



**INSTITUTO POLITÉCNICO NACIONAL  
ESCUELA SUPERIOR DE CÓMPUTO**

**Desarrollo de Sistemas Distribuidos**



# Actividad 18

## Servicio Web SOAP

**PROFESOR:** Pineda Guerrero Carlos

**ALUMNA:** Karina

**GRUPO:** 4CV11

En esta actividad vamos a crear un servicio web SOAP utilizando JDK8.

1. Primeramente, creamos un directorio llamado "ws" en el cual vamos a colocar los programas que vamos a utilizar. Este directorio corresponde al paquete "ws" dónde colocaremos las clases.
2. Vamos a crear el archivo "ServicioWeb.java" en el directorio "ws":

```
package ws;
import javax.ws.WebService;
import javax.ws.WebMethod;
import javax.ws.WebParam;
import java.util.ArrayList;
import java.util.List;
@WebService
public class ServicioWeb
{
    @WebMethod
    public double suma(@WebParam(name="a") double a,@WebParam(name="b") double b)
    {
        return a + b;
    }
    @WebMethod
    public String mayusculas(@WebParam(name="s") String s)
    {
        return s.toUpperCase();
    }
    @WebMethod
    public List<Integer> suma2(@WebParam(name="a") List<Integer> a,@WebParam(name="b")
List<Integer> b)
    {
        List<Integer> c = new ArrayList<Integer>();
        for (int i = 0; i < a.size(); i++)
            c.add(a.get(i) + b.get(i));
        return c;
    }
}
```

En este caso, nuestro servicio web va a incluir las operaciones suma (suma dos números punto flotante), mayúsculas (convierte a mayúsculas una cadena de caracteres) y suyma2 (suma los elementos de dos listas de enteros).

3. Para compilar el servicio web ejecutamos el siguiente comando en el directorio padre del directorio "ws":

javac ws/ServicioWeb.java



4. Para publicar el servicio web utilizamos el programa "Servidor.java" que funciona como servidor de aplicaciones:

```
package ws;
import javax.xml.ws.Endpoint;
public class Servidor
{
    public static void main(String[] args)
    {
        Endpoint.publish("http://localhost:8080/ServicioWeb",new ServicioWeb());
    }
}
```

5. Para compilar el servidor ejecutamos el siguiente comando en el directorio padre del directorio "ws":  
javac ws/Servidor.java



The screenshot shows a Windows Command Prompt window titled "C:\Windows\system32\CMD.exe". The command prompt is open at the directory "C:\Users\Invitado.Kary-PC\Desktop\SISDIS\18.ServiciosWeb". The command "javac ws/Servidor.java" has been entered and executed, resulting in no visible output, indicating successful compilation.

6. Ahora debemos iniciar la ejecución del servidor en otra ventana de cmd en Windows o terminal de Linux o MacOS:

java ws/Servidor



The screenshot shows a Windows Command Prompt window titled "C:\Windows\system32\CMD.exe - java ws/Servidor". The command prompt is open at the directory "C:\Users\Invitado.Kary-PC\Desktop\SISDIS\18.ServiciosWeb". The command "java ws/Servidor" has been entered and executed, resulting in no visible output, indicating successful execution.

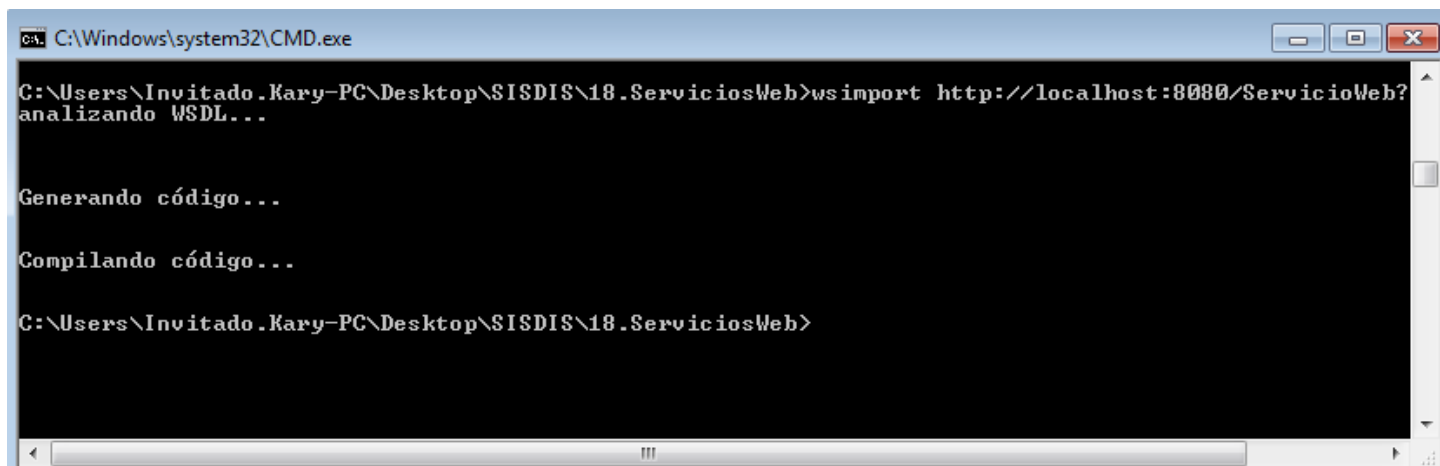
7. Ahora vamos a crea el cliente que consumirá el servicio web. Vamos a utilizar el programa "wsimport" disponible en el JDK8 el cual va a generar las clases que utilizará el cliente.

Al programa "wsimport" se le pasa como parámetro la URL del WSDL del servicio web, en este caso la URL es "<http://localhost:8080/ServicioWeb?wsdl>".

**Nota:** antes de ejecutar el siguiente comando hacer una copia del archivo ServicioWeb.java debido a que wsimport lo borra.

Ejecutar el siguiente comando en el directorio padre del directorio "ws":

wsimport "<http://localhost:8080/ServicioWeb?wsdl>"



```
C:\Windows\system32\CMD.exe

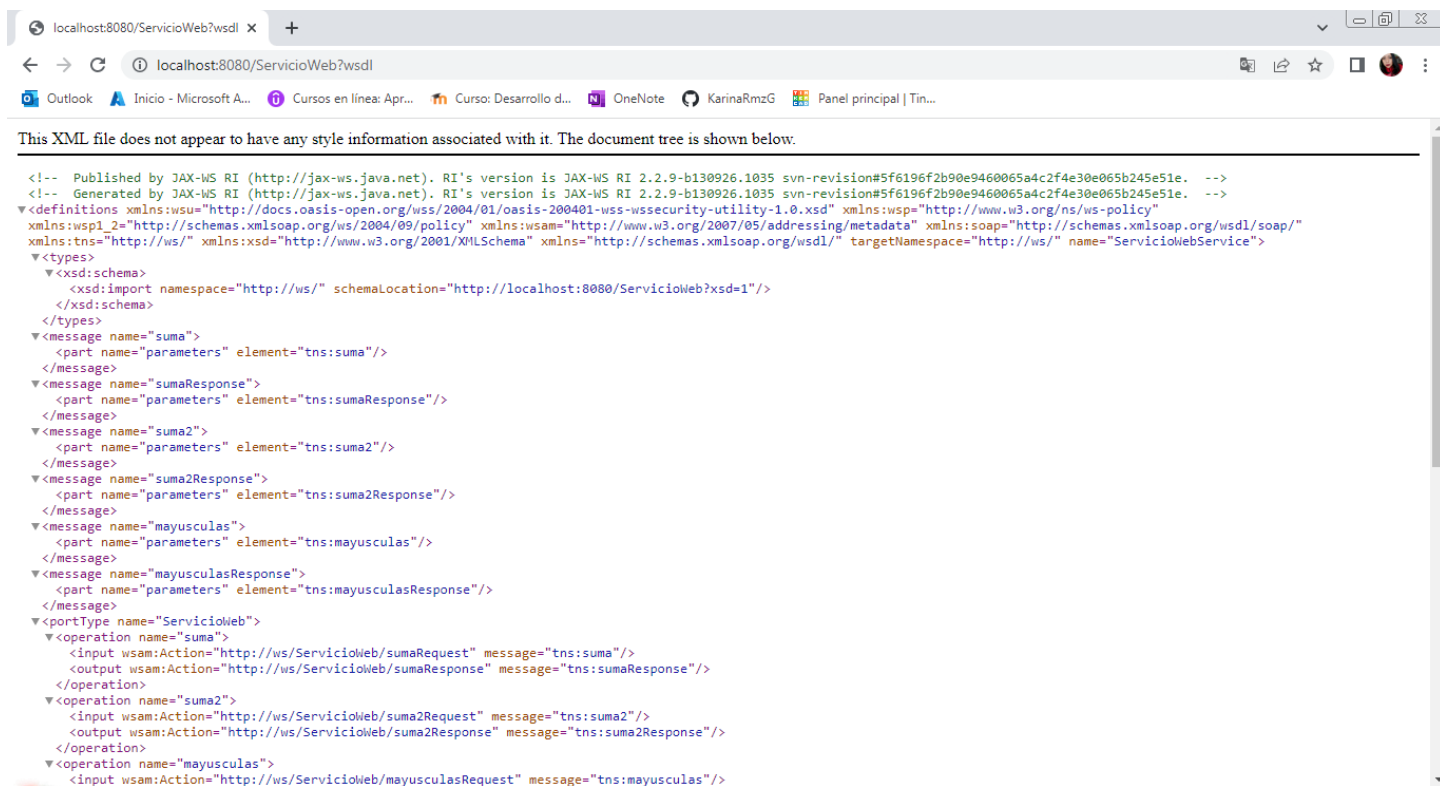
C:\Users\Invitado.Kary-PC\Desktop\SISDIS\18.ServiciosWeb>wsimport http://localhost:8080/ServicioWeb?
analizando WSDL...

Generando código...

Compilando código...

C:\Users\Invitado.Kary-PC\Desktop\SISDIS\18.ServiciosWeb>
```

Para ver el WSDL de nuestro servicio web, se puede ingresar ésta URL en un navegador.  
<http://localhost:8080/ServicioWeb?wsdl>



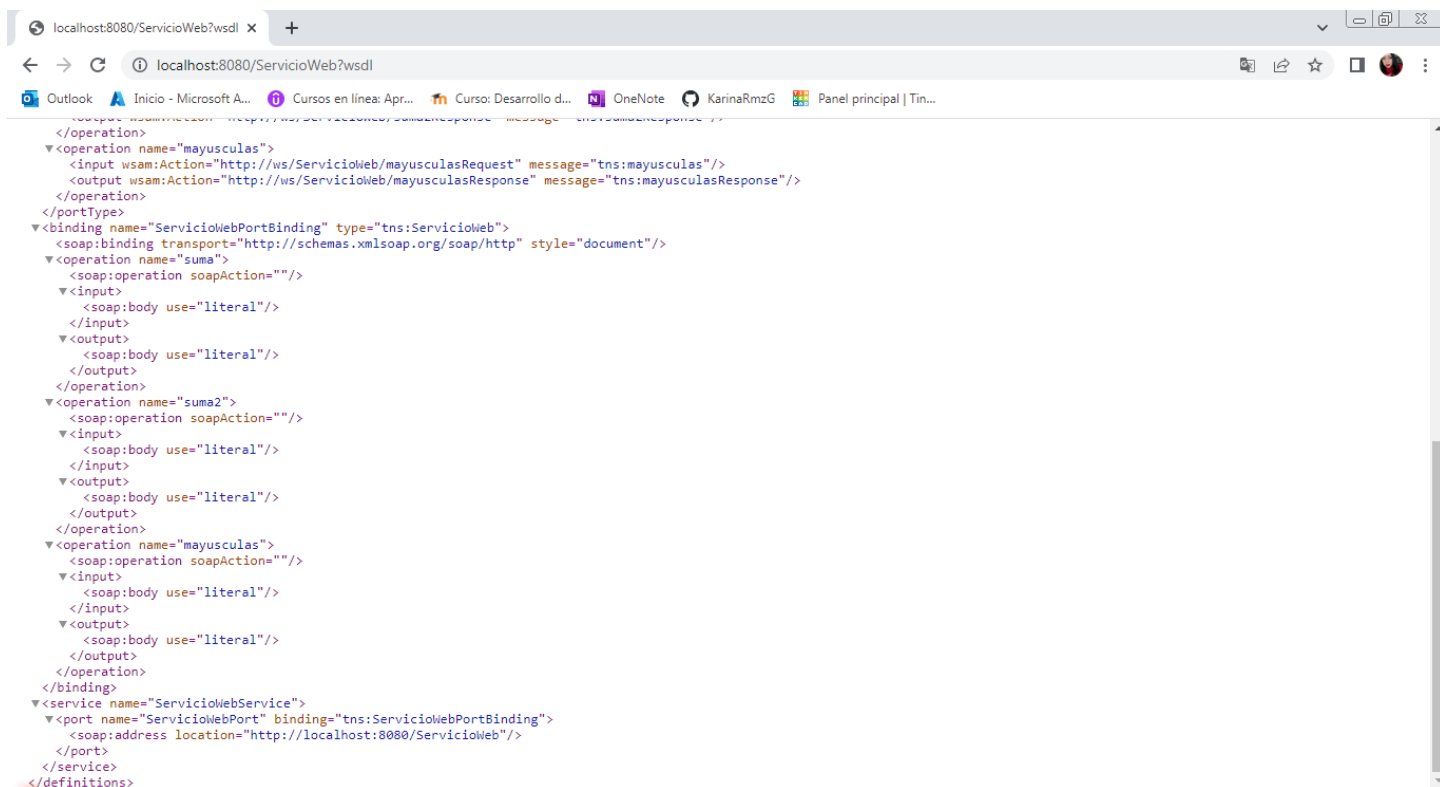
localhost:8080/ServicioWeb?wsdl

localhost:8080/ServicioWeb?wsdl

Outlook Inicio - Microsoft A... Cursos en línea: Apr... Curso: Desarrollo d... OneNote KarinaRmzG Panel principal | Tin...

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 JAX-WS RI 2.2.9-b130926.1035 svn-revision#5f6196f2b90e9460065a4c2f4e30e065b245e51e. -->
<!-- Generated by JAX-WS RI (http://jax-ws.java.net). RI's version is JAX-WS RI 2.2.9-b130926.1035 svn-revision#5f6196f2b90e9460065a4c2f4e30e065b245e51e. -->
<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:wspl_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/soap/"
xmlns:tns="http://ws/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://schemas.xmlsoap.org/wsdl/" targetNamespace="http://ws/" name="ServicioWebService">
  <types>
    <xsd:schema>
      <xsd:import namespace="http://ws/" schemaLocation="http://localhost:8080/ServicioWeb?xsd=1"/>
    </xsd:schema>
  </types>
  <message name="suma">
    <part name="parameters" element="tns:suma"/>
  </message>
  <message name="sumaResponse">
    <part name="parameters" element="tns:sumaResponse"/>
  </message>
  <message name="suma2">
    <part name="parameters" element="tns:suma2"/>
  </message>
  <message name="suma2Response">
    <part name="parameters" element="tns:suma2Response"/>
  </message>
  <message name="mayusculas">
    <part name="parameters" element="tns:mayusculas"/>
  </message>
  <message name="mayusculasResponse">
    <part name="parameters" element="tns:mayusculasResponse"/>
  </message>
  <portType name="ServicioWeb">
    <operation name="suma">
      <input wsam:Action="http://ws/ServicioWeb/sumaRequest" message="tns:suma"/>
      <output wsam:Action="http://ws/ServicioWeb/sumaResponse" message="tns:sumaResponse"/>
    </operation>
    <operation name="suma2">
      <input wsam:Action="http://ws/ServicioWeb/suma2Request" message="tns:suma2"/>
      <output wsam:Action="http://ws/ServicioWeb/suma2Response" message="tns:suma2Response"/>
    </operation>
    <operation name="mayusculas">
      <input wsam:Action="http://ws/ServicioWeb/mayusculasRequest" message="tns:mayusculas"/>
    </operation>
  </portType>
</definitions>
```



8. Vamos a crear el archivo Cliente.java en el directorio "ws":

```
package ws;
import java.net.URL;
import javax.xml.namespace.QName;
import ws.ServicioWeb;
import ws.ServicioWebService;
import java.util.ArrayList;
import java.util.List;
public class Cliente
{
    public static void main(String[] args) throws Exception
    {
        ServicioWebService s = new ServicioWebService(new
        URL("http://localhost:8080/ServicioWeb?wsdl"),
        new
        QName("http://ws/", "ServicioWebService"));
        ServicioWeb obj = s.getServicioWebPort();
        System.out.println(obj.suma(100,200));
        System.out.println(obj.mayusculas("hola"));
        List<Integer> a = new ArrayList<Integer>();
        a.add(1);
        a.add(2);
        a.add(3);
        List<Integer> b = new ArrayList<Integer>();
        b.add(4);
        b.add(5);
        b.add(6);
        List<Integer> c = obj.suma2(a,b);
        for (int i = 0; i < c.size(); i++)
            System.out.println(c.get(i));
    }
}
```

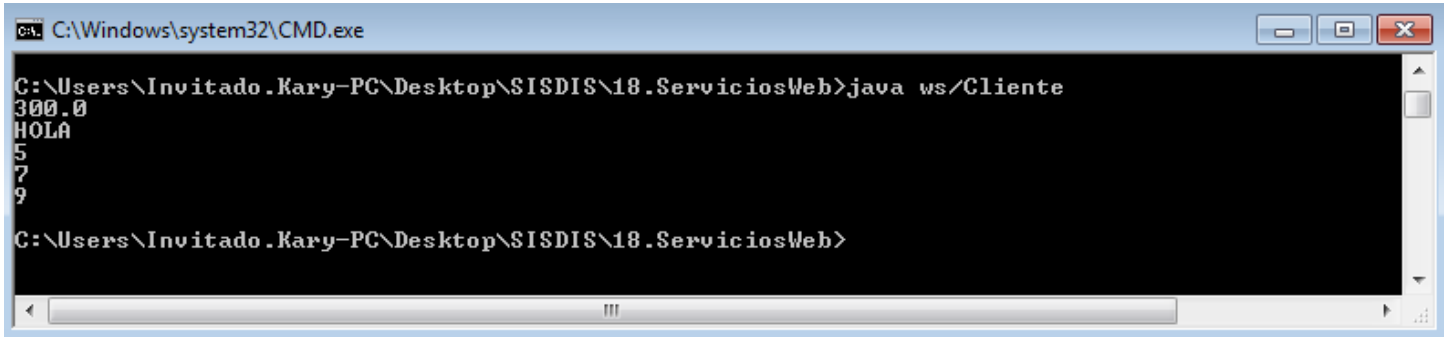
```
}
```

9. Para compilar el cliente ejecutamos el siguiente comando en el directorio padre del directorio "ws":

```
javac ws/Cliente.java
```

10. Para ejecutar el cliente:

```
java ws/Client
```



```
C:\Windows\system32\CMD.exe
C:\Users\Invitado.Kary-PC\Desktop\SISDIS\18.ServiciosWeb>java ws/Cliente
300.0
HOLA
5
7
9
C:\Users\Invitado.Kary-PC\Desktop\SISDIS\18.ServiciosWeb>
```