

# SISTEMAS ORIENTADOS A SERVICIOS

Graduado en Ingeniería Informática

## Tema 3 – Ejemplos.

Marta Patiño

[mpatino@fi.upm.es](mailto:mpatino@fi.upm.es)

## Tema 3 - Ejemplos

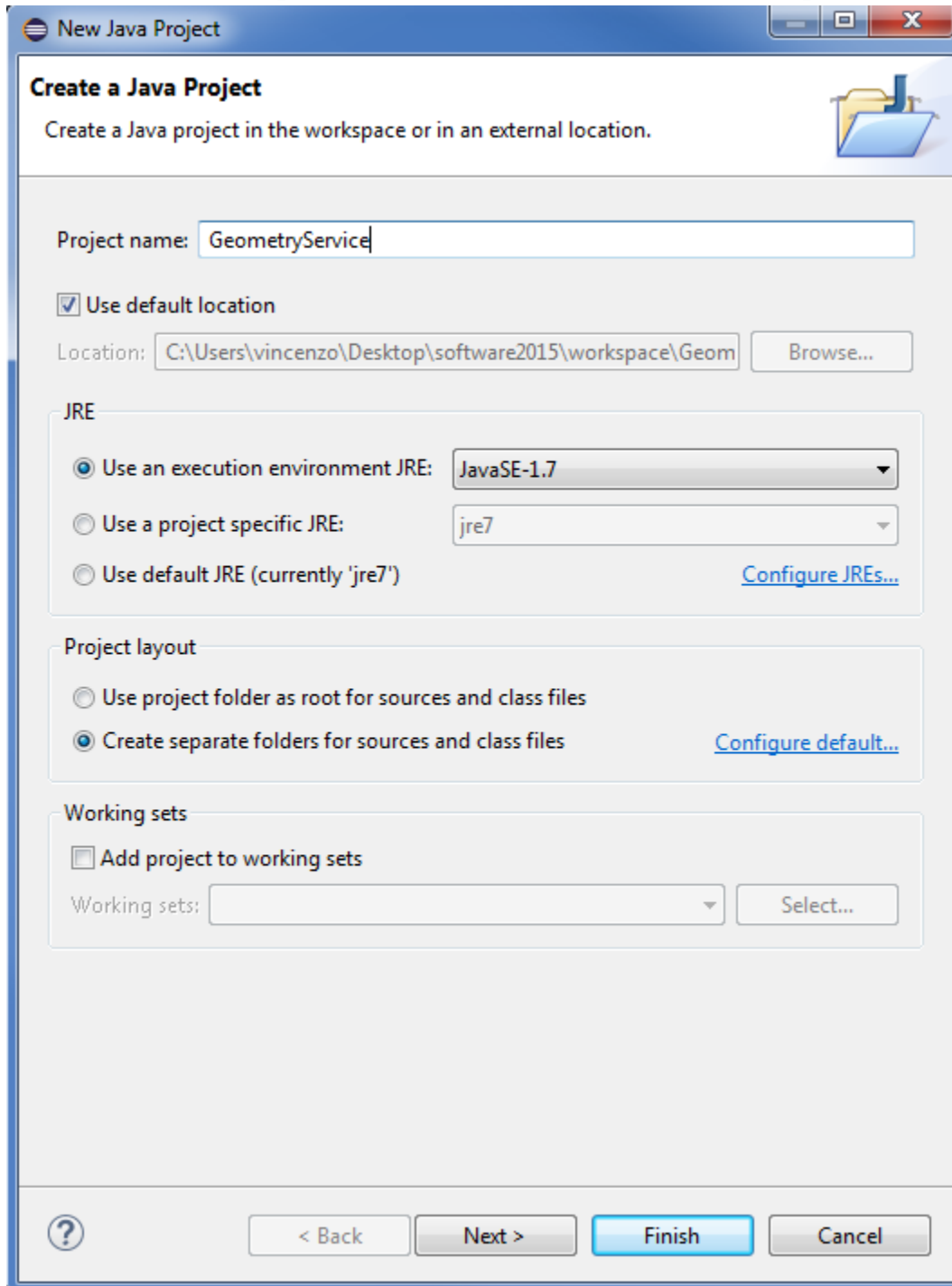
[1]	<i>GeometryService</i> : Generar un servicio a partir de una clase java. ....	3
A.	Crear un nuevo proyecto java ("Java Project") en Eclipse .....	4
B.	Crear la clase java que implemente el servicio .....	4
C.	Generar WSDL.....	5
D.	Crear el archivo services.xml con la descripción del servicio. ....	5
E.	Crear el paquete del servicio .....	6
F.	Despliegue del servicio en Tomcat .....	7
G.	Comprobar en la lista de servicios de axis que nuestro nuevo servicio está disponible .....	8
H.	Invocación del servicio desde el navegador .....	9
I.	Creación de un cliente Java.....	10
[2]	<i>Calculator</i> : Generar un servicio web a partir del fichero WSDL. ....	14
A.	Crear un nuevo proyecto java ("Java Project") en Eclipse. ....	14
B.	Copiar el archivo Calculator.wsdl en la raíz del proyecto CalculatorService .....	15
C.	Generar el <i>skeleton</i> del servicio .....	16
D.	Implementar el servicio .....	17
E.	Crear el paquete de despliegue .....	19
F.	Despliegue del servicio en Tomcat .....	19
G.	Comprobar en la lista de servicios de axis que nuestro nuevo servicio está disponible .....	20
H.	Creación de un cliente Java.....	20

## [1] *GeometryService*: Generar un servicio a partir de una clase java.

Se dispone de una clase java (*GeometryService*) que tiene un método (*circleArea*) que dado el radio de un círculo, devuelve la superficie del mismo. Se quiere generar un servicio web para esa clase empleando Axis2.

A continuación se muestra cómo se genera el servicio web y un cliente de ese servicio.

## A. Crear un nuevo proyecto java ("Java Project") en Eclipse



## B. Crear la clase java que implemente el servicio

GeometryService.java

```

1 package sos.t3.a31.demo;
2
3 public class GeometryService {
4
5     public double circleArea(double radius){
6         return Math.PI * radius * radius;
7     }
8
9 }

```

## C. Generar WSDL

Desde un terminal en la carpeta raíz del proyecto generar el fichero con el wsdl del servicio empleando la herramienta *java2wsdl* de axis2. Las opciones necesarias son:

- wv → Versión WSDL (1.0 o 2.0)
- of → Nombre fichero WSDL
- cn → Nombre clase a traducir
- cp → Classpath

### LINUX

```
$AXIS2_HOME/bin/java2wsdl.sh -cp ./bin -cn sos.t3.a31.demo.GeometryService -wv 2.0 -of
GeometryService.wsdl
```

```

valerio@lsd15:~/lectures/workspace/GeometryService$ $AXIS2_HOME/bin/java2wsdl.sh -cp ./bin -cn
sos.t3.a31.demo.GeometryService -wv 2.0 -of GeometryService.wsdl
Using AXIS2_HOME: /home/valerio/lectures/software2015_linux/axis2-1.6.2
Using JAVA_HOME: /home/valerio/lectures/software2015_linux/jdk1.7.0_75
[INFO] WSDL created at /home/valerio/lectures/workspace/GeometryService/GeometryService.wsdl

```

### WINDOWS

```
%AXIS2_HOME%\bin\java2wsdl.bat -cp .\bin -cn sos.t3.a31.demo.GeometryService -wv 2.0 -of
GeometryService.wsdl
```

```

C:\Users\vincenzo\Desktop\software2015\workspace\GeometryService>%AXIS2_HOME%\bin\java2wsdl.bat -cp .\bin -cn sos.t3.a31.de
ometryService -wv 2.0 -of GeometryService.wsdl
Using AXIS2_HOME: C:\Users\vincenzo\Desktop\software2015\axis2-1.6.2
Using JAVA_HOME: C:\Users\vincenzo\Desktop\software2015\jdk1.7.0_75
log4j:WARN No appenders could be found for logger <org.apache.axis2.util.Loader>.
log4j:WARN Please initialize the log4j system properly.

```

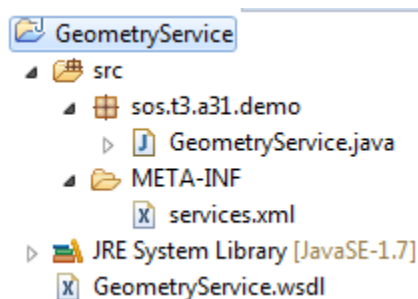
## D. Crear el archivo services.xml con la descripción del servicio.

Crear una carpeta META-INF dentro de la carpeta src del proyecto.

Crear un archivo services.xml dentro de META-INF con los siguientes datos:

```
<service name="GeometryService" scope="application">
    <description>Sample Geometry Service</description>
    <messageReceivers>
        <messageReceiver mep="http://www.w3.org/2004/08/wsdl/in-only"
            class="org.apache.axis2.rpc.receivers.RPCInOnlyMessageReceiver" />
        <messageReceiver mep="http://www.w3.org/2004/08/wsdl/in-out"
            class="org.apache.axis2.rpc.receivers.RPCMessageReceiver" />
    </messageReceivers>
    <parameter name="ServiceClass">
        sos.t3.a31.demo.GeometryService
    </parameter>
</service>
```

La estructura del proyecto actual es:



## E. Crear el paquete del servicio

En este primer servicio creamos el paquete del servicio utilizando el comando *jar*

### LINUX

```
jar -cf GS.aar -C ./bin ./
```

```
valerio@lsd15:~/lectures/workspace/GeometryService$ jar -cf GS.aar -C ./bin ./
valerio@lsd15:~/lectures/workspace/GeometryService$ ls
bin  GeometryService.wsdl  GS.aar  src
```

### WINDOWS

```
jar -cf GS.aar -C .\bin .\ cd I
```

```
C:\Users\vincenzo\Desktop\software2015\workspace\GeometryService>jar -cf GS.aar .\bin .\
C:\Users\vincenzo\Desktop\software2015\workspace\GeometryService>dir
Volume in drive C has no label.
Volume Serial Number is 601A-45D1

Directory of C:\Users\vincenzo\Desktop\software2015\workspace\GeometryService

02/20/2015  04:20 PM    <DIR>          .
02/20/2015  04:20 PM    <DIR>          ..
02/20/2015  03:40 PM             301 .classpath
02/20/2015  03:40 PM             391 .project
02/20/2015  03:40 PM    <DIR>          .settings
02/20/2015  04:07 PM    <DIR>          bin
02/20/2015  03:56 PM             3,410 GeometryService.wsdl
02/20/2015  04:20 PM             4,944 GS.aar
02/20/2015  04:07 PM    <DIR>          src
               4 File(s)              9,046 bytes
```

## F. Despliegue del servicio en Tomcat

Copiar el archivo GS.aar en

### LINUX

\$CATALINA\_HOME/webapps/axis2/WEB-INF/services/

### WINDOWS

%CATALINA\_HOME%\webapps\axis2\WEB-INF\services\

## G. Comprobar en la lista de servicios de axis que nuestro nuevo servicio está disponible

<http://localhost:8080/axis2/services/listServices>

---



### Available services

#### Version

Service Description : Version

Service EPR : <http://localhost:8080/axis2/services/Version>

Service Status : Active

*Available Operations*

- getVersion

#### GeometryService

Service Description : Geometry Service

Service EPR : <http://localhost:8080/axis2/services/GeometryService>

Service Status : Active

*Available Operations*

- circleArea



## H. Invocación del servicio desde el navegador

En el navegador invocamos el método **circleArea** del servicio **GeometryService** con el parámetro **radius=2.0**

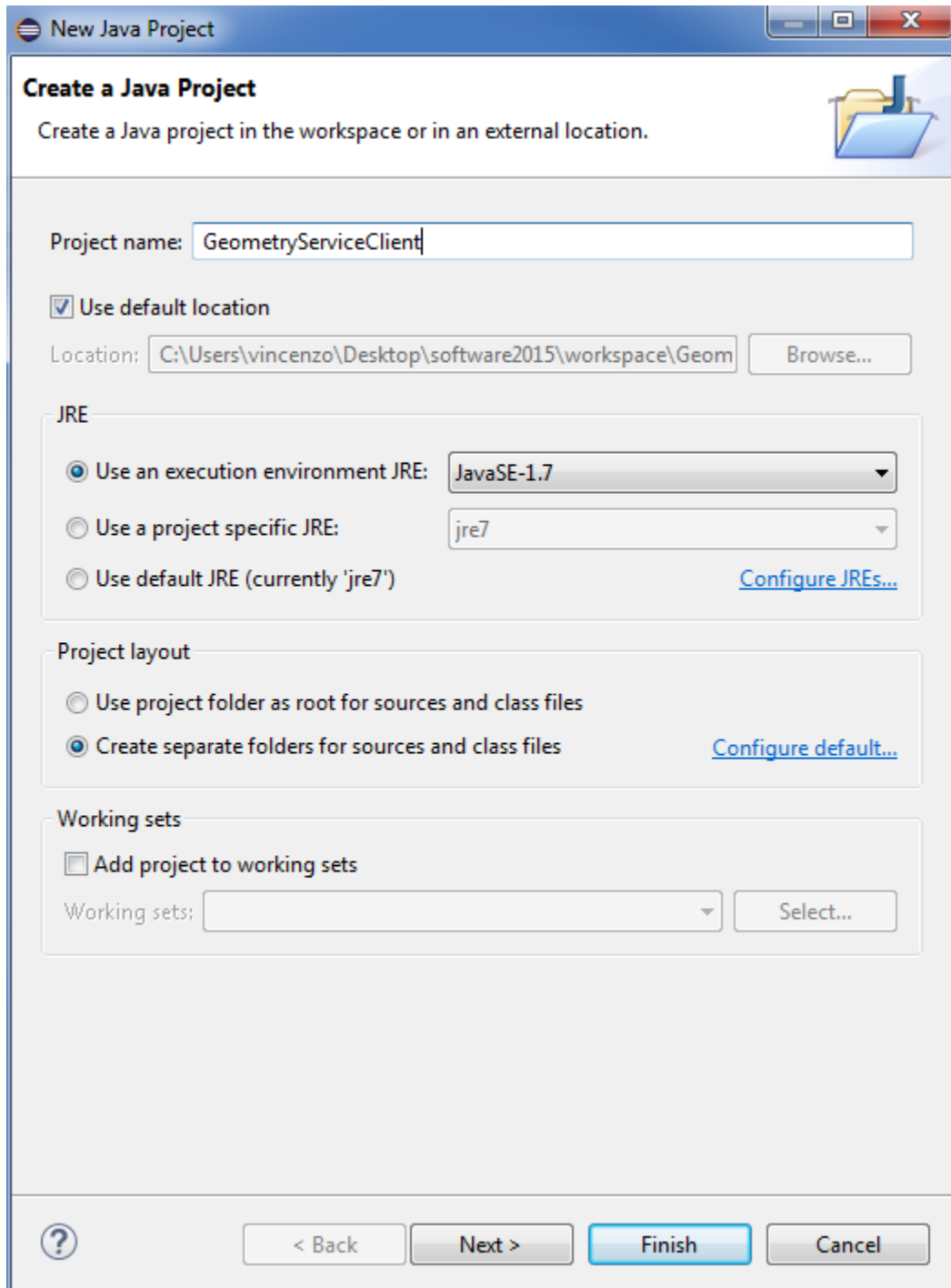
 localhost:8080/axis2/services/GeometryService/circleArea?radius=2.0

sta es la respuesta:

```
<ns:circleAreaResponse xmlns:ns="http://demo.a31.t3.sos">
  <ns:return>12.566370614359172</ns:return>
</ns:circleAreaResponse>
```

## I. Creación de un cliente Java

Crear un nuevo “Java Project” en Eclipse, *GeometryServiceClient*.



The screenshot shows the 'New Java Project' dialog box in the Eclipse IDE. The dialog has a title bar 'New Java Project' and a subtitle 'Create a Java Project'. Below the subtitle is the instruction 'Create a Java project in the workspace or in an external location.' The main area contains several sections: 'Project name' with a text field containing 'GeometryServiceClient'; 'Use default location' checked, with a 'Location' field showing 'C:\Users\vincenzo\Desktop\software2015\workspace\Geom' and a 'Browse...' button; 'JRE' section with three radio buttons: 'Use an execution environment JRE:' (selected) with a dropdown showing 'JavaSE-1.7', 'Use a project specific JRE:' with a dropdown showing 'jre7', and 'Use default JRE (currently 'jre7')' with a 'Configure JREs...' link; 'Project layout' section with two radio buttons: 'Use project folder as root for sources and class files' and 'Create separate folders for sources and class files' (selected) with a 'Configure default...' link; and 'Working sets' section with a checkbox 'Add project to working sets' and a 'Working sets:' dropdown with a 'Select...' button. At the bottom are buttons for '?', '< Back', 'Next >', 'Finish', and 'Cancel'.

**New Java Project**

Create a Java Project

Create a Java project in the workspace or in an external location.

Project name:

☒ Use default location

Location:  [Browse...](#)

JRE

☒ Use an execution environment JRE:

☐ Use a project specific JRE:

☐ Use default JRE (currently 'jre7') [Configure JREs...](#)

Project layout

☐ Use project folder as root for sources and class files

☒ Create separate folders for sources and class files [Configure default...](#)

Working sets

☐ Add project to working sets

Working sets:  [Select...](#)

? < Back Next > Finish Cancel

Copiar el fichero WSDL generado en el punto C en la raíz de este nuevo proyecto

Generar los *stubs* para llamar al servicio. Se usa la herramienta *wsdl2java* de axis2.

## LINUX

```
$AXIS2_HOME/bin/wsdl2java.sh -uri GeometryService.wsdl -vv 2.0 -p sos.t3.a31.demo.client  
-d adb
```

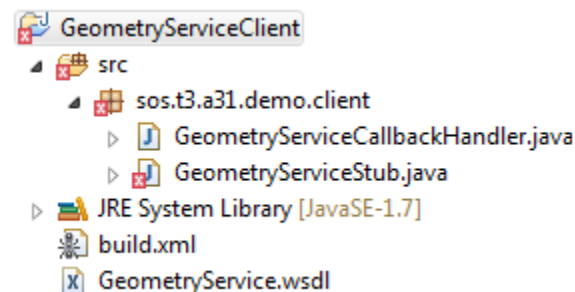
```
valerio@lsd15:~/lectures/workspace/GeometryServiceClient$ $AXIS2_HOME/bin/wsdl2java.sh -uri ./G  
eometryService.wsdl -vv 2.0 -p sos.t3.a31.demo.client -d adb  
Using AXIS2_HOME: /home/valerio/lectures/software2015_linux/axis2-1.6.2  
Using JAVA_HOME: /home/valerio/lectures/software2015_linux/jdk1.7.0_75  
[WARN] Woden[Warning],0:0,Description-1001,The targetNamespace 'http://demo.a31.t3.sos' is not  
dereferencable.
```

## WINDOWS

```
%AXIS2_HOME%\bin\wsdl2java.bat -uri GeometryService.wsdl -vv 2.0 -p sos.t3.a31.demo.client -d  
adb
```

```
C:\Users\vincenzo\Desktop\software2015\workspace\GeometryServiceClient>%AXIS2_HOME%\bin\wsdl2java.bat -uri GeometryService.wsdl -v  
v 2.0 -p sos.t3.a31.demo.client -d adb  
Using AXIS2_HOME: C:\Users\vincenzo\Desktop\software2015\axis2-1.6.2  
Using JAVA_HOME: C:\Users\vincenzo\Desktop\software2015\jdk1.7.0_75  
log4j:WARN No appenders could be found for logger (org.apache.axis2.description.WSDL2ToAxisServiceBuilder).  
log4j:WARN Please initialize the log4j system properly.  
C:\Users\vincenzo\Desktop\software2015\workspace\GeometryServiceClient>
```

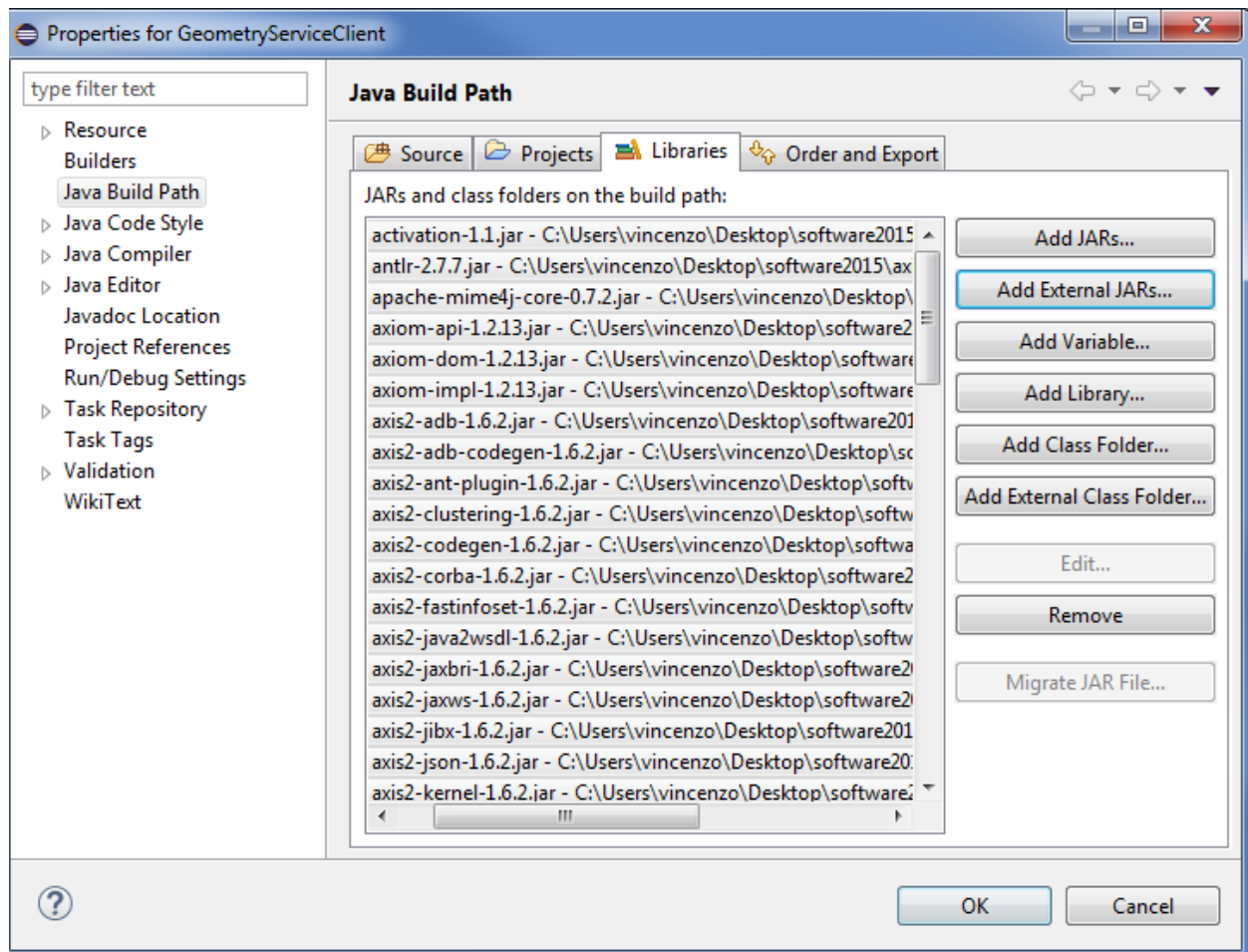
La herramienta añadirá en la carpeta src de Eclipse los stubs generados.



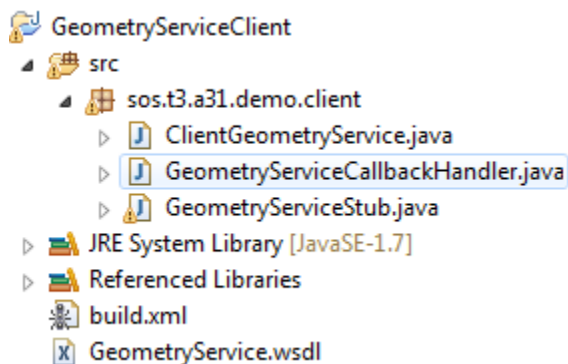
Antes de seguir con el desarrollo de nuestro cliente, añadimos las librerías de axis2 a nuestro proyecto.

Botón derecho en el proyecto -> Build path -> Configure Build Path -> Libraries -> Add External JARs

y añadir todas las librerías de la carpeta lib de la versión de axis2-1.6.2



Ahora creamos una nueva clase *ClientGeometryService.java* de manera que nuestro proyecto queda así:



Código del *GeometryServiceClient*:

```
package sos.t3.a31.demo.client;

import java.rmi.RemoteException;

import sos.t3.a31.demo.client.GeometryServiceStub.CircleArea;
import sos.t3.a31.demo.client.GeometryServiceStub.CircleAreaResponse;

public class ClientGeometryService {

    public static void main(String[] args) throws RemoteException {
        GeometryServiceStub stub = new GeometryServiceStub();
        CircleArea circleArea = new CircleArea();
        double radius = 2.0;
        circleArea.setRadius(radius);
        CircleAreaResponse response = stub.circleArea(circleArea);
        double area = response.get_return();
        System.out.println("Radius: " + radius + ", Area: " + area);
    }
}
```

Ejecutando el código en Eclipse se obtiene la salida:

```
log4j:WARN No appenders could be found for logger (org.apache.axis2.description.AxisOperation).
log4j:WARN Please initialize the log4j system properly.
Radius: 2.0, Area: 12.566370614359172
```

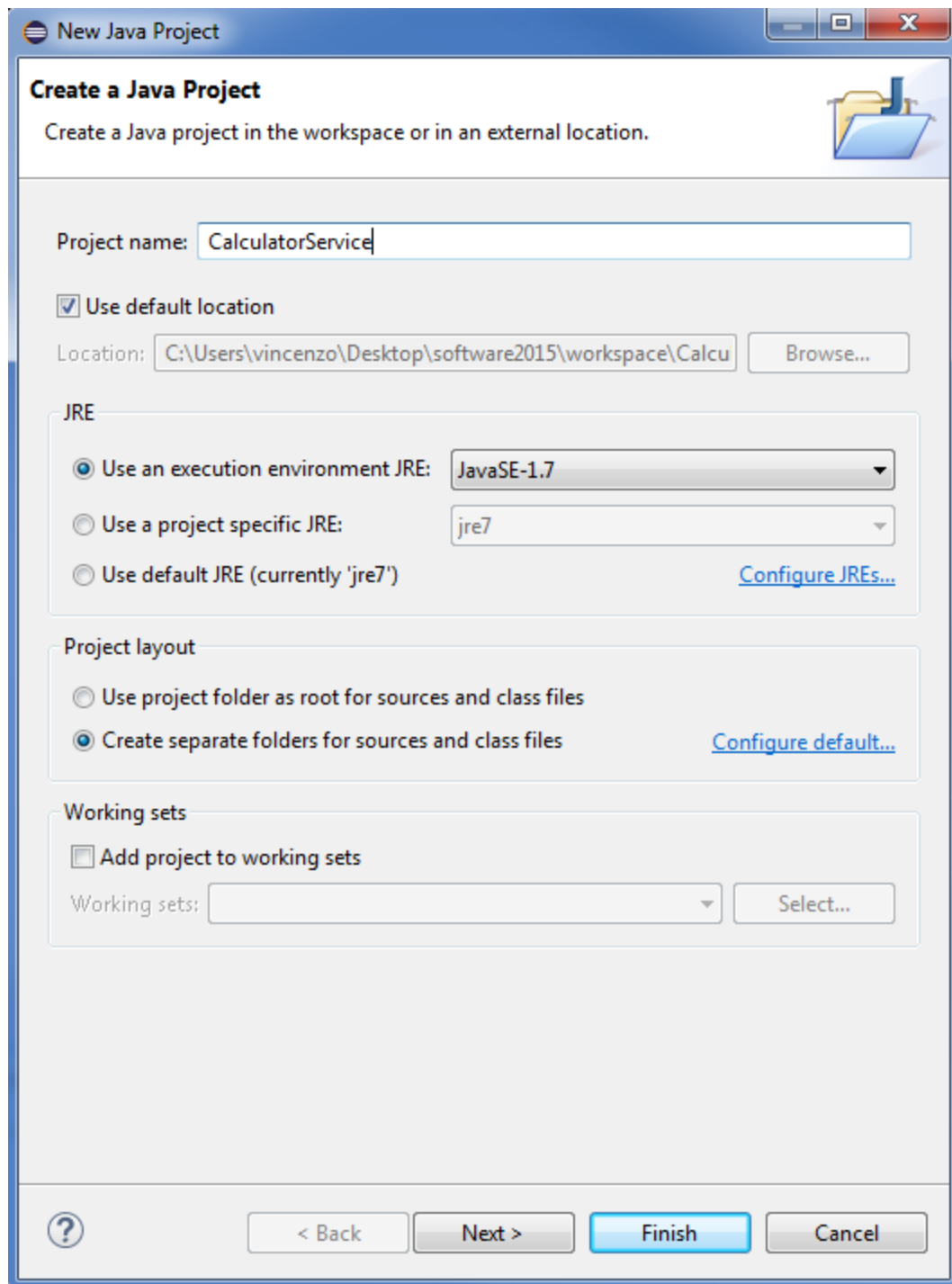
## ***[2] Calculator: Generar un servicio web a partir del fichero WSDL.***

El servicio web tiene un contador que se incrementa con el resultado de cada una de las siguientes operaciones:

- *add*: devuelve la suma de dos números .
- *incrementValue*: incrementa un contador interno en un valor dado.
- *addArray*: devuelve la suma de todos los números en un *array*.

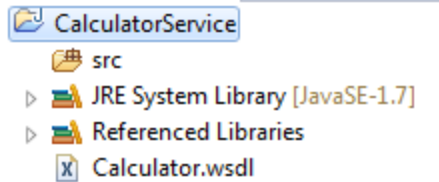
El servicio está definido por el WSDL: Calculator.wsdl

### **A. Crear un nuevo proyecto java (“Java Project”) en Eclipse.**



Importar las librerías de axis2 en el build path.

## B. Copiar el archivo Calculator.wsdl en la raíz del proyecto CalculatorService



## C. Generar el *skeleton* del servicio

Posicionarse en la carpeta raíz del proyecto y crear el *skeleton* del servicio y las demás clases necesarias para desplegar el servicio con la herramienta *wsdl2java* de Axis2.

### LINUX

```
$AXIS2_HOME/bin/wsdl2java.sh -s -ss -sd -wv 2.0 -p sos.t3.a32.calculator -d adb -uri Calculator.wsdl
```

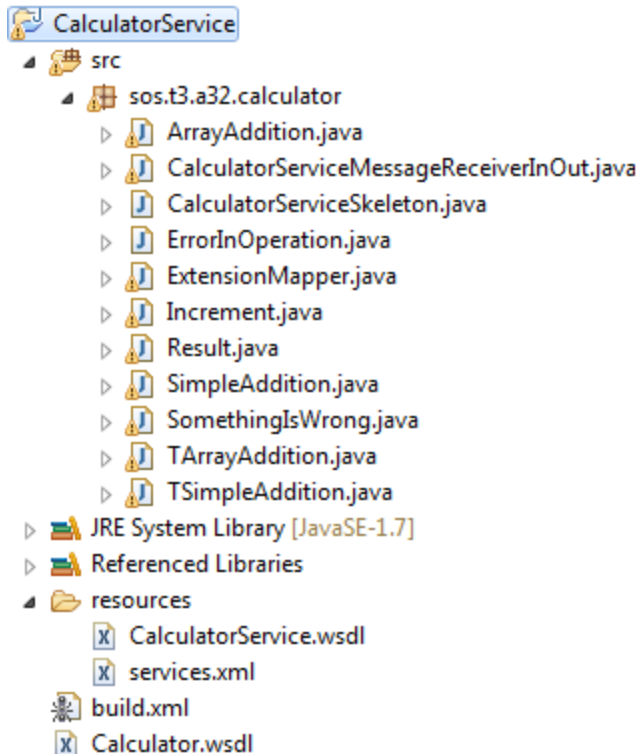
```
valerio@lsd15:~/lectures/workspace/CalculatorService$ $AXIS2_HOME/bin/wsdl2java.sh -s -ss -sd -wv 2.0 -p sos.t3.a32.calculator -d adb -uri Calculator.wsdl
Using AXIS2_HOME: /home/valerio/lectures/software2015_linux/axis2-1.6.2
Using JAVA_HOME: /home/valerio/lectures/software2015_linux/jdk1.7.0_75
[WARN] Woden[Warning],0:0,Description-1001,The targetNamespace 'http://calculator.a32.t3.sos' is not dereferencable.
valerio@lsd15:~/lectures/workspace/CalculatorService$
```

### WINDOWS

```
%AXIS2_HOME%\bin\wsdl2java.bat -s -ss -sd -wv 2.0 -p sos.t3.a32.calculator -d adb -uri Calculator.wsdl
```

```
C:\Users\vincenzo\Desktop\software2015\workspace\CalculatorService>%AXIS2_HOME%\bin\wsdl2java.bat -s -ss -sd -wv 2.0 -p sos.t3.a32.calculator -d adb -uri Calculator.wsdl
Using AXIS2_HOME: C:\Users\vincenzo\Desktop\software2015\axis2-1.6.2
Using JAVA_HOME: C:\Users\vincenzo\Desktop\software2015\jdk1.7.0_75
log4j:WARN No appenders could be found for logger (org.apache.axis2.description.WSDL20ToAxisServiceBuilder).
log4j:WARN Please initialize the log4j system properly.
```





## D. Implementar el servicio

La clase *CalculatorServiceSkeleton.java* generada por la herramienta *wdl2java* es la clase donde hay que añadir la lógica de nuestro servicio.

Por omisión, los métodos del servicio lanzan una excepción para avisar al desarrollador que debe completar el código.

```
//TODO : fill this with the necessary business logic
throw new java.lang.UnsupportedOperationException("Please implement " + this.getClass().getName() + "#incrementValue");
```

Método *add*:

```
public sos.t3.a32.calculator.Result add
(
    sos.t3.a32.calculator.SimpleAddition simpleAddition
)
    throws ErrorInOperation{
    int sum = simpleAddition.getSimpleAddition().getAddend1() +
        simpleAddition.getSimpleAddition().getAddend2();
    Result result = new Result();result.setResult(sum);
    lastResult = sum; return result;
}
```

Método *incrementValue*:

```

public sos.t3.a32.calculator.Result incrementValue
(
    sos.t3.a32.calculator.Increment increment
)
    throws ErrorInOperation{
    int sum = lastResult + increment.getIncrement();
    Result result = new Result();
    result.setResult(sum);
    lastResult = sum;
    return result;
}

```

Método *addArray*:

```

public sos.t3.a32.calculator.Result addArray
(
    sos.t3.a32.calculator.ArrayAddition arrayAddition
)
    throws ErrorInOperation{
    int[] addends = arrayAddition.getArrayAddition().getAddend();
    int sum = 0;
    for(int addend: addends)
        sum += addend;
    Result result = new Result();result.setResult(sum);
    lastResult = sum; return result;
}

```

En este caso la herramienta ha generado el fichero *services.xml*:

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- This file was auto-generated from WSDL -->
<!-- by the Apache Axis2 version: 1.6.2 Built on : Apr 17, 2012 (05:33:49 IST) -->
<serviceGroup>
    <service name="CalculatorService">
        <messageReceivers>
            <messageReceiver mep="http://www.w3.org/ns/wSDL/in-out" class="sos.t3.a32.calculator.CalculatorServiceMessageReceiverInOut"/>
        </messageReceivers>
        <parameter name="ServiceClass">sos.t3.a32.calculator.CalculatorServiceSkeleton</parameter>
        <parameter name="useOriginalwsdl">true</parameter>
        <parameter name="modifyUserWSDLPortAddress">true</parameter>
        <operation name="incrementValue" mep="http://www.w3.org/ns/wSDL/in-out" namespace="http://calculator.a32.t3.sos">
            <actionMapping>urn:incrementValue</actionMapping>
            <outputActionMapping>urn:incrementValueResponse</outputActionMapping>
            <faultActionMapping faultName="errorInOperation">urn:incrementValueerrorInOperation</faultActionMapping>
        </operation>
        <operation name="addArray" mep="http://www.w3.org/ns/wSDL/in-out" namespace="http://calculator.a32.t3.sos">
            <actionMapping>urn:addArray</actionMapping>
            <outputActionMapping>urn:addArrayResponse</outputActionMapping>
            <faultActionMapping faultName="errorInOperation">urn:addArrayerrorInOperation</faultActionMapping>
        </operation>
        <operation name="add" mep="http://www.w3.org/ns/wSDL/in-out" namespace="http://calculator.a32.t3.sos">
            <actionMapping>urn:add</actionMapping>
            <outputActionMapping>urn:addResponse</outputActionMapping>
            <faultActionMapping faultName="errorInOperation">urn:adderrorInOperation</faultActionMapping>
        </operation>
    </service>
</serviceGroup>

```

## E. Crear el paquete de despliegue

La herramienta *wsdl2java* ha generado el fichero *build.xml*. Este es un fichero de configuración de *ant* que facilita la creación del archivo de despliegue.

Posicionarse en la carpeta raíz del proyecto y utilizar *ant* para generar el archivo *aar* para el servicio. Tanto en Windows como en Linux se puede ejecutar *ant* sin ningún parámetro.

```
C:\Users\vincenzo\Desktop\software2015\workspace\CalculatorService>ant
Buildfile: C:\Users\vincenzo\Desktop\software2015\workspace\CalculatorService\build.xml

init:
[mkdir] Created dir: C:\Users\vincenzo\Desktop\software2015\workspace\CalculatorService\build
[mkdir] Created dir: C:\Users\vincenzo\Desktop\software2015\workspace\CalculatorService\build\classes
[mkdir] Created dir: C:\Users\vincenzo\Desktop\software2015\workspace\CalculatorService\build\lib

pre-compile.test:
[echo] Stax Availability= true
[echo] Axis2 Availability= true

compile.src:
[javac] C:\Users\vincenzo\Desktop\software2015\workspace\CalculatorService\build.xml:43: warning: 'includeantruntime' was not
set, defaulting to build.sysclasspath=last; set to false for repeatable builds
[javac] Compiling 11 source files to C:\Users\vincenzo\Desktop\software2015\workspace\CalculatorService\build\classes
[javac] Note: C:\Users\vincenzo\Desktop\software2015\workspace\CalculatorService\src\sos\t3\calculator\CalculatorServiceMe
ssageReceiverInOut.java uses unchecked or unsafe operations.
[javac] Note: Recompile with -Xlint:unchecked for details.

echo.classpath.problem:

jar.server:
[copy] Copying 2 files to C:\Users\vincenzo\Desktop\software2015\workspace\CalculatorService\build\classes\META-INF
[jar] Building jar: C:\Users\vincenzo\Desktop\software2015\workspace\CalculatorService\build\lib\CalculatorService.aar

BUILD SUCCESSFUL
Total time: 3 seconds
```

Se ha creado con *ant* el fichero *CalculatorService.aar* en la carpeta *build\lib\*

## F. Despliegue del servicio en Tomcat

Copiar el archivo *CalculatorService.aar* en

### LINUX

`$CATALINA_HOME/webapps/axis2/WEB-INF/services/`

### WINDOWS

`%CATALINA_HOME%\webapps\axis2\WEB-INF\services\`

## G. Comprobar en la lista de servicios de axis que nuestro nuevo servicio está disponible

<http://localhost:8080/axis2/services/listServices>

### CalculatorService

Service Description : CalculatorService

Service EPR : <http://localhost:8080/axis2/services/CalculatorService>

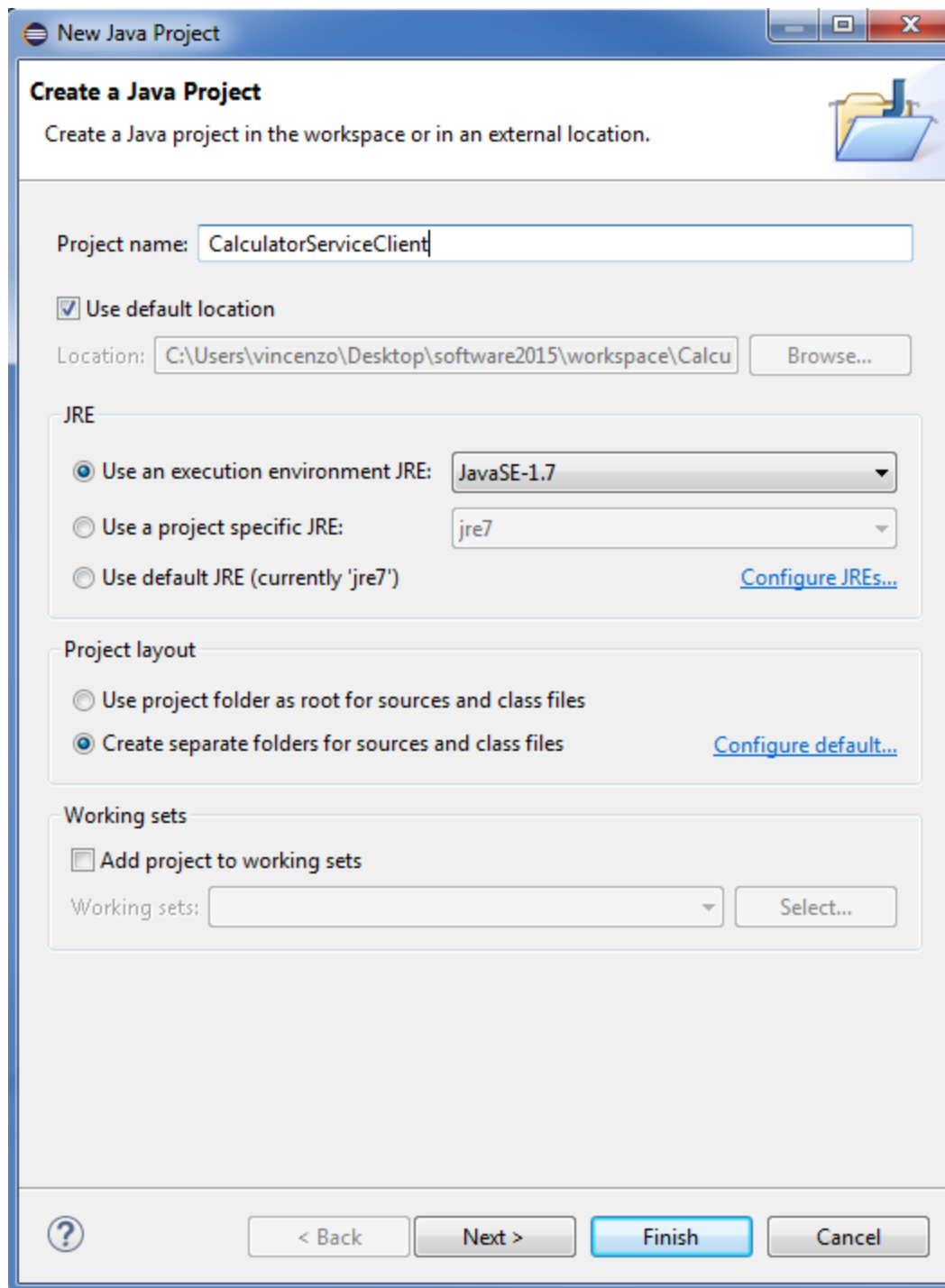
Service Status : Active

#### *Available Operations*

- incrementValue
- addArray
- add

## H. Creación de un cliente Java

Crear un nuevo “Java Project” en Eclipse, *GeometryServiceClient*.



Copiar el fichero *Calculator.wsdl* utilizado para generar el *skeleton* del servicio en la raíz de este nuevo proyecto y añadir las librerías de axis2

Generar los *stubs* necesarios para el cliente para llamar al servicio con *wsdl2java*.

## LINUX

```
$AXIS2_HOME/bin/wsd12java.sh -uri Calculator.wsdl -vv 2.0 -p sos.t3.a32.calculator.client  
-d adb
```

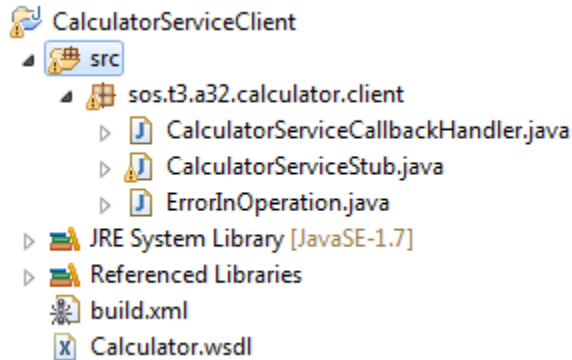
```
valerio@isd15:~/lectures/workspace/CalculatorServiceClient$ $AXIS2_HOME/bin/wsd12java.sh -uri  
Calculator.wsdl -vv 2.0 -p sos.t3.a32.calculator.client -d adb  
Using AXIS2_HOME: /home/valerio/lectures/software2015_linux/axis2-1.6.2  
Using JAVA_HOME: /home/valerio/lectures/software2015_linux/jdk1.7.0_75  
[WARN] Woden[Warning],0:0,Description-1001,The targetNamespace 'http://calculator.a32.t3.sos' i  
s not dereferencable.
```

## WINDOWS

```
%AXIS2_HOME%\bin\wsdl2java.bat -uri Calculator.wsdl -vv 2.0 -p sos.t3.a32.calculator.client -d adb
```

```
C:\Users\vincenzo\Desktop\software2015\workspace\CalculatorServiceClient>%AXIS2_HOME%\bin\wsdl2java.bat -uri Calculator.wsdl -vv  
2.0 -p sos.t3.a32.calculator.client -d adb  
Using AXIS2_HOME: C:\Users\vincenzo\Desktop\software2015\axis2-1.6.2  
Using JAVA_HOME: C:\Users\vincenzo\Desktop\software2015\jdk1.7.0_75  
log4j:WARN No appenders could be found for logger <org.apache.axis2.description.WSDL20ToAxisServiceBuilder>.  
log4j:WARN Please initialize the log4j system properly.
```

En la carpeta *src* de Eclipse están los *stubs* generados.



Creemos la clase *CalculatorServiceClient* con este código:

```
package sos.t3.a32.calculator.client;

import sos.t3.a32.calculator.client.CalculatorServiceStub.ArrayAddition;
import sos.t3.a32.calculator.client.CalculatorServiceStub.Increment;
import sos.t3.a32.calculator.client.CalculatorServiceStub.Result;
import sos.t3.a32.calculator.client.CalculatorServiceStub.SimpleAddition;
import sos.t3.a32.calculator.client.CalculatorServiceStub.TArrayAddition;
import sos.t3.a32.calculator.client.CalculatorServiceStub.TSimpleAddition;

public class CalculatorServiceClient {
    public static void main(String[] args) throws Exception {
        CalculatorServiceStub cs = new CalculatorServiceStub();
        TSimpleAddition twoAddends = new TSimpleAddition();
        twoAddends.setAddend1(5);
        twoAddends.setAddend2(4);
        SimpleAddition simpleAddition = new SimpleAddition();
        simpleAddition.setSimpleAddition(twoAddends);
        System.out.println("5+4="+cs.add(simpleAddition).getResult());

        ArrayAddition arrayAddition = new ArrayAddition();
        TArrayAddition param = new TArrayAddition();
        int[] array = {3,6,9};
        param.setAddend(array);
        arrayAddition.setArrayAddition(param);
        Result addArray = cs.addArray(arrayAddition );
        System.out.println("Sum array {3,6,9}= " + addArray.getResult());

        Increment increment = new Increment();
        increment.setIncrement(10);
        System.out.println("Prev+10="+cs.incrementValue(increment).getResult());

        increment.setIncrement(5);
        System.out.println("Prev+5="+cs.incrementValue(increment).getResult());
    }
}
```

Ejecutando el código en Eclipse se obtiene la salida:

```
log4j:WARN No appenders could be found for logger (org.apache.axis2.description.AxisOperation).
log4j:WARN Please initialize the log4j system properly.
5+4=9
Sum array {3,6,9}= 18
Prev+10=10
Prev+5=5
```

¿Es correcto el resultado de incrementValue?