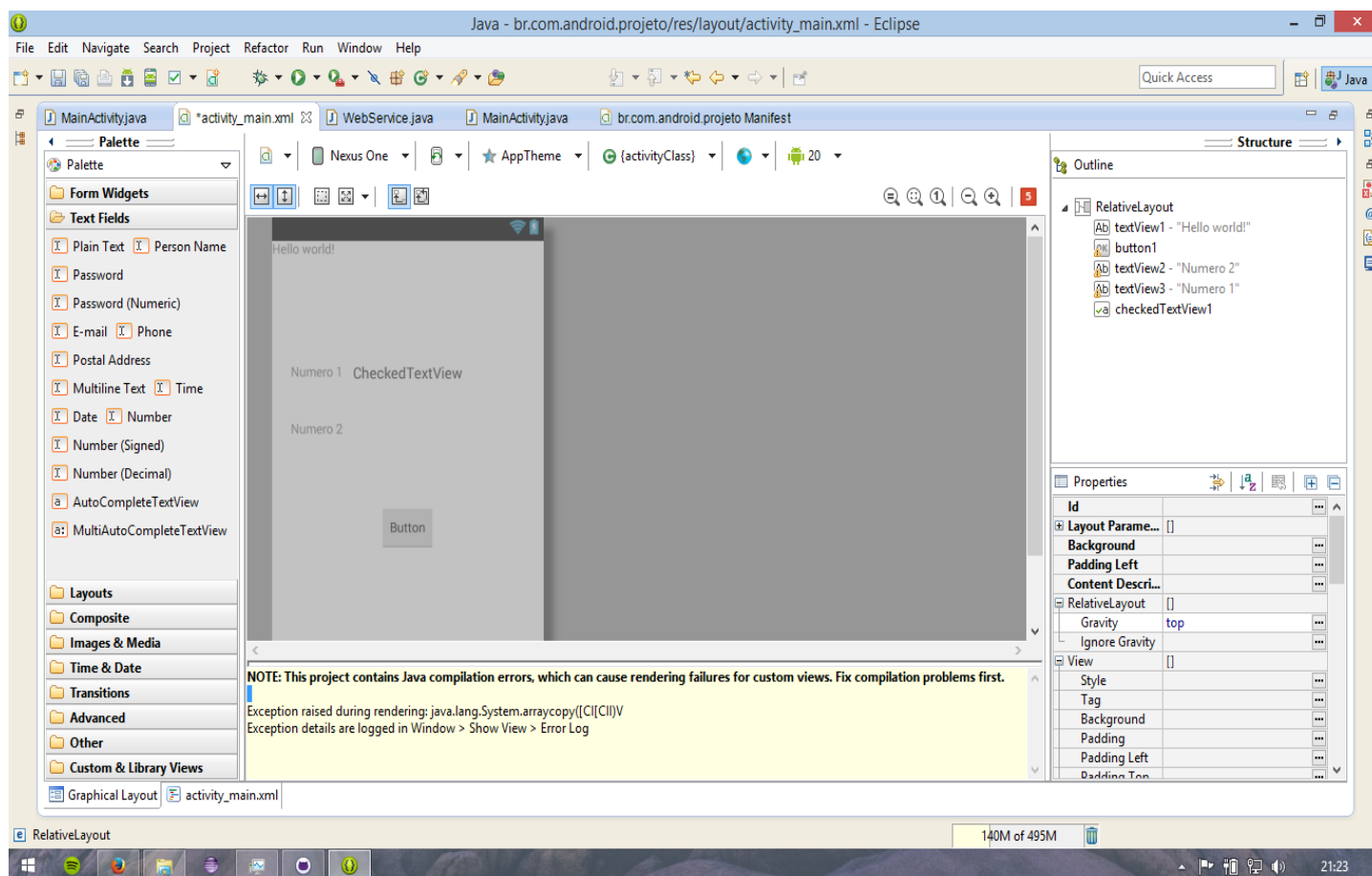
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
```

O documento XML não está associado a estilos. A estrutura do documento está representada abaixo.

```
<decimal>8</decimal>
```

- Parte Cliente

Professor, os prints da tela não consegui realizar porque esta dando um excessao desconhecida.



## • Codigo

```
package android.projeto;

import org.ksoap2.SoapEnvelope;
import org.ksoap2.serialization.SoapObject;
import org.ksoap2.serialization.SoapSerializationEnvelope;
import org.ksoap2.transport.HttpTransportSE;

public class Webservice {
    private static final String NAMESPACE = "com.service.Calculator";
    private static final String URL = "localhost:8080";
    private static final String SOAP_ACTION = "Calculator";

    public String webserviceOperacao(String operacao, Integer numA, Integer numB) {
        SoapObject request = new SoapObject(NAMESPACE, operacao);
        SoapSerializationEnvelope envelope = new
        SoapSerializationEnvelope(SoapEnvelope.VERSION1);
        String retorno = null;
        // Adiciona parâmetros
        request.addProperty("numA", numA);
        request.addProperty("numB", numB);
        envelope.setOutputSoapObject(request);
        HttpTransportSE androidHttpTransport = new HttpTransportSE(URL);
        try {
            androidHttpTransport.call(SOAP_ACTION, envelope);
            SoapObject resultsRequestSOAP = (SoapObject) envelope.bodyIn;
            retorno = resultsRequestSOAP.toString();
        } catch (Exception e) {
            e.printStackTrace();
        }
        return retorno;
    }
}
```

```
package android.projeto;

import android.app.Activity;
import android.os.Bundle;
import android.util.Log;

public class MainActivity extends Activity {
    private WebService ws = new WebService();

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        operacao("somar", 2, 2, "Somar");
        operacao("subtrair", 6, 3, "Subtrair");
        operacao("multiplicar", 7, 3, "Multiplicar");
        operacao("dividir", 21, 3, "Dividir");
    }

    public void operacao(String operacao, Integer numA, Integer numB, String texto) {
        operacao = ws.webServiceOperacao(operacao, numA, numB);
        // Filtra o valor retornado pelo WebService
        String[] primeiraParte = operacao.split("=");
        String[] segundaParte = primeiraParte[1].split(";");
        String retorno = segundaParte[0];
        Log.i("webservice", texto + ": " + Integer.valueOf(retorno));
    }
}
```