

MTIS	Metodologías y Tecnologías de Integración de Sistemas
	Práctica 3
	Ejercicio Guiado 1 de composición de servicios con BPEL Designer y ODE

Ejercicio Guiado 1 de composición de servicios con BPEL Designer y ODE

Prerrequisitos

1. Instalación del contenedor Web Apache Tomcat 8.0.20
2. Instalación de MySQL 5.x
3. Instalación de Apache ODE 1.3.6
4. Instalación de Eclipse Java EE Mars
5. Máquina virtual de Java 1.8
6. Apache Axis2

Objetivos

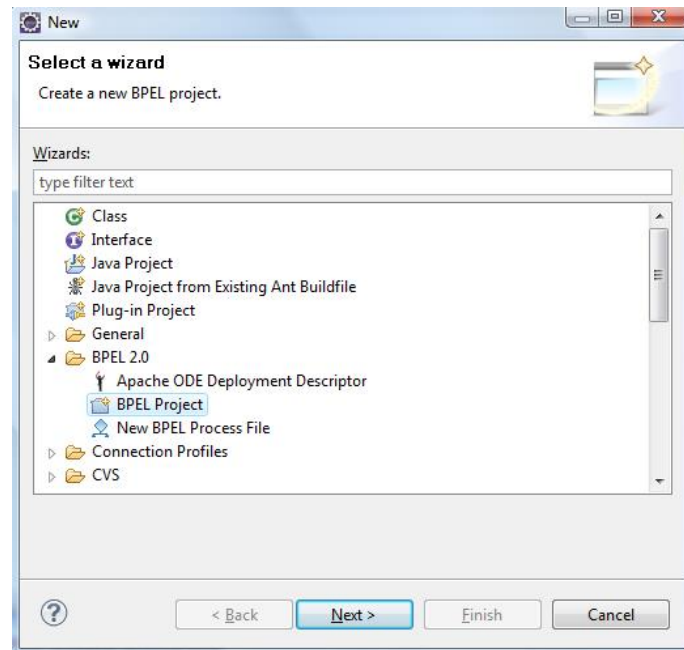
El objetivo de estos ejercicios es familiarizarse con el estándar WS-BPEL y sus principales elementos para la composición. Para ello usaremos como entorno de modelado de procesos de Eclipse BPEL Designer y Apache ODE como motor de ejecución de procesos.

1. Enunciado

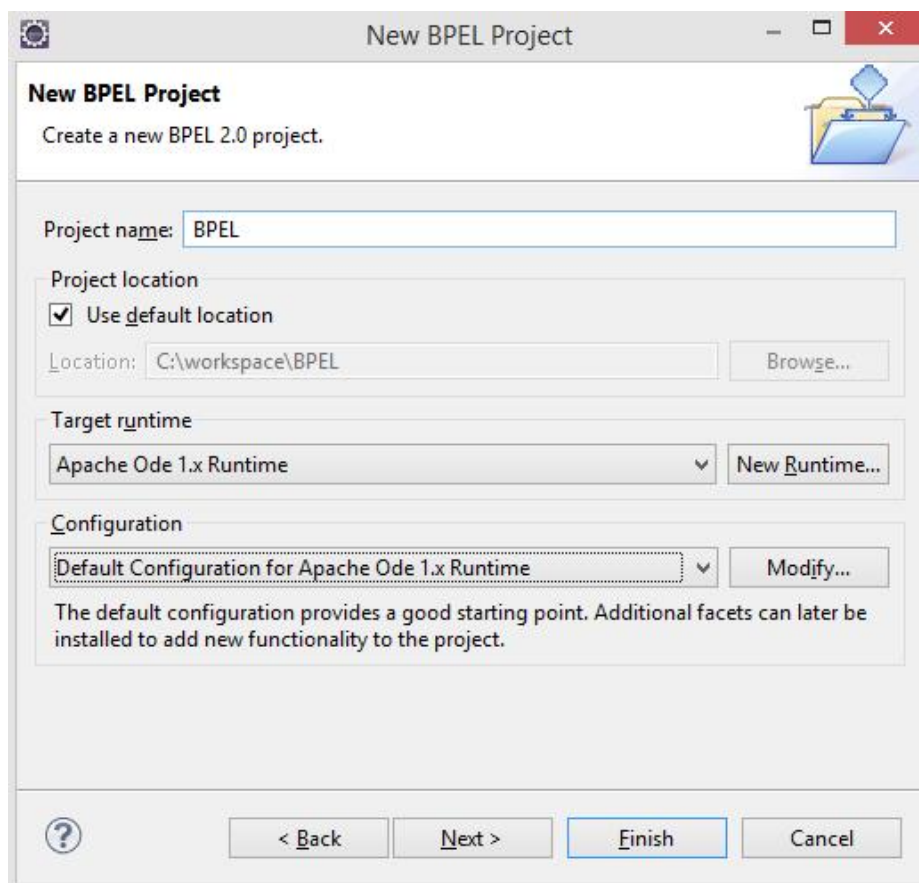
En este ejercicio se introducirá al alumno a los siguientes elementos de BPEL a través del ejemplo *HolaMundo*:

- Variables
- Actividad *Receive*
- Actividad *Assign*
- Actividad *Reply*
- Actividad *Sequence*

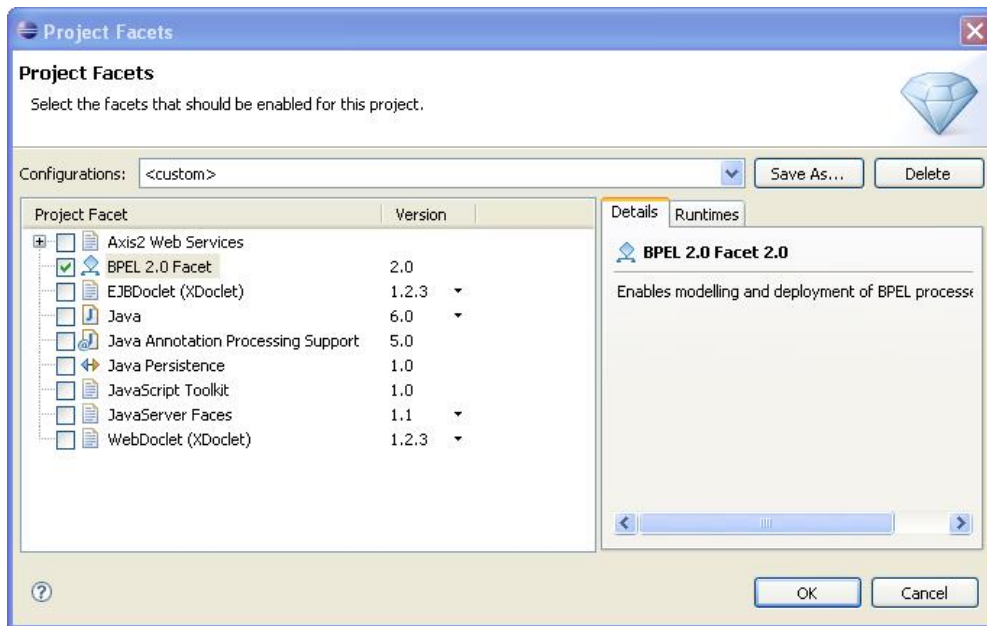
- 1.1 Crear un nuevo proyecto BPEL seleccionando en el menú *File* → *New* → *Other...* → *BPEL 2.0* → *BPEL Project* y seleccionar *Next*.



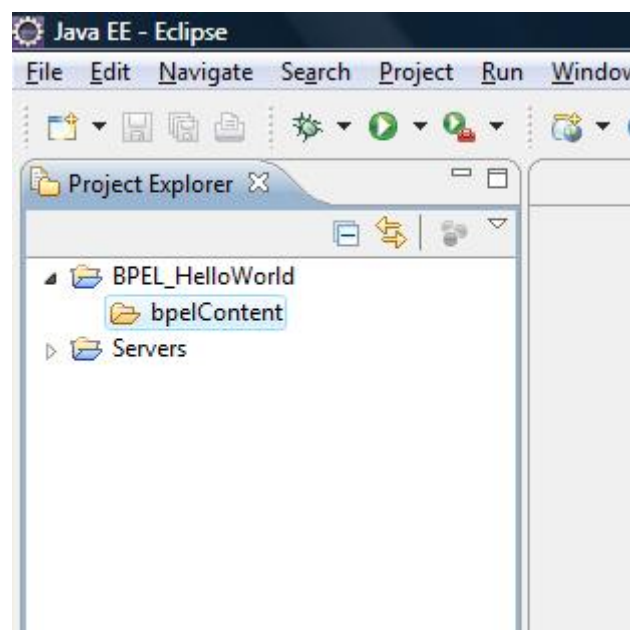
- 1.2 Dar un nombre al proyecto, por ejemplo `BPEL_HelloWorld`, y seleccionar como motor de ejecución (tarjet runtime) *Apache ODE 1.x Runtime*.



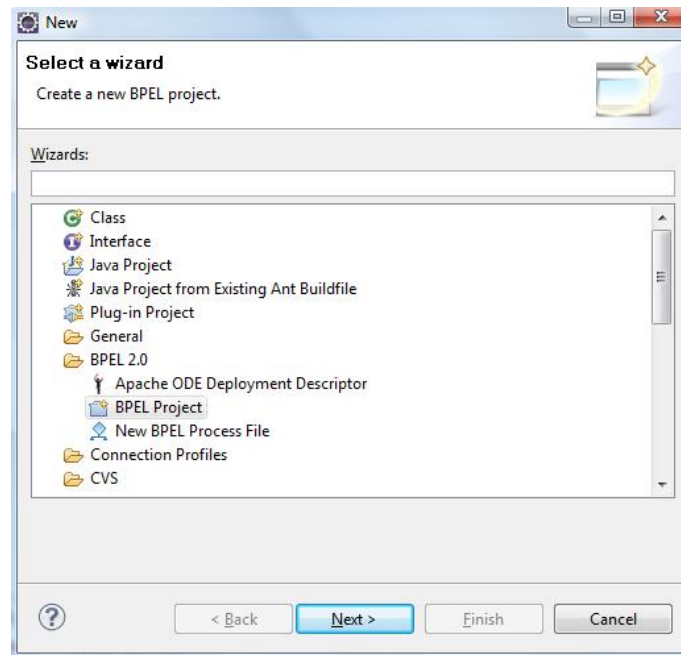
- 1.3 Pulsar en el botón *Modify* para seleccionar *BPEL 2.0 Facet*, que es el perfil que vamos a utilizar. Si aparece la configuración directamente no hace falta realizar este paso.



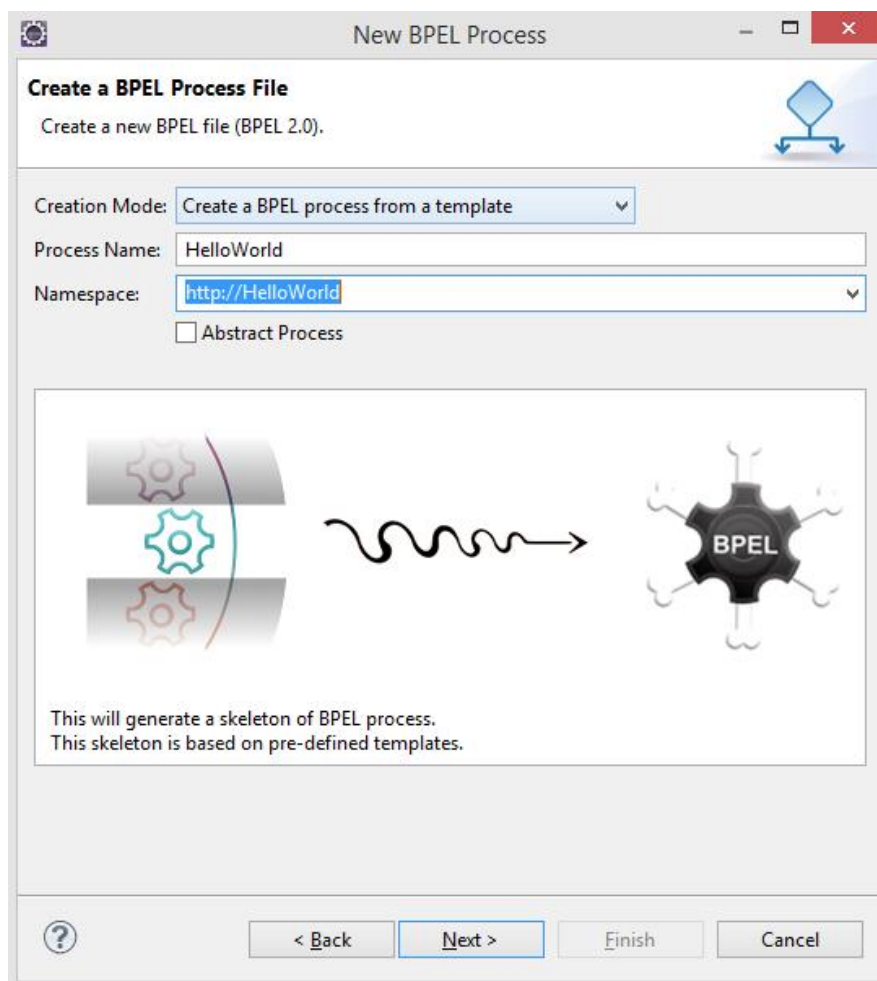
1.4 Pulsamos *Ok* y después *Finish*.

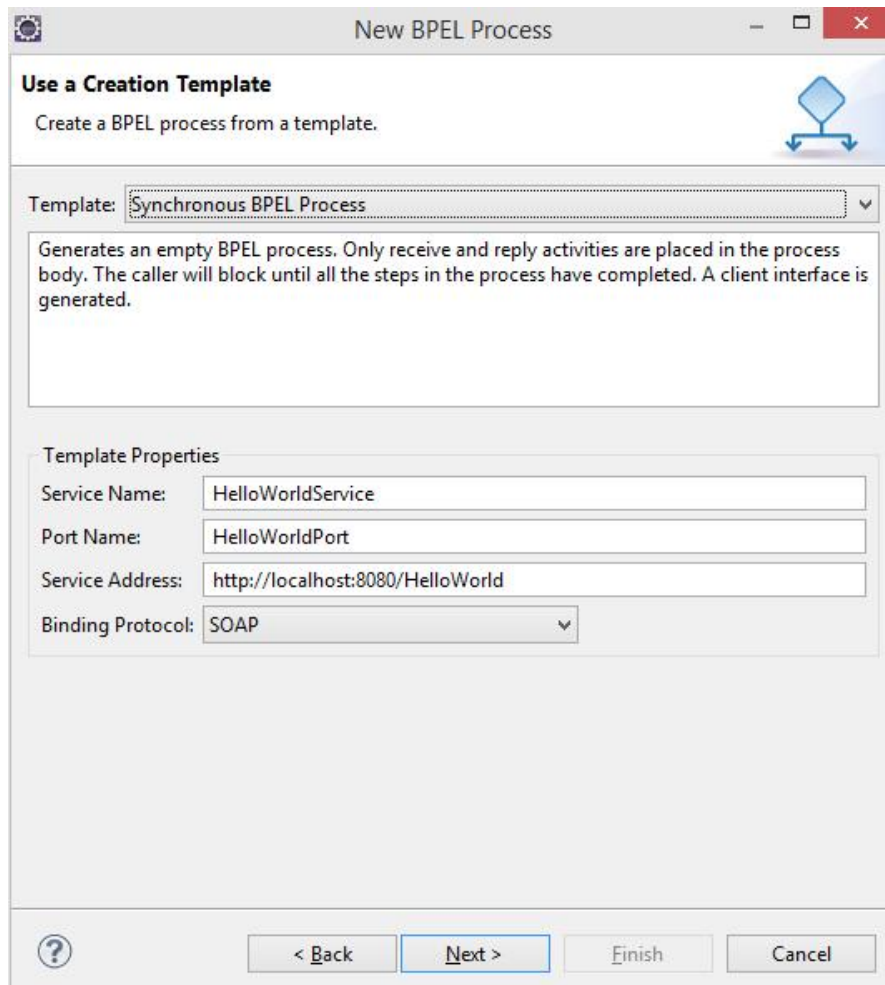


1.5 Ahora, dentro del proyecto, creamos un archivo BPEL pulsando con el botón derecho sobre la carpeta *bpelContent* del proyecto y seleccionando la opción *New* → *Other...* → *BPEL 2.0* → *New BPEL Process File* y pulsamos *Next*.

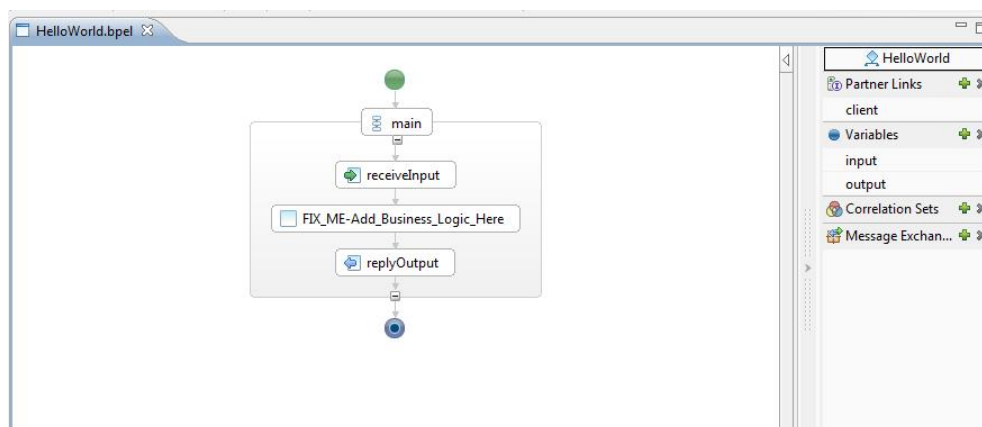


1.6 Indicamos el nombre del proyecto, su *namespace* y le indicamos que se trata de un proceso síncrono y aceptamos los siguientes pasos por defecto pulsando en *Next* varias veces. Pulsar *Finish* al finalizar.

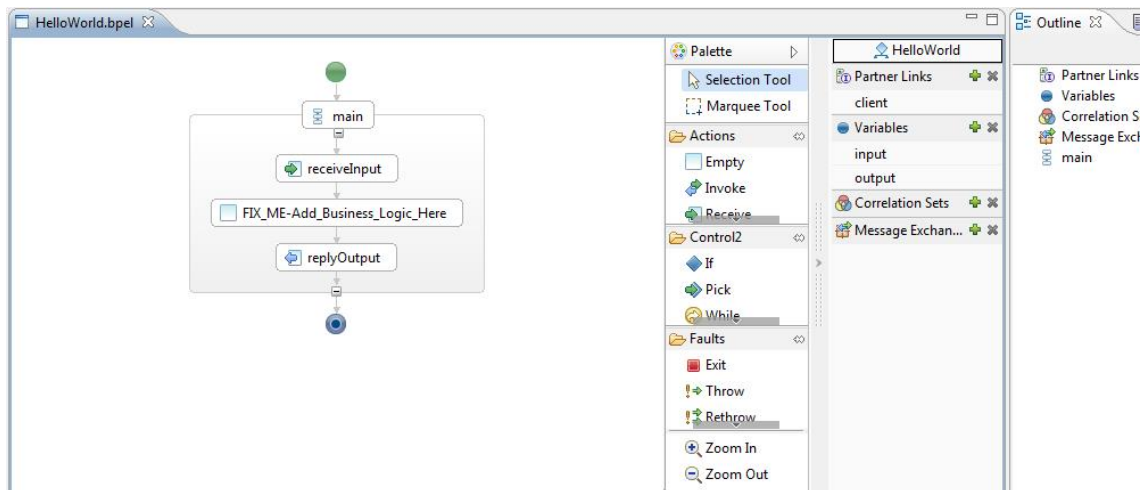




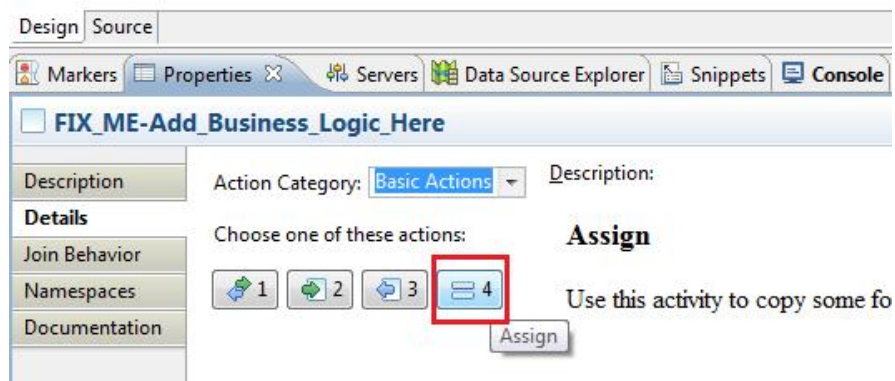
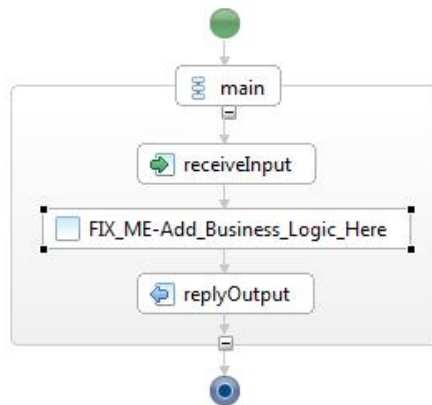
1.7 El editor BPEL mostrará el proceso definido sobre el que debemos trabajar.



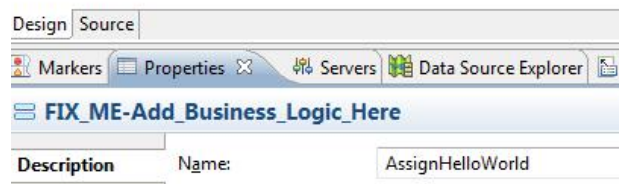
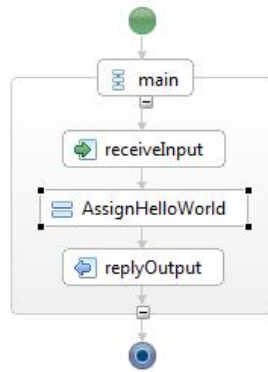
1.8 Para modificar el proceso usaremos la paleta de componentes BPEL situada a la derecha del editor.



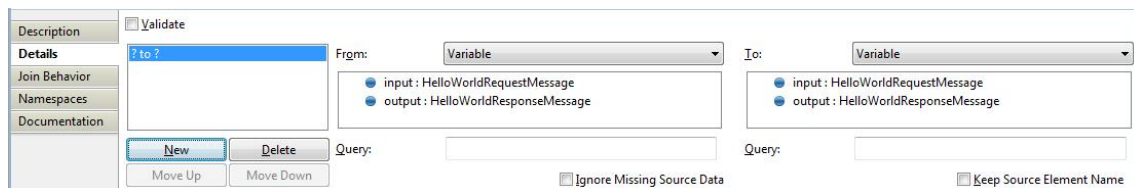
- 1.9 En el ejemplo, vamos a devolver la cadena *Hello World* que envía el cliente que invoca el proceso. Para ello, usaremos la actividad *Assign* que asignará la entrada del proceso a la salida del mismo. Primero debemos seleccionar la actividad *empty* denominada *FIX_ME-Add_Business_Logic_Here* y seleccionar en el espacio inferior la pestaña *Properties* → *Details* y convertir la actividad en un tipo *Assign*.



- 1.10 En la opción *Description* podemos cambiar el nombre de la actividad



- 1.11 Seleccionamos la actividad *Assign* y en *Properties* → *Details* pulsamos en el botón *New* para crear nuevas variables que nos sirvan en el intercambio de valores entre las actividades de entrada y salida (*receive-reply*).



- 1.12 Las actividades *receive* y *reply* incluyen por defecto las variables de entrada y salida respectivamente (*input* es la variable de la actividad *receive* y *output* de la actividad *reply*).

```

<bpel:variables>
  <!-- Reference to the message passed as input during initiation -->
  <bpel:variable name="input"
    messageType="tns:HelloWorldRequestMessage"/>

  <!--
    Reference to the message that will be returned to the requester
  -->
  <bpel:variable name="output"
    messageType="tns:HelloWorldResponseMessage"/>
</bpel:variables>

```

...


```

<bpel:receive name="receiveInput" partnerLink="client"
  portType="tns:HelloWorld"
  operation="process" variable="input"
  createInstance="yes"/>

<bpel:reply name="replyOutput"
  partnerLink="client"
  portType="tns:HelloWorld"
  operation="process"
  variable="output"
/>

```

1.13 Los tipos de estas variables están definidas en la WSDL asociada al proceso *HelloWorldArtifacts.wsdl*.

```

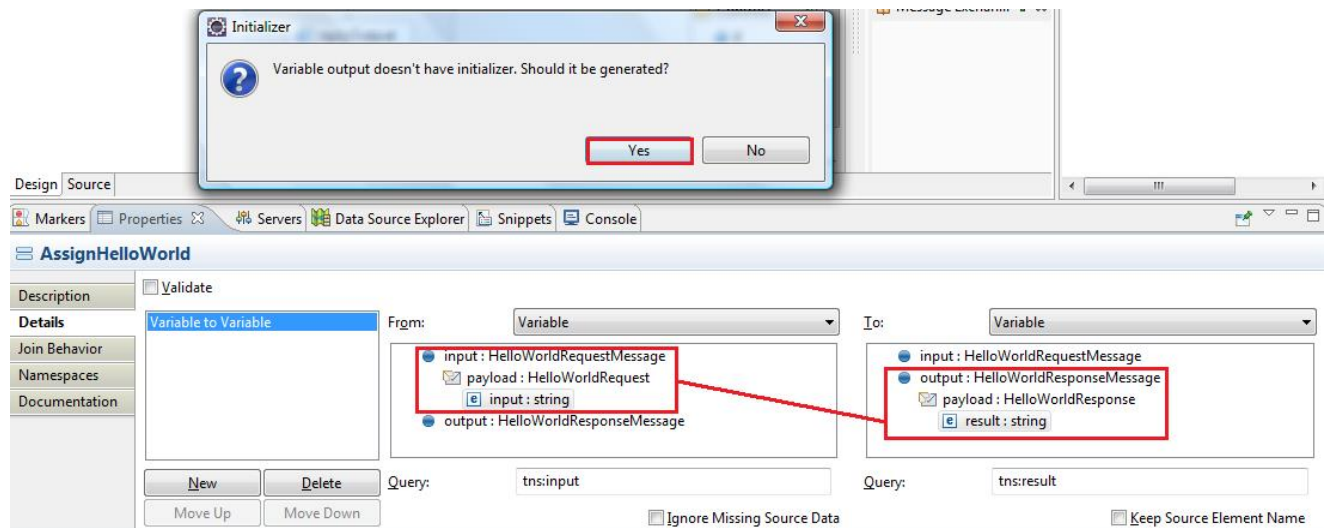
<types>
  <schema attributeFormDefault="unqualified" elementFormDefault="qualified"
    targetNamespace="http://helloworld"
    xmlns="http://www.w3.org/2001/XMLSchema">

    <element name="HelloWorldRequest">
      <complexType>
        <sequence>
          <element name="input" type="string"/>
        </sequence>
      </complexType>
    </element>

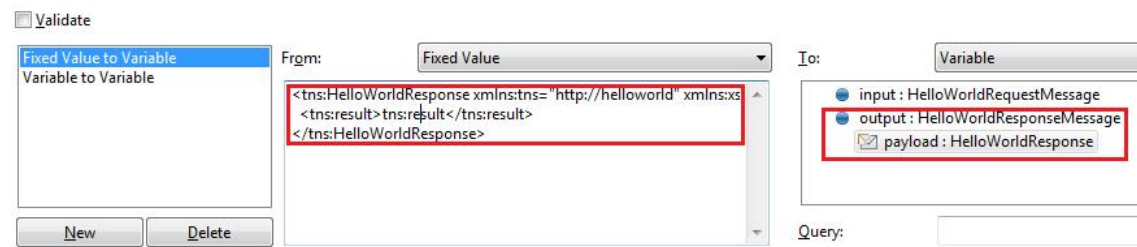
    <element name="HelloWorldResponse">
      <complexType>
        <sequence>
          <element name="result" type="string"/>
        </sequence>
      </complexType>
    </element>
  </schema>
</types>

```

1.14 Con la actividad *Assign* asignamos la entrada recibida por el proceso a la salida del mismo (De input → payload → input a output → payload → result). Aparecerá un menú emergente indicando si deseamos inicializar las variables. Debemos indicarle que sí.



1.15 Podemos observar que inicializa la variable de salida del proceso.

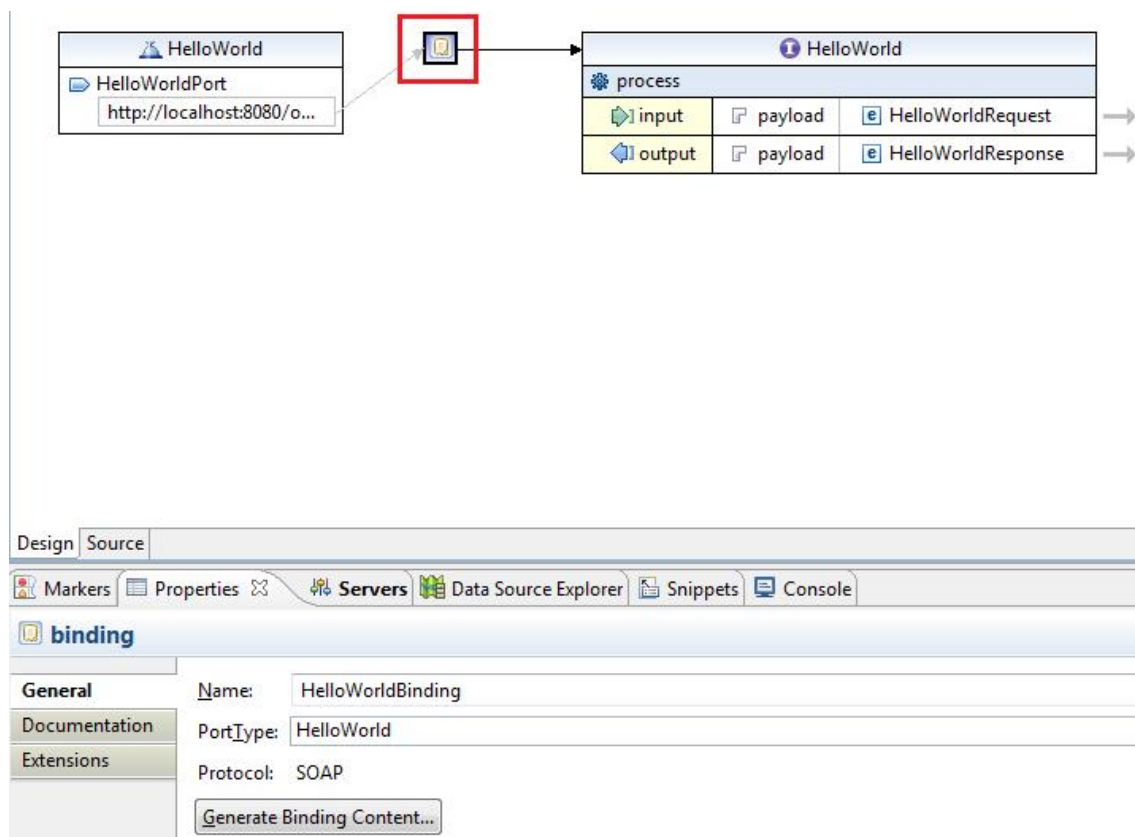
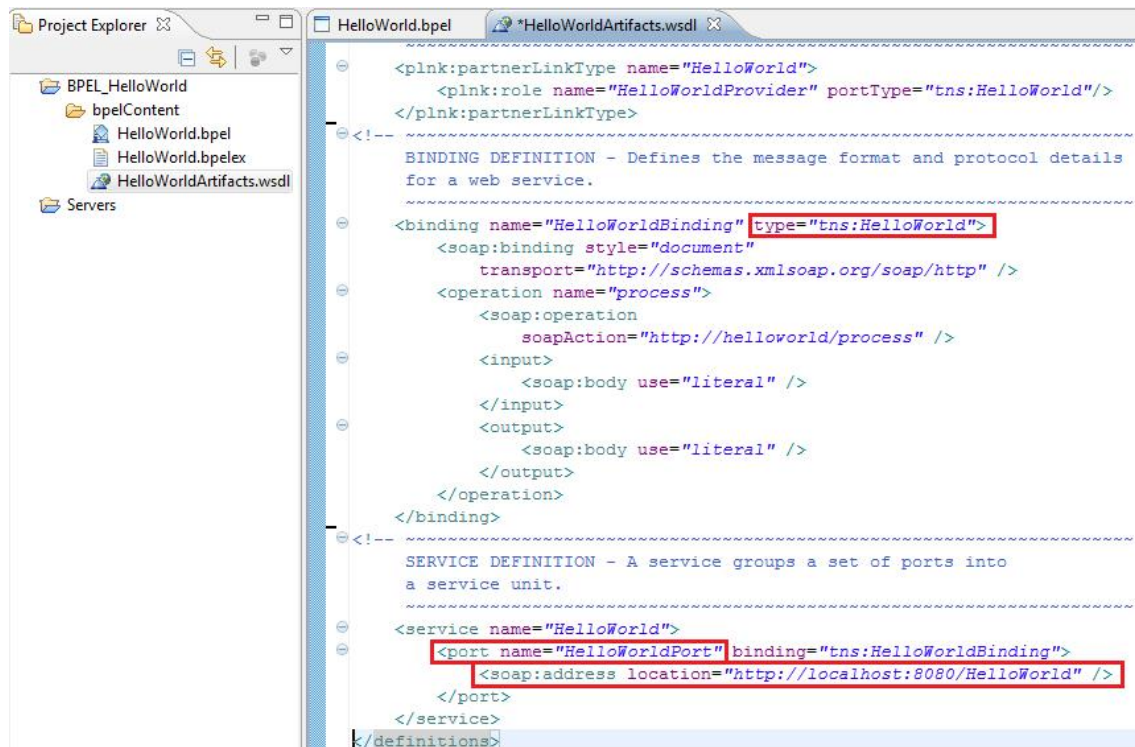


Y en el código XML podemos ver el código BPEL generado donde asigna en este caso...

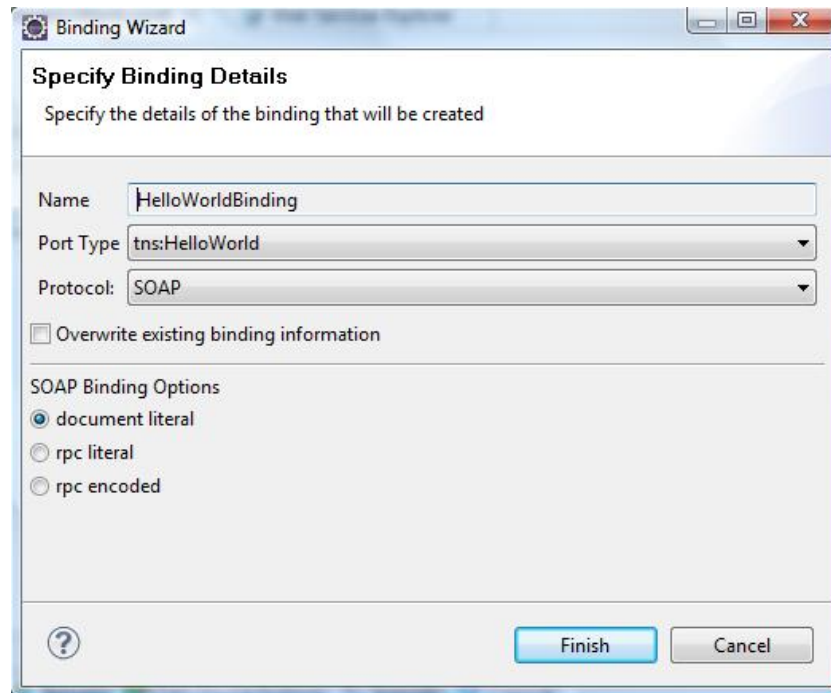
```
<bpel:assign validate="no" name="AssignHelloWorld">
  <bpel:copy>
    <bpel:from>
      <bpel:literal>
        <tns:HelloWorldResponse xmlns:tns="http://helloworld"
          <tns:result>tns:result</tns:result>
        </tns:HelloWorldResponse>
      </bpel:literal>
    </bpel:from>
    <bpel:to variable="output" part="payload"></bpel:to>
  </bpel:copy>
</bpel:assign>
```

1.16 Pulsando en la pestaña *source* en la parte inferior del espacio de modelado podemos ver el archivo BPEL generado.

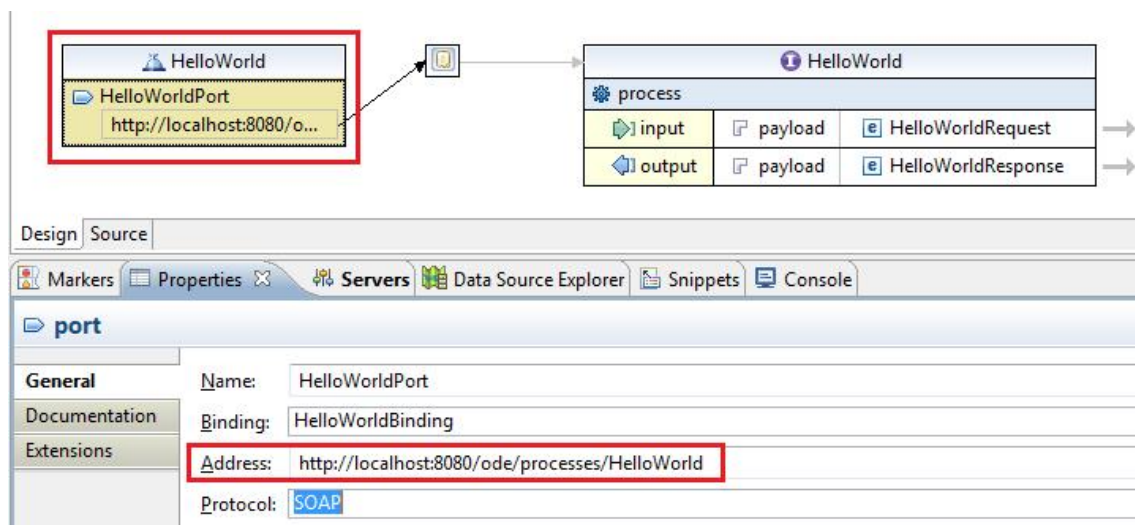
1.17 Una vez hemos diseñado el proceso podemos acceder al documento WSDL donde se define la interfaz de acceso al proceso donde se especifica el servicio, el tipo de enlace y la operación (*service*, *binding type* y *portType*) usando el editor WSDL en modo gráfico o texto.



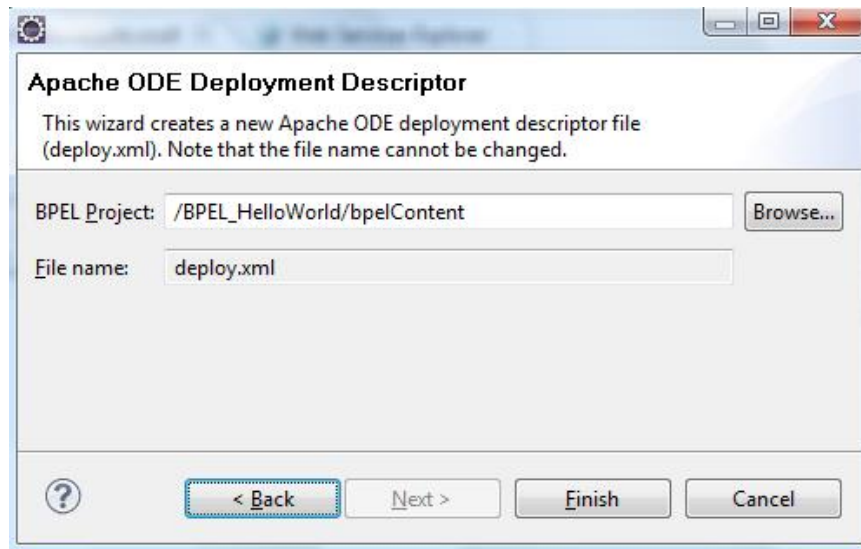
1.18 Podemos modificar el tipo de enlace según nuestras necesidades pulsando el botón *Generate Binding Content...*



- 1.19 Ahora, seleccionamos el elemento `service` en el editor gráfico y cambiamos el `address-location` del servicio, es decir, la URL de acceso al servicio sustituyéndola por <http://localhost:8080/ode/processes/HelloWorld>.



- 1.20 Ahora podemos realizar el despliegue y la ejecución del proceso. Para ello, creamos el archivo de despliegue para Apache ODE (sobre la carpeta `bpelContent`, pulsando el botón derecho, seleccionamos `New` → `Other` → `Bpel 2.0` → `Apache ODE Deployment Descriptor`).



- 1.21 Abrimos el archivo creado *deploy.xml* y en la sección *Inbound Interfaces* asociamos al *PartnerLink client* el *Port HelloWorld* lo que hará que el resto de campos de esa fila se completen.

General

This process is **activated**

☐ Run this process in memory

Inbound Interfaces (Services)

The table contains interfaces the process provides. Specify the service, port and binding you want to use for each PartnerLink listed

Partner Link	Associated Port	Related Service	Binding Used
client	HelloWorldPort	{http://helloworld}HelloWorld	HelloWorldBinding

Outbound Interfaces (Invokes)

The table contains interfaces the process invokes. Specify the service, port and binding you want to use for each PartnerLink listed

Partner Link	Associated Port	Related Service	Binding Used

Process-level Monitoring Events

☐ None
☒ All
☐ Selected

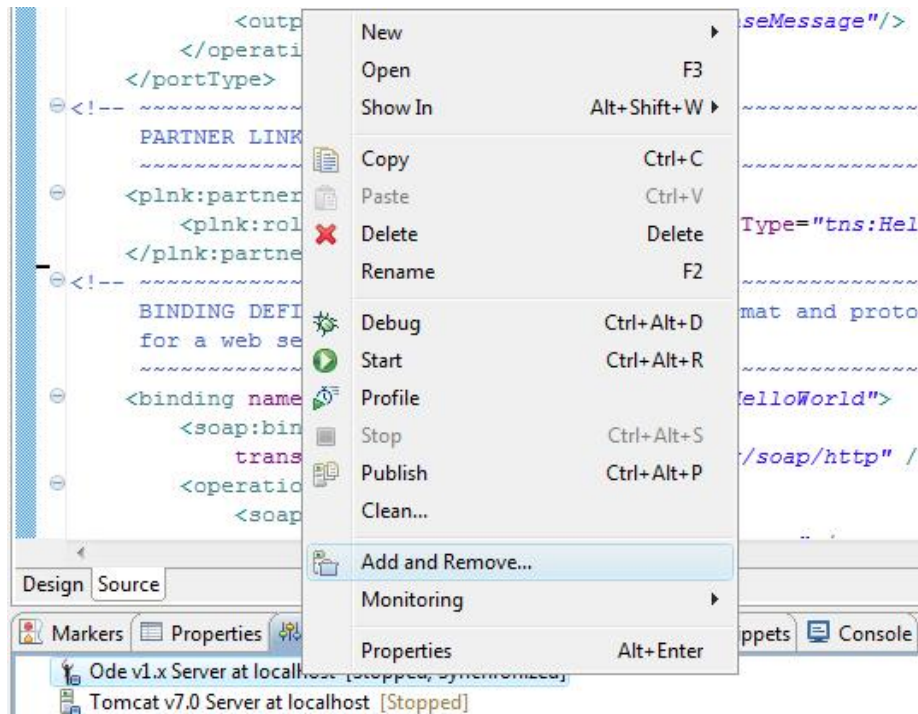
☐ Instance life cycle
☐ Activity life cycle
☐ Data handling
☐ Scope handling
☐ Correlation

Scope-level Monitoring Events

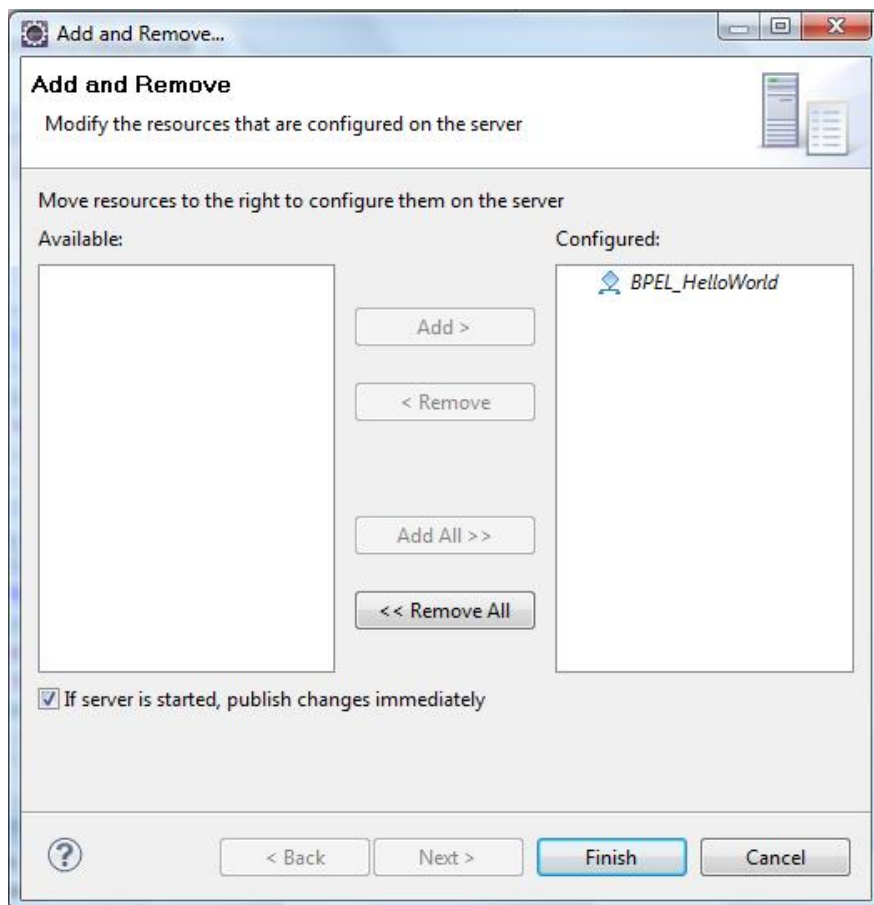
Scope	Instance life cycle	Activity life cycle	Data handling	Scope handling	Correlation
HelloWorld					

1.22 Guardamos los cambios.

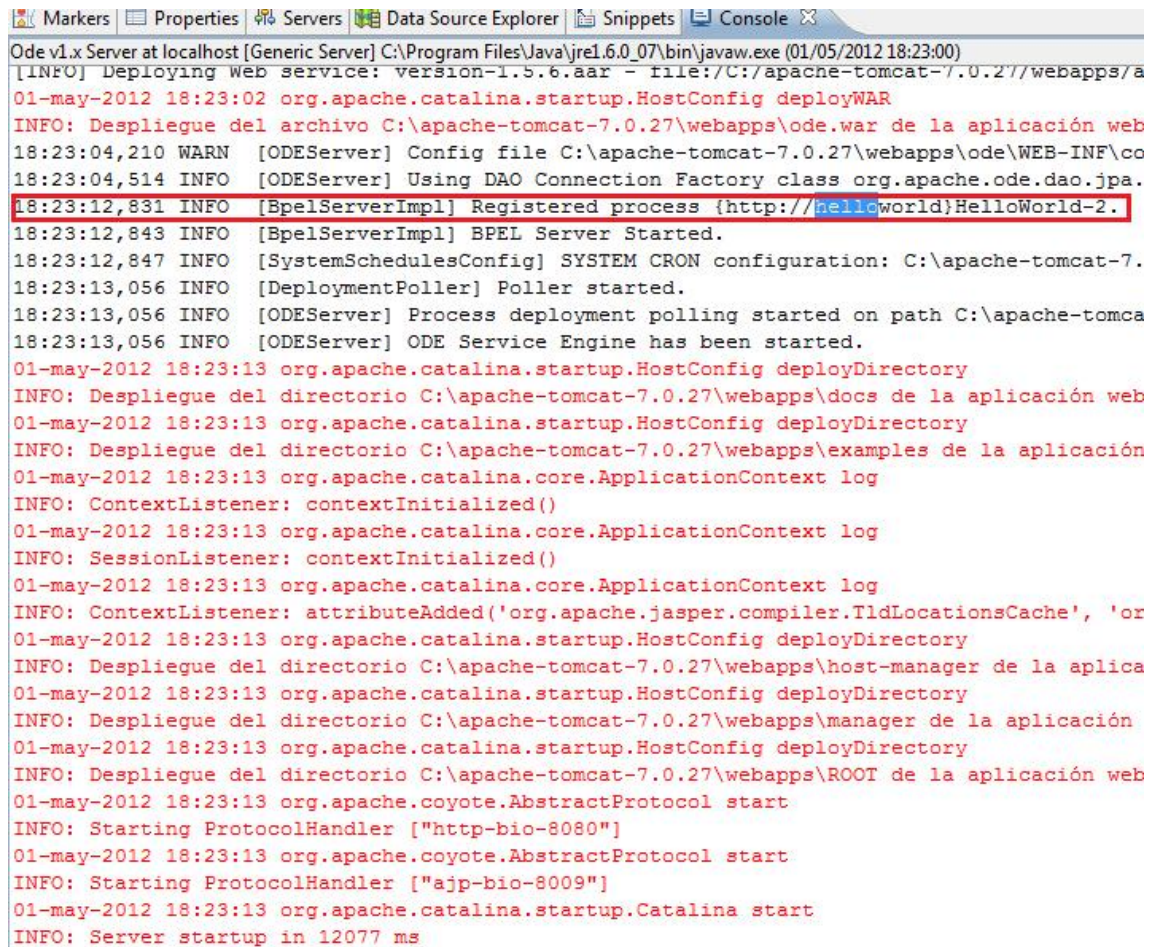
1.23 Pulsar sobre el servidor ODE en la pestaña server con el botón derecho seleccionando la opción *Add and Remove Projects*.



1.24 Selecciona el proyecto BPEL_HelloWorld y pulsa el botón *Add*.



- 1.25 Pulsamos el botón *Finish* y arrancamos el servidor Apache ODE. Podemos ver si el proceso ha sido registrado correctamente.



```
Markers Properties Servers Data Source Explorer Snippets Console X
Ode v1.x Server at localhost [Generic Server] C:\Program Files\Java\jre1.6.0_07\bin\javaw.exe (01/05/2012 18:23:00)
[INFO] Deploying Web service: version-1.5.6.aar - file:/C:/apache-tomcat-7.0.27/webapps/a
01-may-2012 18:23:02 org.apache.catalina.startup.HostConfig deployWAR
INFO: Despliegue del archivo C:\apache-tomcat-7.0.27\webapps\ode.war de la aplicación web
18:23:04,210 WARN [ODEServer] Config file C:\apache-tomcat-7.0.27\webapps\ode\WEB-INF\co
18:23:04,514 INFO [ODEServer] Using DAO Connection Factory class org.apache.ode.dao.jpa.
18:23:12,831 INFO [BpelServerImpl] Registered process {http://helloworld}HelloWorld-2.
18:23:12,843 INFO [BpelServerImpl] BPEL Server Started.
18:23:12,847 INFO [SystemSchedulesConfig] SYSTEM CRON configuration: C:\apache-tomcat-7.
18:23:13,056 INFO [DeploymentPoller] Poller started.
18:23:13,056 INFO [ODEServer] Process deployment polling started on path C:\apache-tomca
18:23:13,056 INFO [ODEServer] ODE Service Engine has been started.
01-may-2012 18:23:13 org.apache.catalina.startup.HostConfig deployDirectory
INFO: Despliegue del directorio C:\apache-tomcat-7.0.27\webapps\docs de la aplicación web
01-may-2012 18:23:13 org.apache.catalina.startup.HostConfig deployDirectory
INFO: Despliegue del directorio C:\apache-tomcat-7.0.27\webapps\examples de la aplicación
01-may-2012 18:23:13 org.apache.catalina.core.ApplicationContext log
INFO: ContextListener: contextInitialized()
01-may-2012 18:23:13 org.apache.catalina.core.ApplicationContext log
INFO: SessionListener: contextInitialized()
01-may-2012 18:23:13 org.apache.catalina.core.ApplicationContext log
INFO: ContextListener: attributeAdded('org.apache.jasper.compiler.TldLocationsCache', 'or
01-may-2012 18:23:13 org.apache.catalina.startup.HostConfig deployDirectory
INFO: Despliegue del directorio C:\apache-tomcat-7.0.27\webapps\host-manager de la aplica
01-may-2012 18:23:13 org.apache.catalina.startup.HostConfig deployDirectory
INFO: Despliegue del directorio C:\apache-tomcat-7.0.27\webapps\manager de la aplicación
01-may-2012 18:23:13 org.apache.catalina.startup.HostConfig deployDirectory
INFO: Despliegue del directorio C:\apache-tomcat-7.0.27\webapps\ROOT de la aplicación web
01-may-2012 18:23:13 org.apache.coyote.AbstractProtocol start
INFO: Starting ProtocolHandler ["http-bio-8080"]
01-may-2012 18:23:13 org.apache.coyote.AbstractProtocol start
INFO: Starting ProtocolHandler ["ajp-bio-8009"]
01-may-2012 18:23:13 org.apache.catalina.startup.Catalina start
INFO: Server startup in 12077 ms
```

- 1.26 Para invocar el proceso, seleccionamos el archivo WSDL
(*HelloWorldArtifact.wsdl*) desde SOAPUI y crearemos un nuevo proyecto SOAP.