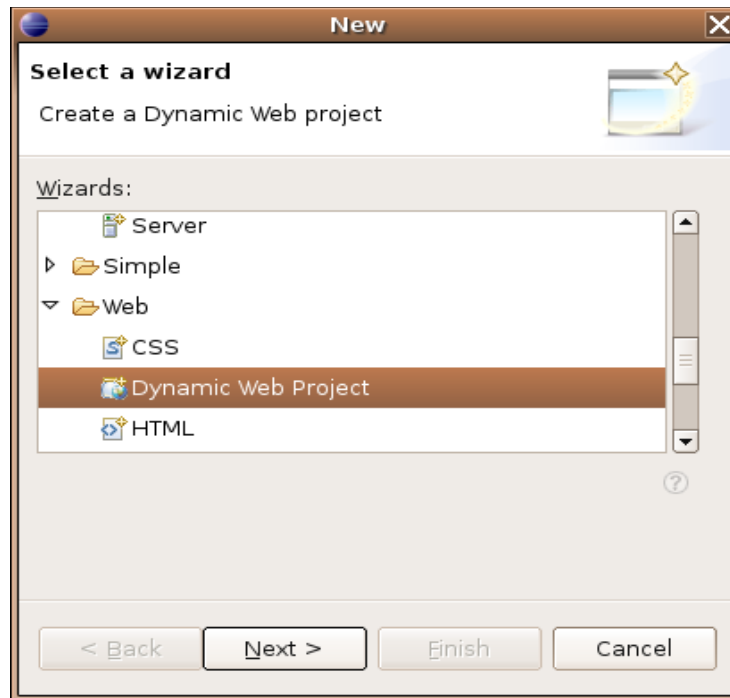


Creació d'un servei web XML amb Eclipse. Top down Java bean Web Service

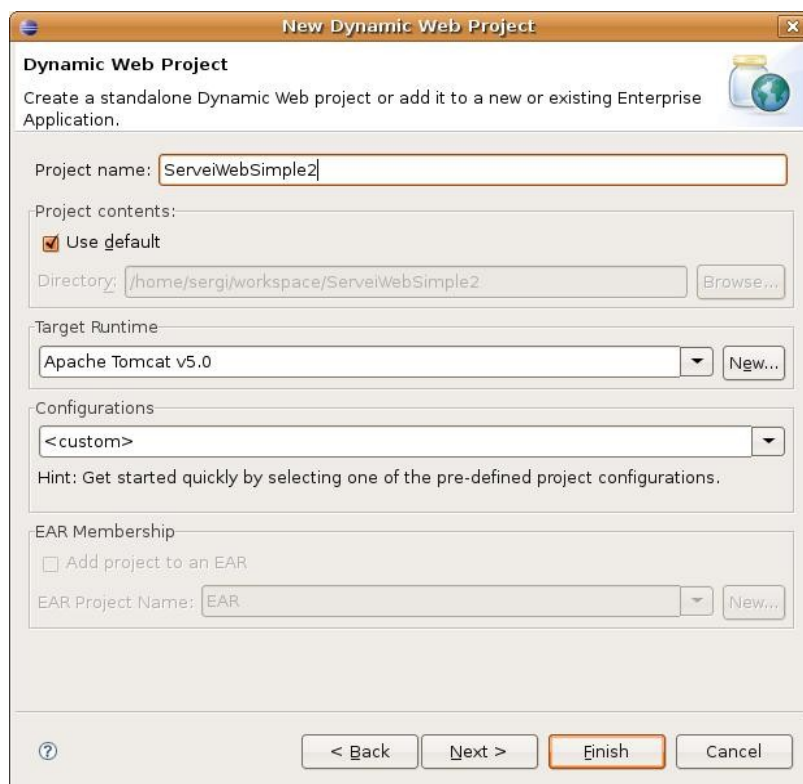
L'objectiu d'aquest exercici és construir un servei web seguint la filosofia inversa a la del exercici anterior (és a dir crear un servei web “Top down Java bean Web Service”)

Creació d'un projecte Web Dinàmic

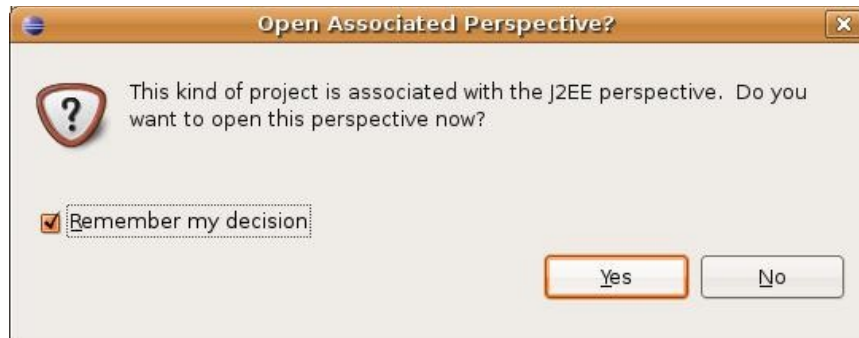
Tornem a pressionar les tecles **Ctrl+N** i creem un **Dynamic Web Project**.



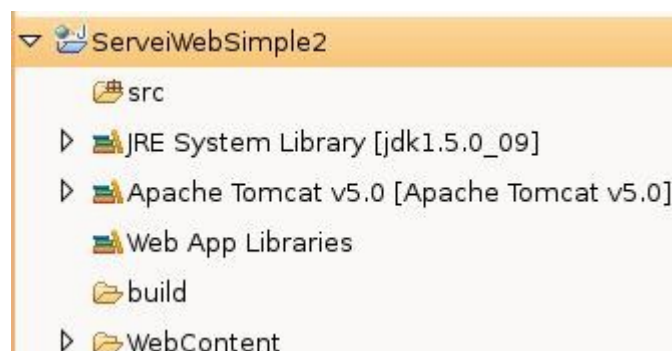
A la següent finestra creem un projecte anomenen **ServeiWebSimple2** i seleccionem **Finish**:



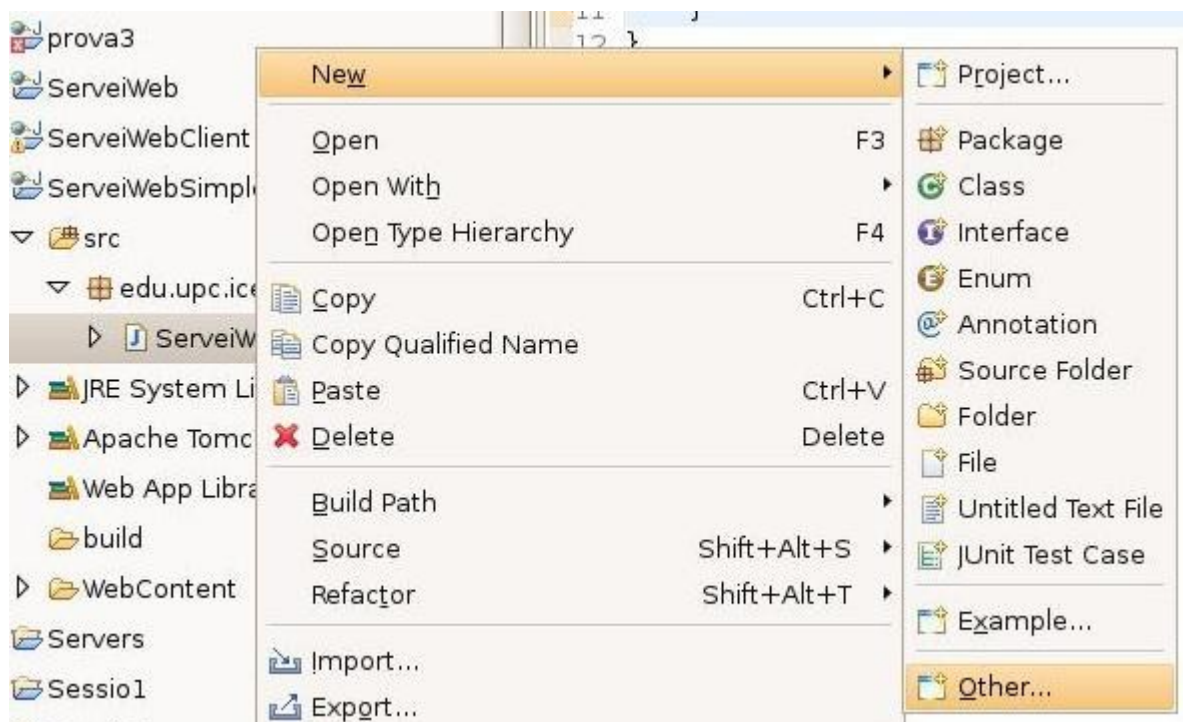
Indiquem que NO volem canviar la perspectiva:

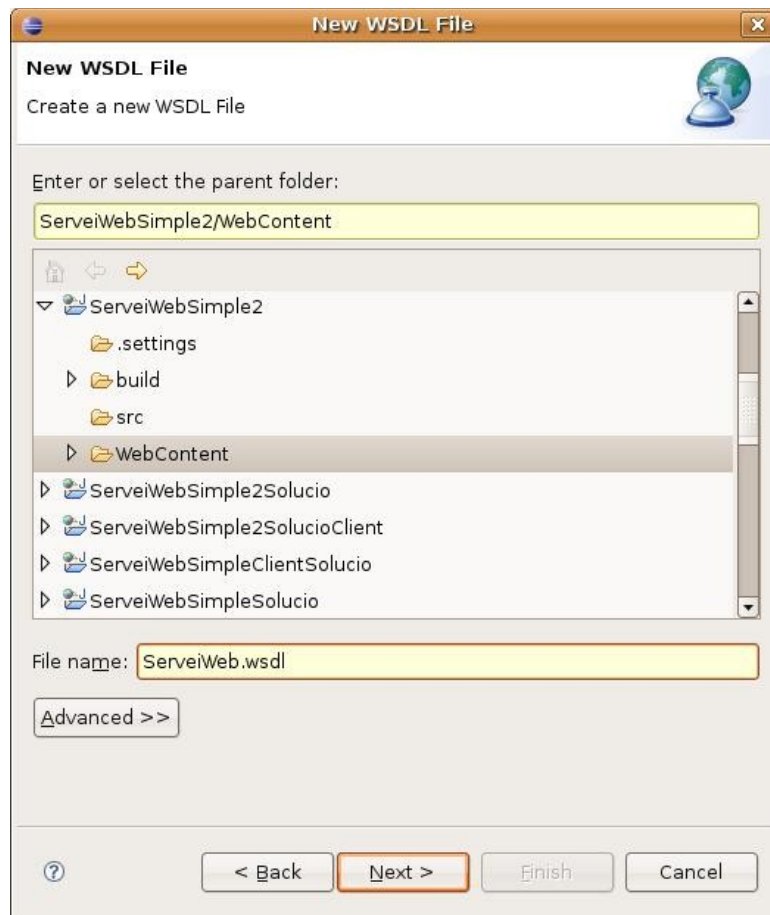
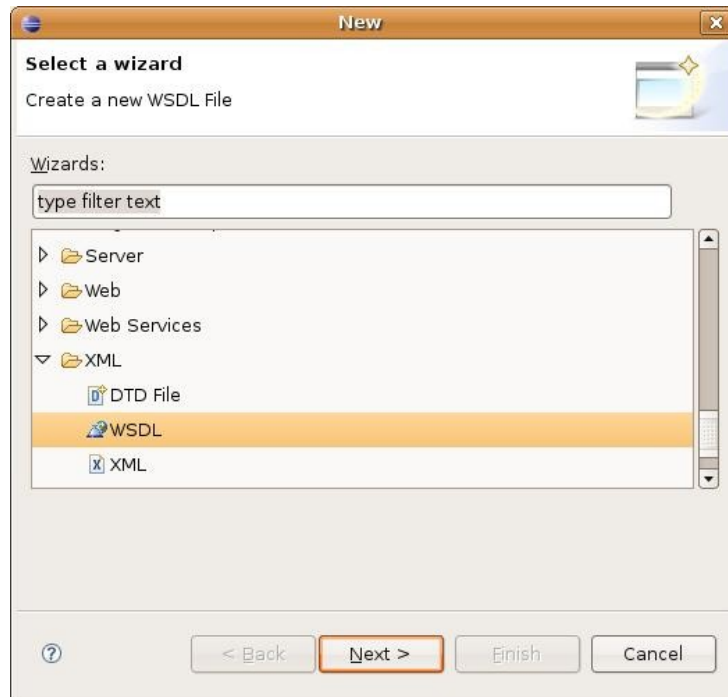


I tindrem un projecte com el següent:



Ara creem un fitxer de descripció d'un servei web (fitxer WSDL) (boto dret a la carpeta WebContent/new/Other) que anomenarem **ServeiWeb**.





A la següent finestra canviem el camp **Target namespace** establint el valor a :

<http://localhost:8080/ServeiWeb>



New WSDL File

Options
Specify the attributes for the new WSDL file.

Target namespace:

Prefix:

☒ Create WSDL Skeleton

Protocol:

SOAP Binding Options

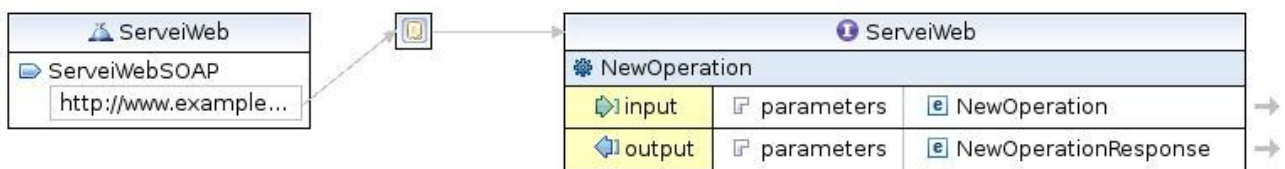
☒ document literal

☐ rpc literal

☐ rpc encoded

[Modify project compliance setting](#)

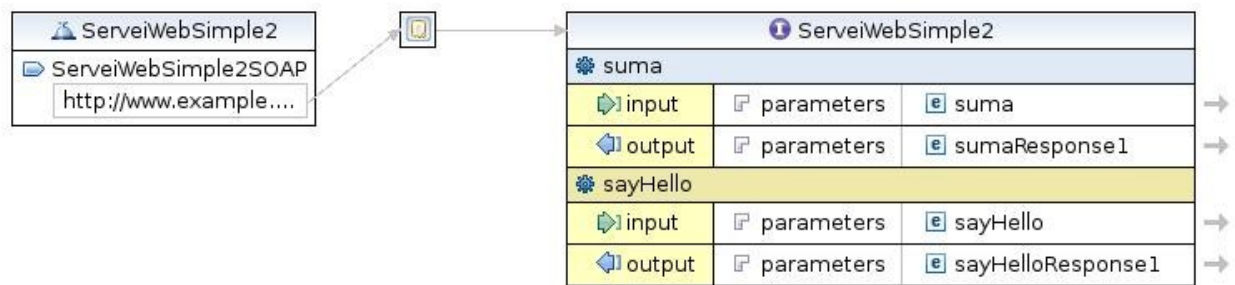
L'assistent ens creara un document base WSDL com el següent:



Utilitzant els diagrama i el codi font hem d'executar el següents passos:

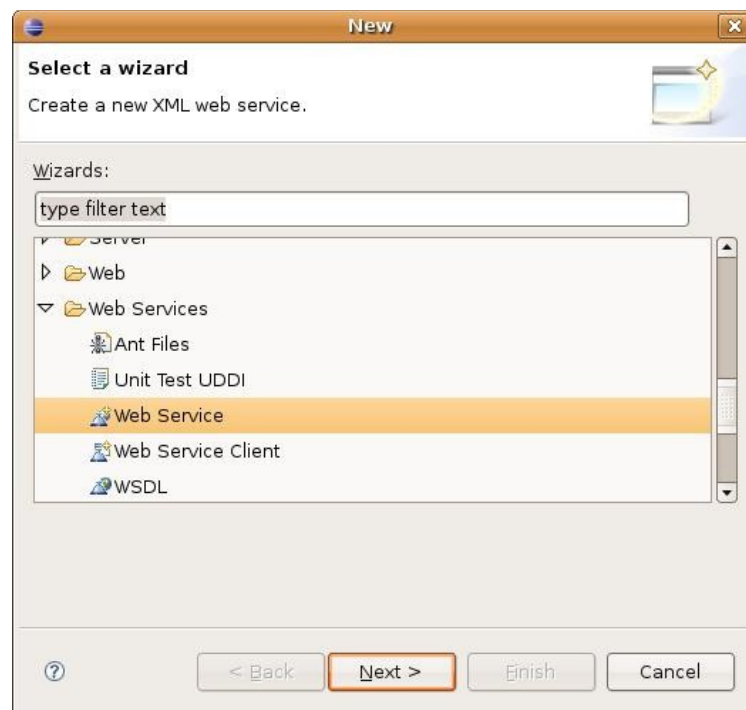
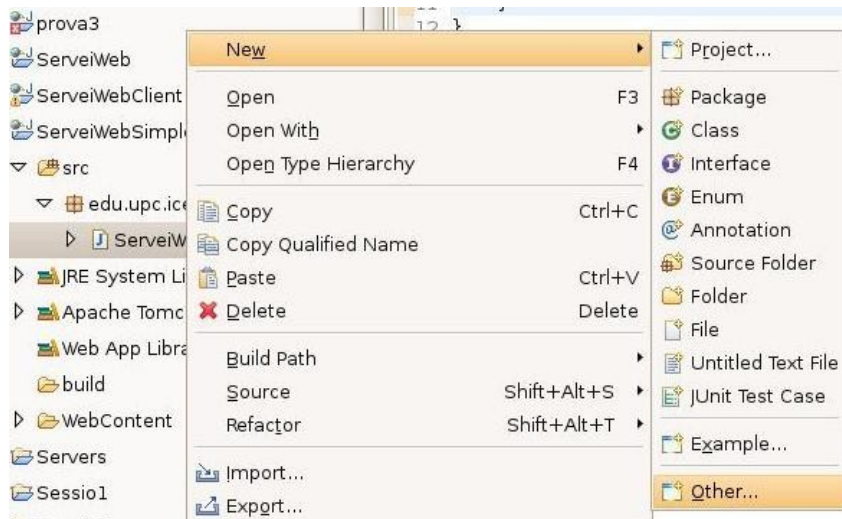
- Eliminar l'operació NewOperation
- Afegir dues operacions:
 - suma
 - sayHello

El document WSDL resultant ha de quedar de la següent manera:

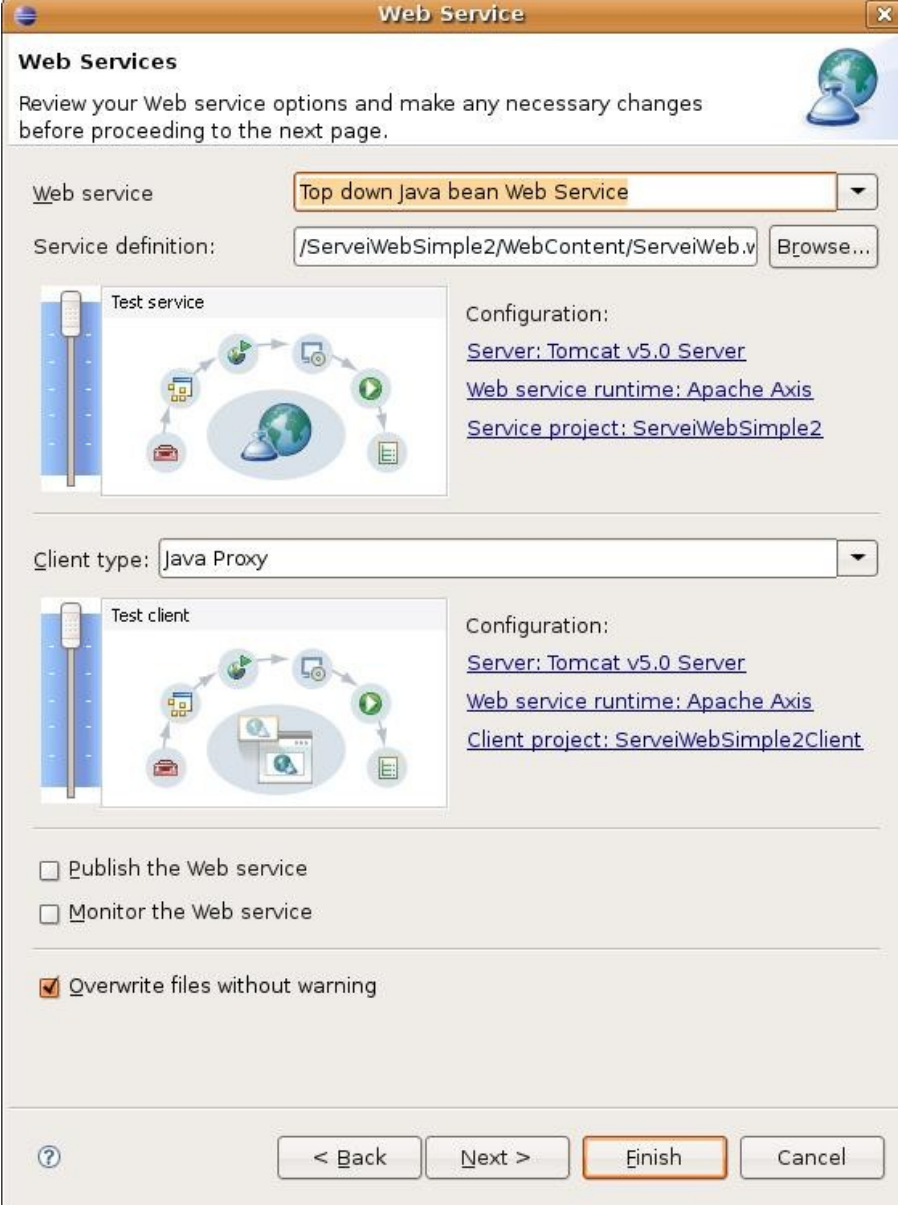


Per assegurar-vos de no tenir cap problema podeu consultar el fitxer WSDL que ha creat l'assistent de l'exemple anterior.

Un cop tenim el document WSDL creat tornem a executar com en l'exemple anterior el assistent de creació d'un servei web. Executem botó dret sobre el fitxer **ServeiWeb.wsdl/New/Other**:



Omplim la pàgina de configuració del servei web i del seu corresponent client de la següent forma:



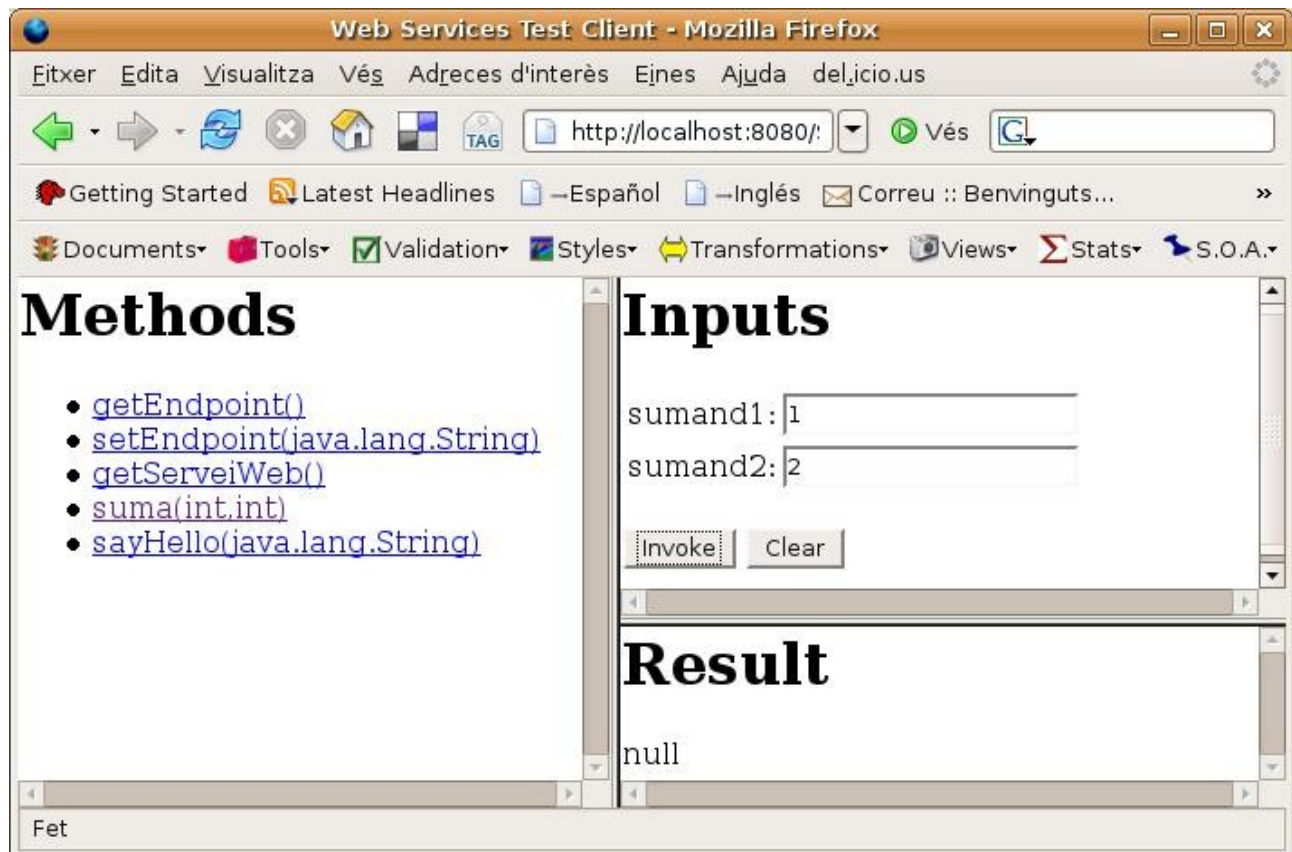
The image shows a 'Web Service' configuration window. At the top, it says 'Web Services' and 'Review your Web service options and make any necessary changes before proceeding to the next page.' Below this, there's a 'Web service' dropdown menu set to 'Top down Java bean Web Service'. Under 'Service definition:', there's a text box with '/ServeiWebSimple2/WebContent/ServeiWeb.v' and a 'Browse...' button. To the left, there's a 'Test service' diagram showing a server, a client, and a globe. To the right, under 'Configuration:', there are links for 'Server: Tomcat v5.0 Server', 'Web service runtime: Apache Axis', and 'Service project: ServeiWebSimple2'. Below this, there's a 'Client type:' dropdown menu set to 'Java Proxy'. Underneath, there's a 'Test client' diagram showing a server, a client, and a globe. To the right, under 'Configuration:', there are links for 'Server: Tomcat v5.0 Server', 'Web service runtime: Apache Axis', and 'Client project: ServeiWebSimple2Client'. At the bottom, there are three checkboxes: 'Publish the Web service' (unchecked), 'Monitor the Web service' (unchecked), and 'Overwrite files without warning' (checked). At the very bottom, there are four buttons: '< Back', 'Next >', 'Finish' (highlighted with an orange border), and 'Cancel'.

NOTA: És important que tant el servei web com el client estiguin en mode test (barra al màxim).

Executem **Finish** i el resultat serà un nou projecte client:



que inclou la següent aplicació web per provar el servei web:



Al executar els tests veureu que sempre ens torna null. Això es degut ha que l'assistent ens crea tot l'esquelet del servei web però la implementació de les funcions **suma** i **sayHello** encara esta pendent. Les trobareu al fitxer **ServeiWebSoapBindingImpl.java**:

```
* ServeiWebSoapBindingImpl.java

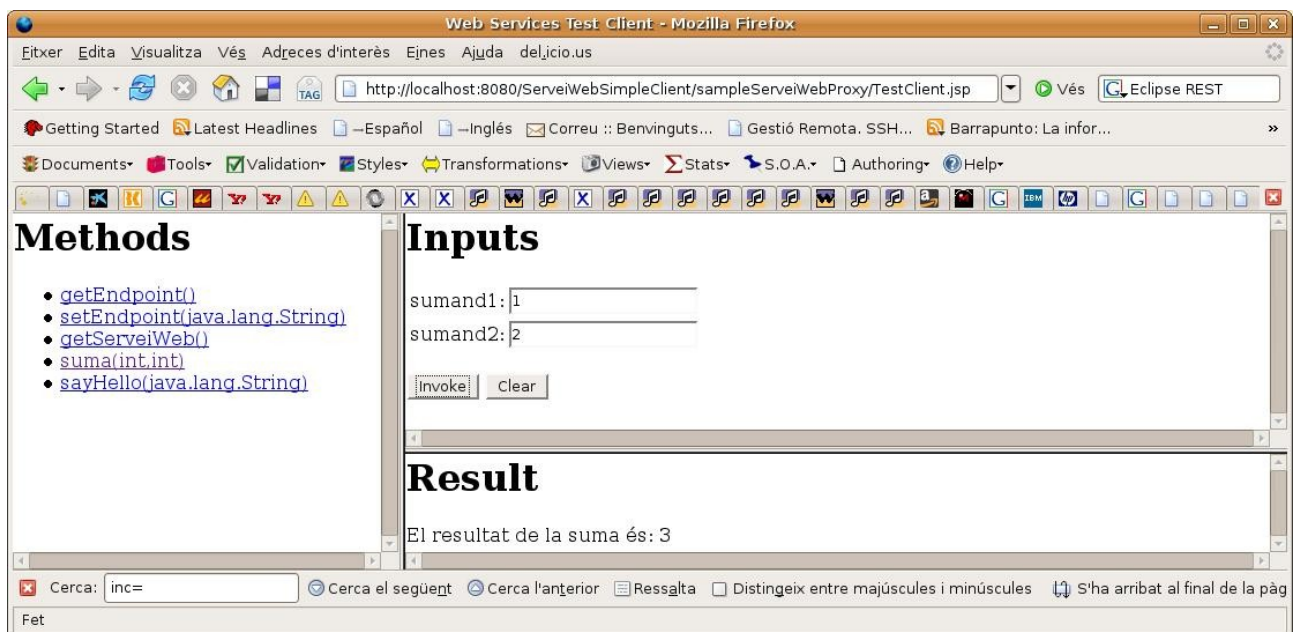
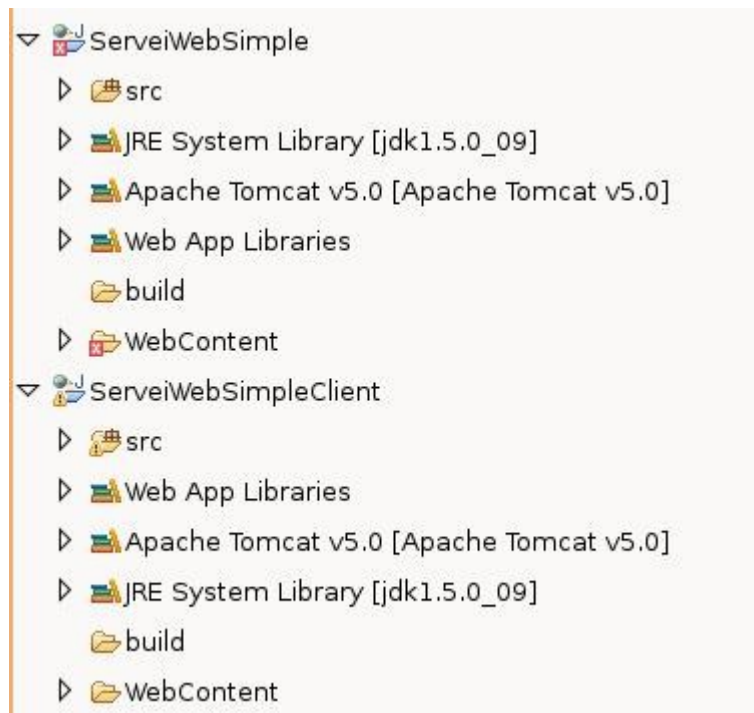
package edu.upc.ice.cursXML;

public class ServeiWebSoapBindingImpl implements edu.upc.ice.cursXML.ServeiWeb{
    public java.lang.String suma(int sumand1, int sumand2) throws java.rmi.RemoteException {
        return null;
    }

    public java.lang.String sayHello(java.lang.String name) throws java.rmi.RemoteException {
        return null;
    }
}
```

Per acabar, només us faltara implementar aquestes funcions i ja teniu el servei web fet.

Seleccionem **Finish** i es creara tan el Servei web com un client per provar el servei.



EXERCICI: Afegiu alguna funció més o menys simple a l'objecte **ServeiWeb** i torneu a construir el servei web.