

Asignatura: Sistemas Distribuidos**Fecha:** Octubre - 2016**Nombre de la Practica:** Conexión a un Servidor JAXWS**Unidad Temática:** Servicios**Contenido Programático:** Web Services**Objetivo de la Practica:** Consumir exitosamente un WEB Service**Fecha de Entrega:** 6 de Octubre de 2016**Corte:** Segundo**Puntos:** 5**Enunciado:****Instrucciones para crear el proyecto servidor del web service:**

1. Crear un proyecto java en eclipse llamado JAXWSServer
2. Crear un paquete llamado co.dlegoprog.webservice
3. Crear una clase interfaz llamada Calculadora dentro del paquete creado

```
import javax.jws.WebMethod;  
import javax.jws.WebService;  
  
@WebService  
public interface Calculadora {  
  
    @WebMethod  
    public String sumar(Integer numero1, Integer numero2);  
  
    @WebMethod  
    public String restar(Integer numero1, Integer numero2);  
  
    @WebMethod  
    public String multiplicar(Integer numero1, Integer numero2);  
  
    @WebMethod  
    public String dividir(Integer numero1, Integer numero2);  
}
```

A esta interfaz se crearan los métodos de una calculadora básica (sumar,restar,multiplicar y dividir)

4. Crear una clase java llamada CalculadoraImpl (dentro del paquete creado anteriormente) la cual tendrá la implementación de la interfaz anteriormente creada



```
public class CalculadoraImpl {  
  
    public String sumar(Integer numero1, Integer numero2) {  
        Integer resultado = numero1 + numero2;  
        return "El resultado de la suma "+numero1+"-"+numero2+" es: " + resultado;  
    }  
  
    public String restar(Integer numero1, Integer numero2) {  
        Integer resultado = numero1 - numero2;  
        return "El resultado de la resta "+numero1+"-"+numero2+" es: " + resultado;  
    }  
  
    public String multiplicar(Integer numero1, Integer numero2) {  
        Integer resultado = numero1 * numero2;  
        return "El resultado de la multiplicacion "+numero1+"*"+numero2+" es: " + resultado;  
    }  
  
    public String dividir(Double numero1, Double numero2) {  
        Double resultado = numero1 / numero2;  
        return "El resultado de la division "+numero1+"/"+numero2+" es: " + resultado;  
    }  
}
```

5. Crear una clase java main llamada CalculadoraWSPublisher (dentro del paquete creado anteriormente) la cual se encargara de publicar el web service

```
public class CalculadoraWSPublisher {  
  
    public static void main(String[] args) {  
        Endpoint.publish("http://localhost:8089/WS/Calculadora",new CalculadoraImpl());  
    }  
}
```

En esta clase se necesita mencionar la URL especifica con la cual se accederá al web service con su respectivo puerto el cual debe estar libre para su uso

6. Correr la clase CalculadoraWSPublisher la cual dejara la consola de eclipse iniciada la cual no debe detenerse hasta que se termine el ejercicio
7. Verificar si el web service esta arriba accediendo a la URL mencionada en la clase CalculadoraWSPublisher mas un parámetro adicional desde el navegador web, quedando de la siguiente forma:

<http://localhost:8089/WS/Calculadora?wsdl>

Al buscarse en el navegador debería mostrar un archivo xml el cual es el wsdl que contendrá la información de los métodos y el servicio a llamar

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>  
<message name="sumar">  
  <part name="parameters" element="tns:sumar"/>  
</message>  
<message name="sumarResponse">  
  <part name="parameters" element="tns:sumarResponse"/>  
</message>  
<message name="restar">  
  <part name="parameters" element="tns:restar"/>  
</message>  
<message name="restarResponse">  
  <part name="parameters" element="tns:restarResponse"/>  
</message>
```

Nota: SI esto se muestra es porque el servicio se levanto correctamente, de lo contrario revisar

la configuración del web service o la disponibilidad de los puertos.

Instrucciones para crear el proyecto cliente del web service:

1. Crear un proyecto java en eclipse llamado JAXWSCClient
2. Crear un paquete llamado co.dlegoprog.webservice
3. Abrir la ventana de comandos e ingresar la siguiente línea de comandos:
Cd ruta_de_proyecto_cliente/src wsimport -s . http://localhost:8089/WS/Calculadora?wsdl

Para ejecutar este commando es necesario tener java instalado, puede ser jre o jdk y tener el bin de java dentro de las variables de entorno.

En caso contrario escribir el comando

C:\WebService\JAXWSCClient\src>C:\WTW\Java\jdk1.7.0_60\bin\wsimport -s .

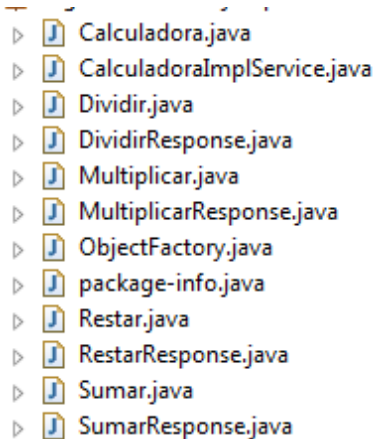
http://localhost:8089/WS/Calculadora?wsdl

```
C:\WebService\JAXWSCClient\src>C:\WTW\Java\jdk1.7.0_60\bin\wsimport -s . http://localhost:8089/WS/Calculadora?wsdl
parsing WSDL...

Generating code...

Compiling code...
```

Cuando termine el comando de realizar las funciones respectivas se refresca el proyecto para ver que el src creo un paquete nuevo con las clases que se utilizan para consumir el web service.



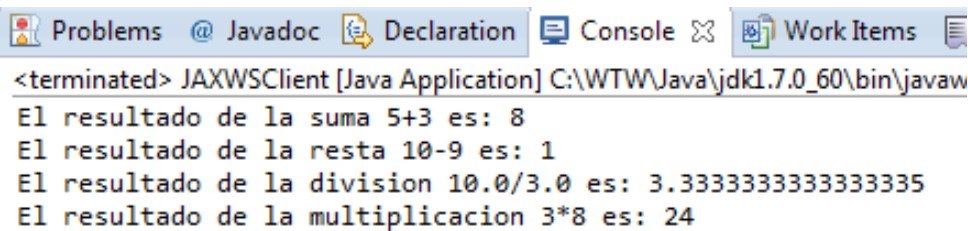
4. En el paquete co.dlegoprog.webservice.client se crea una clase java main llamada JAXWSCClient



```
public class JAXWSCClient {  
    public static void main(String[] args) {  
        CalculadoraImplService calculadoraService = new CalculadoraImplService();  
        Calculadora calculadora = calculadoraService.getCalculadoraImplPort();  
        System.out.println(calculadora.sumar(5,3));  
        System.out.println(calculadora.restar(10,9));  
        System.out.println(calculadora.dividir(10.0,3.0));  
        System.out.println(calculadora.multiplicar(3,8));  
    }  
}
```

Esta clase llamara a las clases creadas automáticamente para consumir el web service y lo único que hace es llamar las funciones respectivas de dicho web service y enviarle los parámetros correspondientes.

5. Ejecutar la clase JAXWSCClient y ver el resultado que da el web service



```
<terminated> JAXWSCClient [Java Application] C:\WTW\Java\jdk1.7.0_60\bin\javaw  
El resultado de la suma 5+3 es: 8  
El resultado de la resta 10-9 es: 1  
El resultado de la division 10.0/3.0 es: 3.3333333333333335  
El resultado de la multiplicacion 3*8 es: 24
```

Listado de Entregables:

1. Código fuente de ambas soluciones (Cliente y Servidor)
2. Informe detallado de la implementación (Cliente Gráfico)