

SOA

Interoperabilidad WSDL

Profesor: Alejandro Sirvent Llamas

Curso: 2019-2020

- La interoperabilidad es el principio más importante de SOA.
- Como método de implementación de SOA, los **Servicios Web (SW)** **deben ofrecer** importantes **beneficios** de **interoperabilidad**,
 - Deben de **permitir** la **ejecución** de **SW** distribuidos en **múltiples plataformas** de **software** y **arquitecturas** de **hardware**.
- La WS-I (Web Services Interoperability Organization www.ws-i.org) se encarga de promover la interoperabilidad entre SW a partir de definiciones comúnmente aceptadas por la industria y a partir del apoyo a estándares basados en XML.
 - WS-I crea guías y herramientas para ayudar a los desarrolladores a construir SW interoperables.

- Los asuntos de interoperabilidad se manejan con perfiles, tales como WS-I Basic Profile 1.2
 - Incluye XML Schema 1.0, SOAP 1.1, WSDL 1.1 y UDDI 2.0.
 - J2EE y Java se adhieren al WS-I Basic Profile 1.2.
 - Esto asegura no solamente que las aplicaciones desarrolladas podrán ser portadas a través de múltiples implementaciones de J2EE, sino que **los SW implementados con J2EE serán interoperables con aquellos implementados con otras plataformas** que adhieren a los estándares WS-I, tales como Microsoft .NET.

- La clave de la **interoperabilidad** entre servicios web SOAP está en su interfaz **WSDL**.
- El **WS-I Basic Profile** es un **conjunto de especificaciones y buenas prácticas** definidos por la industria para desarrollar SW interoperables independientes de la tecnología con la que fueron desarrollados.
 - Debería ser una referencia, para todos aquellos que tengan que definir un WSDL.



<http://www.ws-i.org/profiles/BasicProfile-1.0-2004-04-16.html>

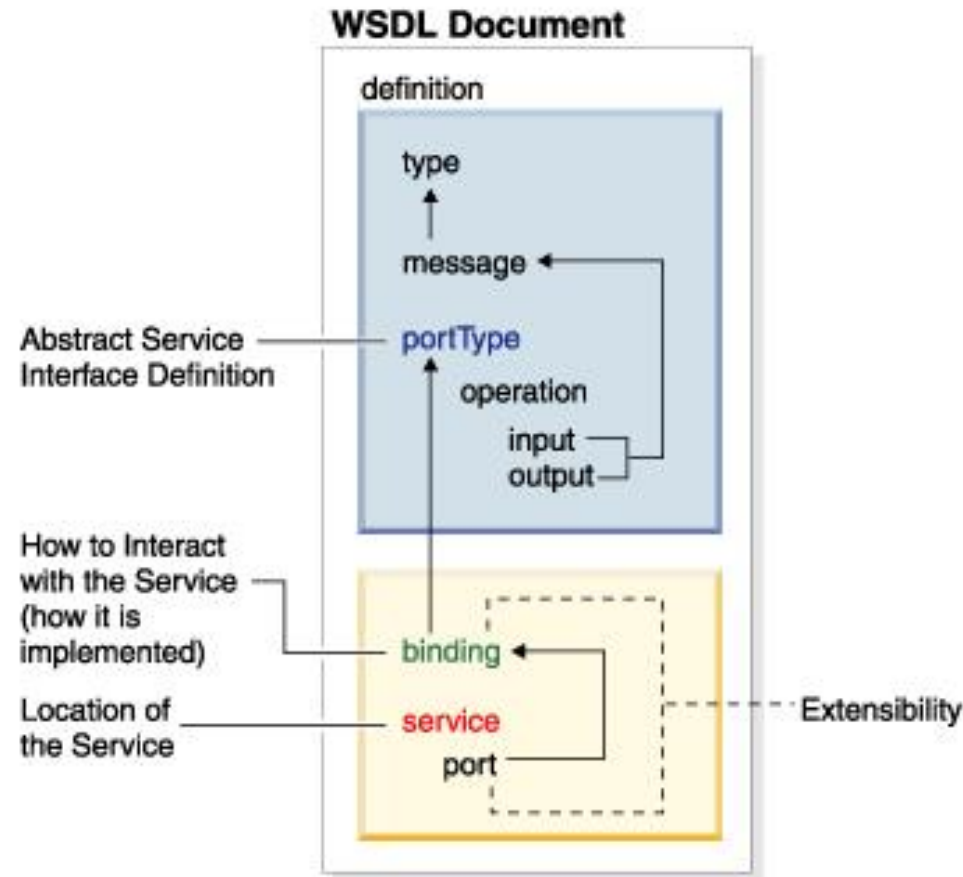
- **Contract First or WSDL First Web Service (Top-Down Approach)**
 - En Contract First approach , tomamos como primer paso, la creación de contratos wsdl, seguido por la lógica de negocio de servicio en el código.
 - En la primera aproximación del contrato, el enfoque es la creación del Schema asociado al **wsdl**.
 - WSDL y el Schema, tienen que definir:
 - el formato del mensaje,
 - operaciones,
 - nombre de la interfaz,
 - y otra información para el servicio web.

- **Contract First or WSDL First Web Service (Top-Down Approach)**

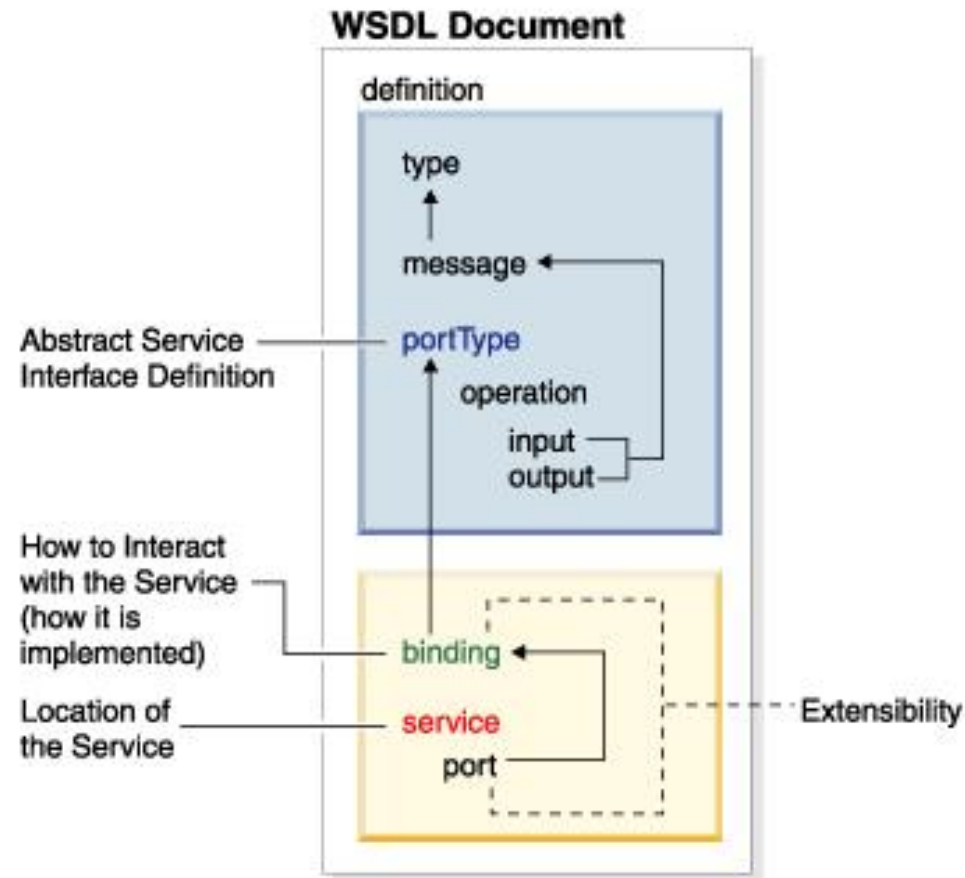
- Tras crear el WSDL y el Schema, podemos crear un SW utilizando algún framework adecuado (eclipse, netbeans,...).
 - Los principales frameworks de SW permiten la generación de servicios a partir del WSDL.
 - » El código de todo el servicio es generado por el framework, quedando únicamente por implementar la lógica de negocio.

- **¿Qué es WSDL?**
- ***El Web Services Description Language o WSDL:***
 - Describe una interfaz de SW.
 - Se trata de mensajes que se intercambian entre el cliente y el servidor.
 - Los mensajes se describen de forma abstracta y luego unidos a un protocolo de red concreto y formato de mensaje.
 - Las definiciones de SW se pueden asignar a :
 - Cualquier lenguaje de implementación,
 - Plataforma,
 - Modelo de objetos,
 - Sistema de mensajería.

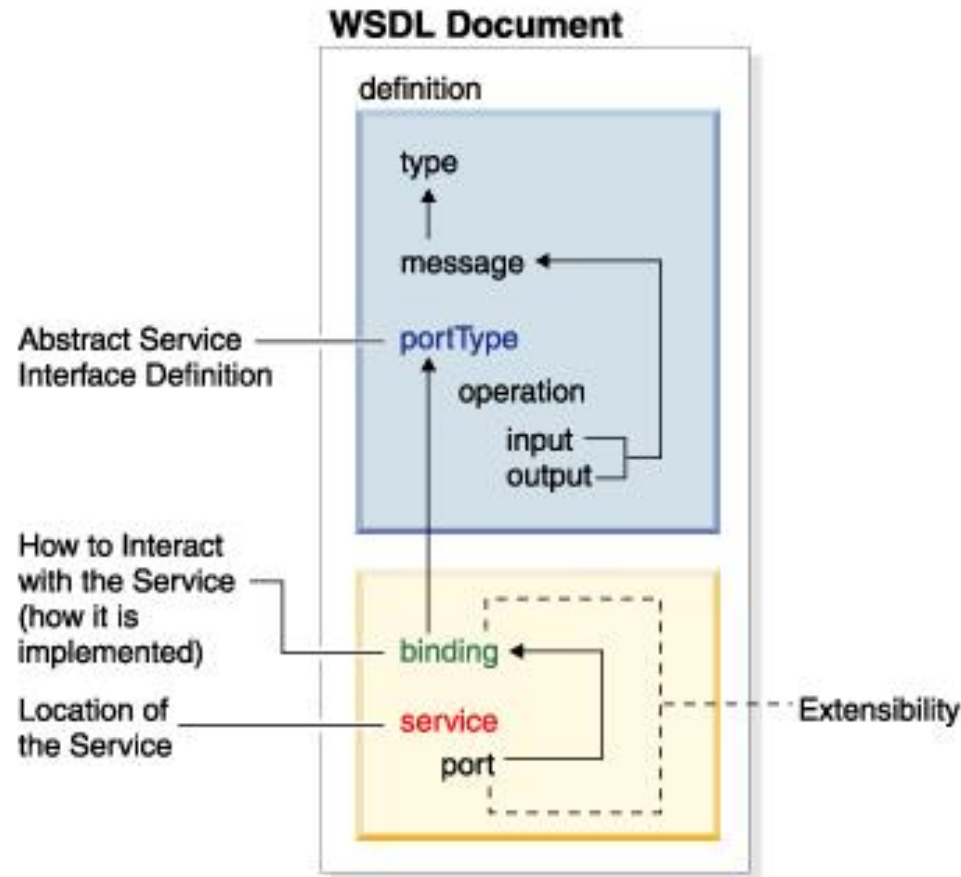
- **WSDL Elementos**
- **Definition**
 - Este contiene el nombre del atributo, que a su vez cuenta con el **nombre del servicio web**.
- **Types**
 - Este elemento utiliza el lenguaje de **XML Schema** para declarar **tipos de datos complejos** y los elementos que se utilizan en otras partes del documento WSDL.



- **WSDL Elementos**
- **Message**
 - Este elemento describe la **carga útil del mensaje** utilizando XML Schema **con construcciones de tipos** y tipos complejos.
- **PortType/Interface and Operation**
 - Estos elementos **describen la interfaz** que define un **SW** y sus **métodos**.



- **WSDL Elementos**
- ***Binding***
 - Este elemento **asigna** un **portType** y sus elementos de operación para un **protocolo determinado** (por ejemplo, SOAP 1.1) y estilo de codificación.
- ***Service***
 - Este elemento es el **responsable** de la **asignación de una dirección de Internet a una unión específica**.



- Pasos a realizar en eclipse:

1. Crear el fichero WSDL, mediante el asistente de eclipse.
2. Generar interfaz y código, utilizando el motor de servicio de WSDL.
3. Añadir la lógica de negocio mediante la implementación de la interfaz del servicio.
4. Exposición de la interfaz con **la lógica de negocio** como servicios web.
5. El Servicio ya está disponible para el cliente, para comprobarlo podemos usar la herramienta **Soap UI**.

MTIS

- Crear Web

Creación WS

New Dynamic Web Project

Dynamic Web Project

Create a standalone Dynamic Web project or add it to a new or existing Enterprise Application.

Project name: proyectoConcatenacion

Project location

☒ Use default location

Location: C:\Users\Alex\workspace\proyectoConcatenacion Browse...

Target runtime

Apache Tomcat v7.0 New Runtime...

Dynamic web module version

2.5

Configuration

< custom> Modify...

Hint: Get started quickly by selecting one of the pre-defined project configurations.

EAR membership

☐ Add project to an EAR

EAR project name: proyectoConcatenacionEAR New Project...

Working sets

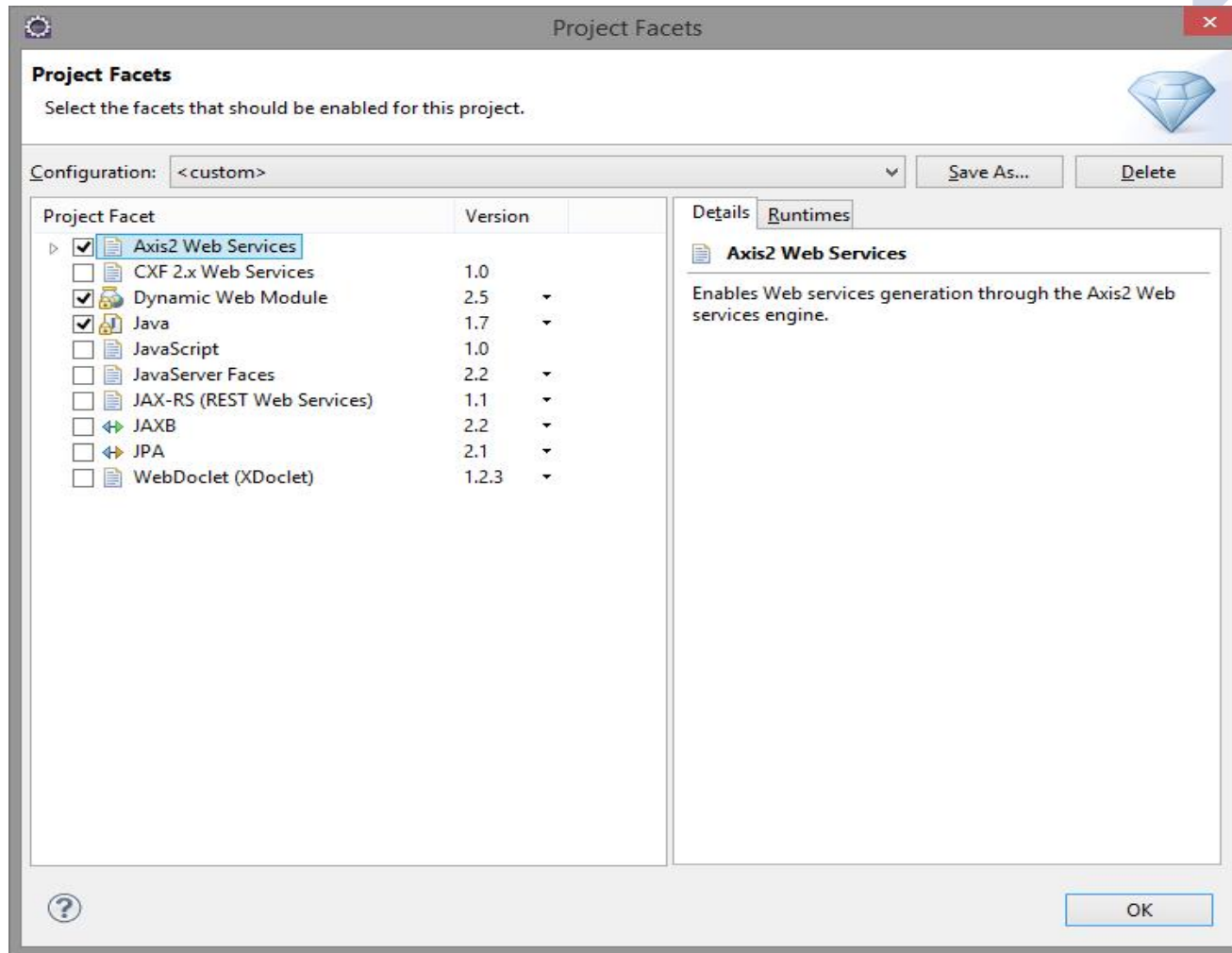
☐ Add project to working sets

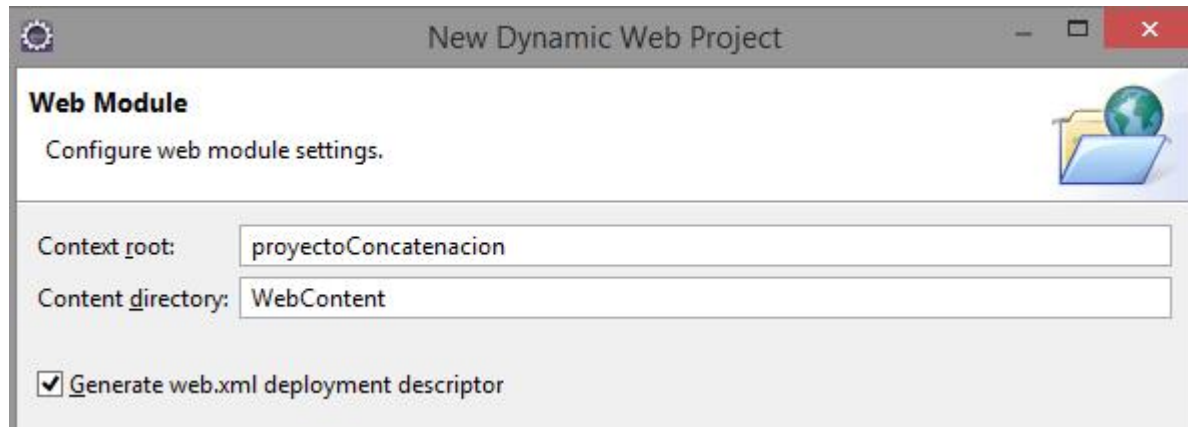
Working sets: Select...

? < Back Next > Finish Cancel

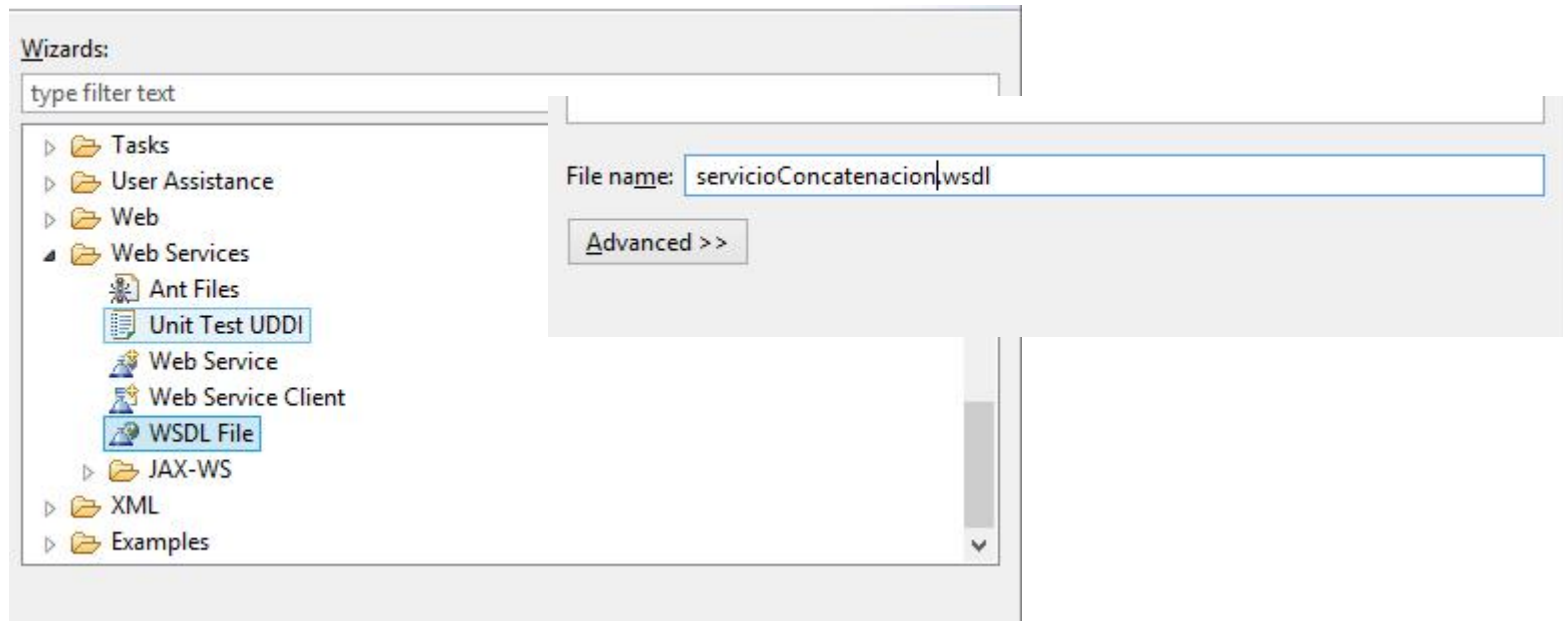
Other –







- Ahora Creamos el WSDL, mediante:
 - File – new –other – web services → WSDL File



Options
Specify the attributes for the new WSDL file.

Target namespace:

Prefix:

☒ Create WSDL Skeleton

Protocol:

SOAP Binding Options

☒ document literal

☐ rpc literal

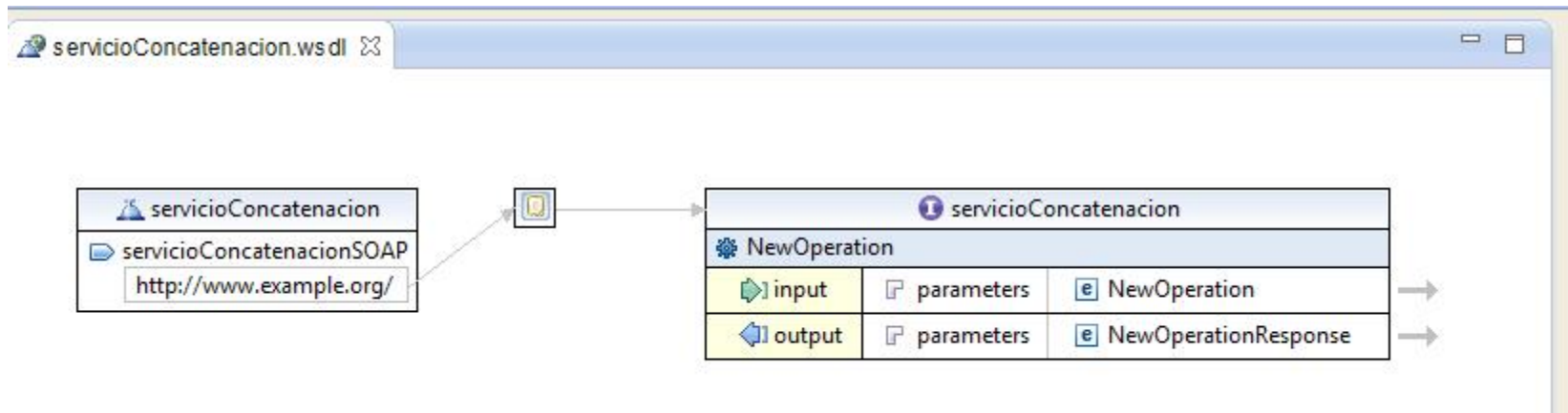
☐ rpc encoded

[Modify project compliance setting](#)

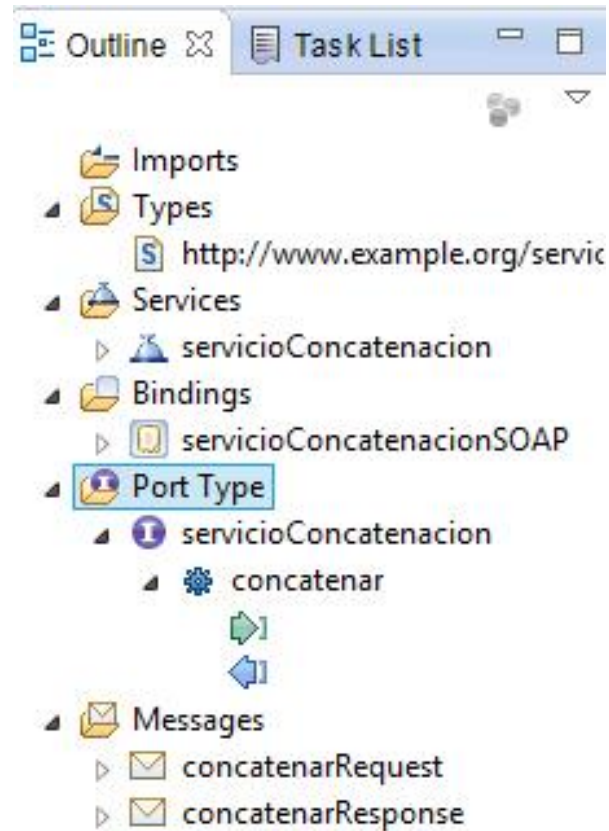
[Modify WSDL Files preferences](#)

? < Back Next > **Finish** Cancel

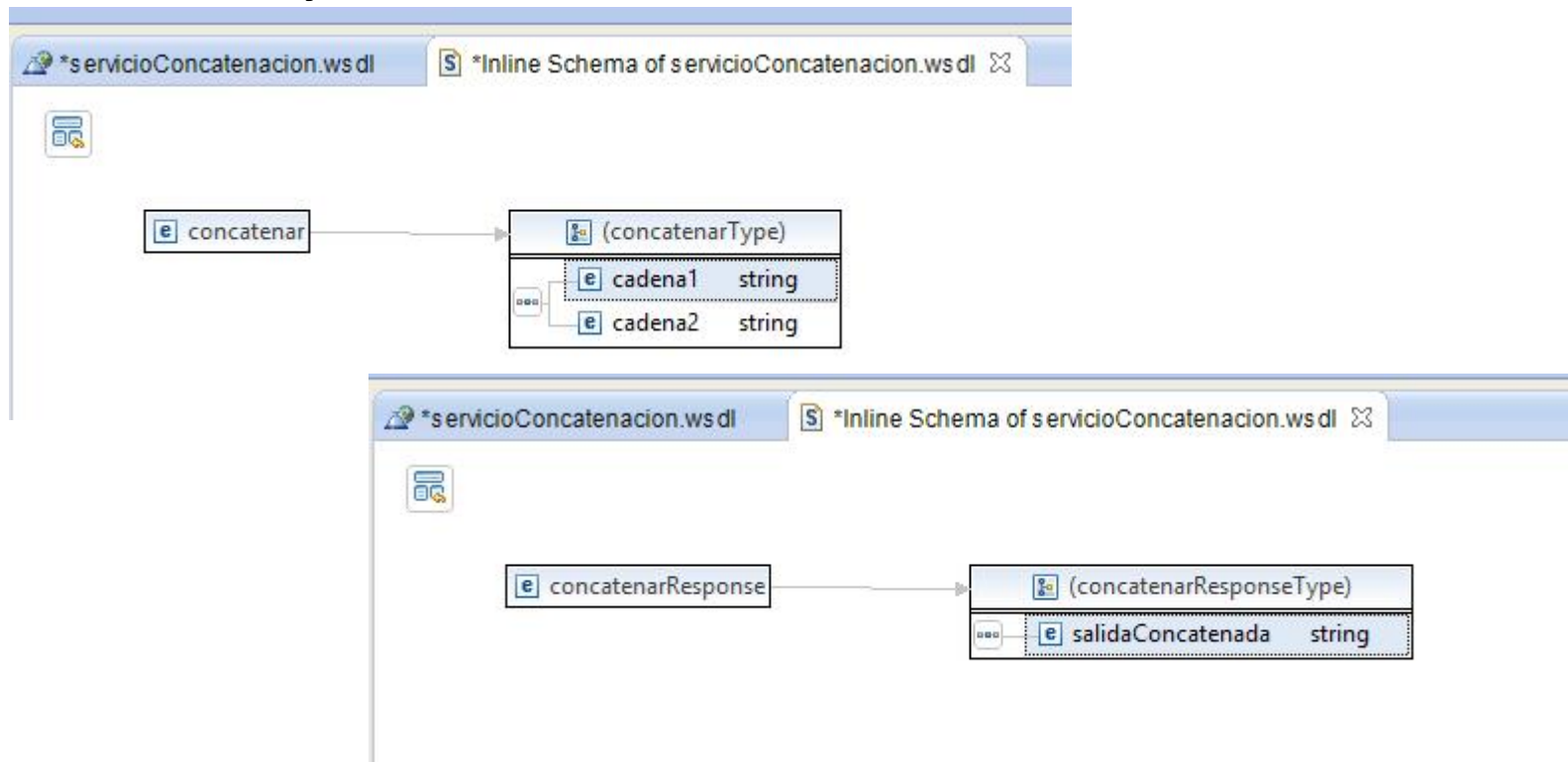
- Vista del servicio WSDL:



- Vista del outline:



- Modifico nombre operación, parámetros entrada y salida.



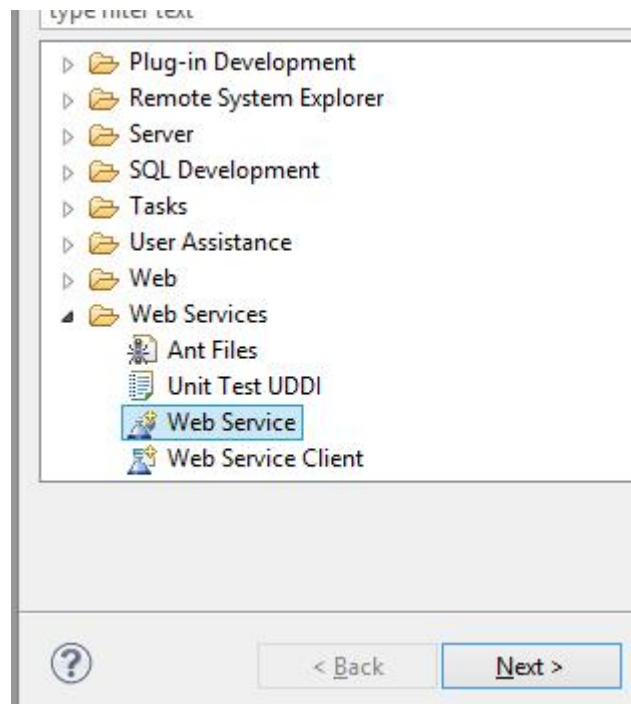
- Posibilidad de ver el código wsdl.



```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://www.example.org/se
<wsdl:types>
  <xsd:schema targetNamespace="http://www.example.org/servicioConcatenacion/">
    <xsd:element name="concatenar">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="cadena1" type="xsd:string" />
          <xsd:element name="cadena2" type="xsd:string"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="concatenarResponse">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="salidaConcatenada" type="xsd:string"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:schema>
</wsdl:types>
<wsdl:message name="concatenarRequest">
```

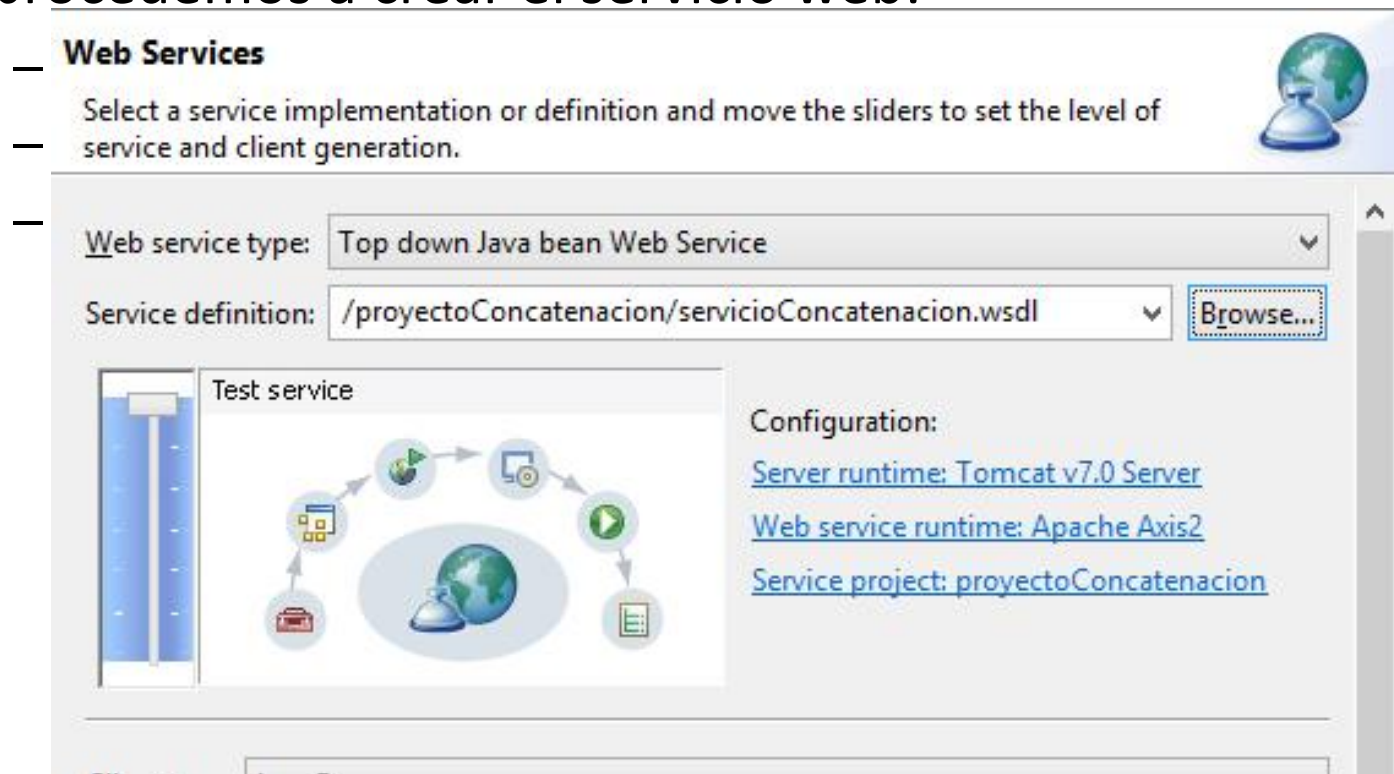
Creación servicio web con eclipse

- Una vez creado el WSDL, mediante un método u otro, procedemos a crear el servicio web:
- File – new – other – web services → Web service



- Una vez creado el WSDL, mediante un método u otro, procedemos a crear el servicio web:
 - File – new – other – web services → Web service
 - Web service type → Top down
 - Service definition →
/proyectoConcatenacion/servicioConcatenacion.wsdl

- Una vez creado el WSDL, mediante un método u otro, procedemos a crear el servicio web:



- Creamos el servicio web y añadimos la lógica de negocio (para el ejemplo de la concatenación):

```
String cadena1 = concatenar.getCadena1 ();  
String cadena2 = concatenar.getCadena2();
```

```
String cadenaConcatenada = new  
StringBuilder(cadena1).append(cadena2).toString();
```

```
ConcatenarResponse cs = new ConcatenarResponse();  
cs.setSalidaConcatenada(cadenaConcatenada);  
return cs;
```

- Finalmente comprobamos el resultado:

The screenshot displays the SoapUI 5.3.0 interface. At the top, a 'New SOAP Project' dialog box is open, showing the project name 'HolaMundo' and the initial WSDL URL 'tp://localhost:9080/HolaMundoSW/services/HolaMundo?wsdl'. The 'Create Requests' checkbox is checked. Below the dialog, the main SoapUI window is visible, showing the 'SoapUI Starter Page' with a 'Request 1' tab. The 'Request 1' tab shows the XML structure of the SOAP request, including the envelope, header, body, and the 'hol:NewOperation' element. The URL bar shows 'http://localhost:9080/HolaMundoSW/services/HolaMundo/'.

New SOAP Project

New SOAP Project
Creates a WSDL/SOAP based Project in this workspace

Project Name: HolaMundo

Initial WSDL: tp://localhost:9080/HolaMundoSW/services/HolaMundo?wsdl Browse...

Create Requests: ☒ Create sample requests for all operations?

Create TestSuite: ☐ Creates a TestSuite for the imported WSDL

Relative Paths: ☐ Stores all file paths in project relatively to project file (requires save)

OK Cancel

SoapUI 5.3.0

File Project Suite Case Step Tools Desktop Help

Empty SOAP REST Import Save All Forum Trial Preferences Proxy

Search Forum

Onl

Navigator

Projects

HolaMundo

HolaMundoSOAP

NewOperation

Request 1

SoapUI Starter Page

Request 1

http://localhost:9080/HolaMundoSW/services/HolaMundo/

XML

```
<?xml version='1.0' encoding='UTF-8'>
<soapenv:Envelope xmlns:soapenv='http://schemas.xmlsoap.org/soap/envelope/'>
  <soapenv:Header/>
  <soapenv:Body>
    <hol:NewOperation>
      <in>Pepe</in>
    </hol:NewOperation>
  </soapenv:Body>
</soapenv:Envelope>
```

Raw

XML

```
<?xml version='1.0' encoding='UTF-8'>
<soapenv:Envelope xmlns:soapenv='http://schemas.xmlsoap.org/soap/envelope/'>
  <soapenv:Header/>
  <soapenv:Body>
    <ns1:NewOperationResponse xmlns:ns1='http://www.example.org/HolaMundo/'>
      <out>Pepe</out>
    </ns1:NewOperationResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Raw