



# **JAVA WEB SERVICES**



## **WS SOAP**

**ERIC GUSTAVO CORONEL CASTILLO**

[www.desarrollasoftware.com](http://www.desarrollasoftware.com)

[gcoronelc@gmail.com](mailto:gcoronelc@gmail.com)



# Contenido

<b>1</b>	<b>CASO A DESARROLLAR .....</b>	<b>3</b>
1.1	ENUNCIADO .....	3
1.2	SERVICIO.....	3
1.3	CONSULTANDO EL WSDL .....	4
1.4	PROBANDO EL SERVICIO CON SOAPUI.....	5
1.5	CREANDO UN CLIENTE JAVA .....	6
1.6	CREANDO UN CLIENTE PHP.....	8
1.7	CREAR UN CLIENTE .NET CON C# .....	9
1.8	CREANDO UN CLIENTE ANDROID .....	10
<b>2</b>	<b>EJERCICIO PROPUESTO.....</b>	<b>11</b>
<b>3</b>	<b>RETO.....</b>	<b>11</b>



# 1 CASO A DESARROLLAR

## 1.1 Enunciado

Implemente un Servicio Web SOAP utilizando NetBeans y GlassFish que implemente los siguientes servicios:

- Suma de dos números
- Resta de dos números

## 1.2 Servicio

```
package pe.egcc.ws;

import javax.jws.WebMethod;
import javax.jws.WebService;

/**
 *
 * @author Eric Gustavo Coronel Castillo
 * @blog www.desarrollasoftware.com
 * @email gcoronelc@gmail.com
 */
@WebService(serviceName = "WSMate")
public class MateService {

    @WebMethod
    public int sumar(int n1, int n2){
        return (n1+n2);
    }

    @WebMethod
    public int resta(int n1, int n2){
        return (n1-n2);
    }
}
```



## 1.3 Consultando el WSDL

<http://localhost:8080/WSMate/WSMate?wsdl>

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<!--
  Published by JAX-WS RI (http://jax-ws.java.net). RI's version is Metro/2.3.1-b419 (branches/2.3.1.x-79
-->
<!--
  Generated by JAX-WS RI (http://jax-ws.java.net). RI's version is Metro/2.3.1-b419 (branches/2.3.1.x-79
-->
<definitions xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
1.0.xsd" xmlns:wsp="http://www.w3.org/ns/ws-policy"
xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://ws.egcc.pe/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://schemas.xmlsoap.org/wsdl/"
targetNamespace="http://ws.egcc.pe/" name="WSMate">
  <types>...</types>
  <message name="resta">...</message>
  <message name="restaResponse">...</message>
  <message name="sumar">...</message>
  <message name="sumarResponse">...</message>
  <portType name="MateService">...</portType>
  <binding name="MateServicePortBinding" type="tns:MateService">...</binding>
  <service name="WSMate">...</service>
</definitions>
```



## 1.4 Probando el Servicio con SoapUI

Sumar:

The screenshot shows the SoapUI interface for a SOAP request. The URL is `http://localhost:8080/WSMate/WSMate`. The request is a SOAP envelope with a `ws:sumar` body containing `arg0=20` and `arg1=10`. The response is a SOAP envelope with a `ns2:sumarResponse` body containing `return=30`. The status bar shows a response time of 45ms (221 bytes).

```
<?xml version='1.0' encoding='utf-8'>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <ws:sumar>
      <arg0>20</arg0>
      <arg1>10</arg1>
    </ws:sumar>
  </soapenv:Body>
</soapenv:Envelope>
```

```
<?xml version='1.0' encoding='utf-8'>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:sumarResponse xmlns:ns2="http://ws.egcc.pe/">
      <return>30</return>
    </ns2:sumarResponse>
  </S:Body>
</S:Envelope>
```

response time: 45ms (221 bytes)

Restar:

The screenshot shows the SoapUI interface for a SOAP request. The URL is `http://localhost:8080/WSMate/WSMate`. The request is a SOAP envelope with a `ws:resta` body containing `arg0=50` and `arg1=20`. The response is a SOAP envelope with a `ns2:restaResponse` body containing `return=30`. The status bar shows a response time of 4ms (221 bytes).

```
<?xml version='1.0' encoding='utf-8'>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <ws:resta>
      <arg0>50</arg0>
      <arg1>20</arg1>
    </ws:resta>
  </soapenv:Body>
</soapenv:Envelope>
```

```
<?xml version='1.0' encoding='utf-8'>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:restaResponse xmlns:ns2="http://ws.egcc.pe/">
      <return>30</return>
    </ns2:restaResponse>
  </S:Body>
</S:Envelope>
```

response time: 4ms (221 bytes)



## 1.5 Creando un Cliente Java

1. Crear una aplicación de escritorio.

The 'New Java Application' dialog box is shown with the 'Name and Location' tab selected. The 'Steps' list on the left shows '1. Choose Project' and '2. Name and Location'. The 'Name and Location' section contains the following fields and options:

- Project Name:** WSMateClient
- Project Location:** H:\JAVA\10\_WS\CursoVer02\Tema06 (with a 'Browse...' button)
- Project Folder:** H:\JAVA\10\_WS\CursoVer02\Tema06\WSMateClient
- ☐ **Use Dedicated Folder for Storing Libraries**  
Libraries Folder: (empty field with a 'Browse...' button)  
Different users and projects can share the same compilation libraries (see Help for details).
- ☒ **Create Main Class** pe.egcc.app.ClasePrincipal

At the bottom, there are buttons for '< Back', 'Next >', 'Finish' (highlighted with a dashed border), 'Cancel', and 'Help'.

2. Crear el Web Service Client.

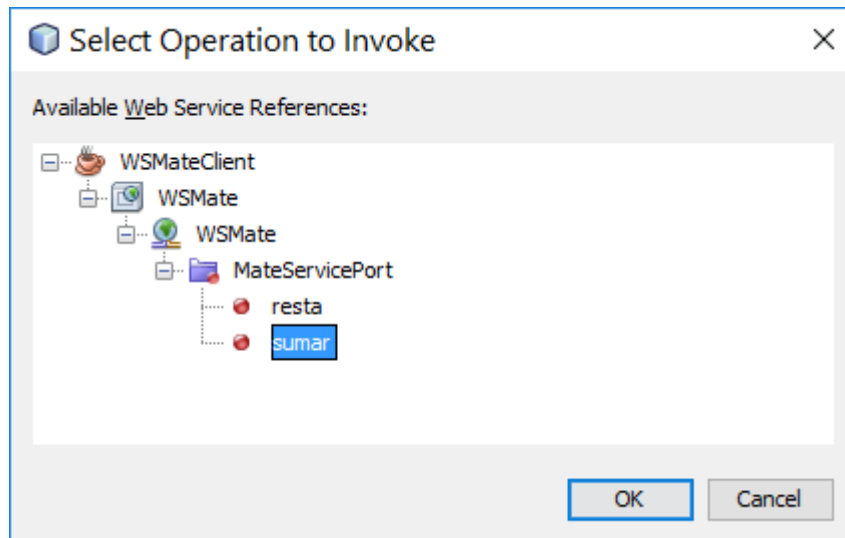
The 'New Web Service Client' dialog box is shown with the 'WSDL and Client Location' tab selected. The 'Steps' list on the left shows '1. Choose File Type' and '2. WSDL and Client Location'. The 'WSDL and Client Location' section contains the following fields and options:

- Specify the WSDL file of the Web Service.**
  - ☐ **Project:** (empty field with a 'Browse...' button)
  - ☐ **Local File:** (empty field with a 'Browse...' button)
  - ☒ **WSDL URL:** http://localhost:8080/WSMate/WSMate?wsdl
  - ☐ **IDE Registered:** (empty field with a 'Browse...' button)
- Specify a package name where the client java artifacts will be generated:**
  - Project:** WSMateClient
  - Package:** pe.egcc.wsclient (dropdown menu)
- ☐ **Generate Dispatch code**

At the bottom, there are buttons for '< Back', 'Next >', 'Finish' (highlighted with a dashed border), 'Cancel', and 'Help'.



### 3. Probando el servicio **sumar**.



```
package pe.egcc.app;

/**
 *
 * @author Eric Gustavo Coronel Castillo
 * @blog www.desarrollasoftware.com
 * @email gcoronelc@gmail.com
 */
public class Prueba01 {

    public static void main(String[] args) {
        System.out.println("30 + 6 = " + sumar(30,6));
    }

    private static int sumar(int arg0, int arg1) {
        pe.egcc.wsclient.WSMate service = new pe.egcc.wsclient.WSMate();
        pe.egcc.wsclient.MateService port = service.getMateServicePort();
        return port.sumar(arg0, arg1);
    }

}
```

### 4. Probando el servicio **resta**.

Crear la clase de prueba para probar el servicio **resta**.



## 1.6 Creando un cliente PHP

1. Probando el servicio **sumar**.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title></title>
  </head>
  <body>
    <?php
      //url del servicio
      $servicio = "http://localhost:8080/WSMate/WSMate?wsdl";
      //parametros de la llamada
      $parametros = array();
      $parametros["arg0"] = 5;
      $parametros["arg1"] = 6;
      // Creando el servicio
      $soapClient = new SoapClient($servicio, $parametros);
      //llamado al método sumar
      $result = $soapClient->sumar($parametros);
      // Mostrando el resultado
      echo "Tipo: " . getType($result) . "<br/>";
      print_r($result);
      echo "<br/>Suma: " . $result->return;
    ?>
  </body>
</html>
```

2. Probando el servicio **resta**.

Desarrolle un programa PHP para probar el servicio **resta**.





## 1.7 Crear un Cliente .NET con C#

1. Crear una aplicación Windows con C#
2. Agregarle una referencia al Servicio Web.

Agregar referencia de servicio

Para ver una lista de los servicios disponibles en un servidor específico, escriba una dirección URL de servicio y haga clic en Ir. Para buscar servicios disponibles, haga clic en Detectar.

Dirección:

Servicios:	Operaciones:
<ul style="list-style-type: none"><li>WSMate</li><li><b>MateService</b></li></ul>	<ul style="list-style-type: none"><li>resta</li><li>sumar</li></ul>

1 servicios encontrados en la dirección 'http://localhost:8080/WSMate/WSMate?wsdl'.

Espacio de nombres:



3. Diseñe el siguiente formulario:

Ejemplo de WS

Número 1:

Número 2:

Procesar

Suma: 110

Resta: 50

4. Programación del botón:

```
private void btnProcesar_Click(object sender, EventArgs e)
{
    ServiceReference1.MateServiceClient soap;
    soap = new ServiceReference1.MateServiceClient();

    int n1 = int.Parse(txtNum1.Text);
    int n2 = int.Parse(txtNum2.Text);

    int suma = soap.sumar(n1, n2);
    int resta = soap.resta(n1, n2);

    lblSuma.Text = "Suma: " + suma;
    lblResta.Text = "Resta: " + resta;
}
```

## 1.8 Creando un Cliente Android

Desarrollar una aplicación Android para consumir el servicio web.



---

## 2 EJERCICIO PROPUESTO

Desarrollar un servicio web que permita convertir:

- De Soles a Dólares y viceversa
- De Soles a Euros y viceversa
- De Dólares a Euros y viceversa

Luego programar los clientes en:

- SOAPUI
- Java
- PHP
- .NET
- Android

## 3 RETO

Utilizando la base de datos EUREKABANK, desarrollar los servicios web que permita consultar:

- Los datos de una cuenta.
- Los movimientos de una cuenta.