

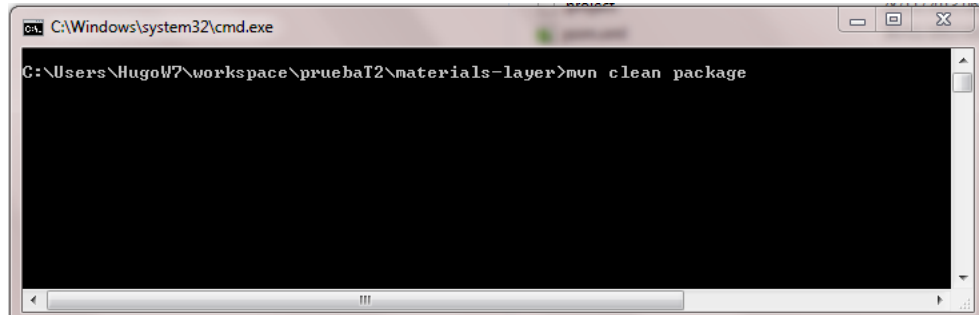
Tutorial prueba integración

December 1, 2013

1 Pasos para probar conectarse con integración

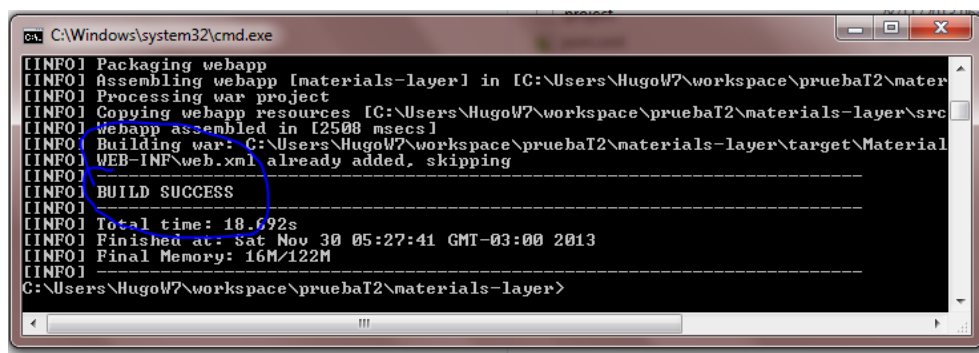
- Ejecutar:

```
mvn clean package
```



- Verificar:

BUILD SUCCESS



Si no llegan a esto me avisan lo antes posible

- Ejecutar:

```
mvn eclipse:eclipse -Dwtpversion=2.0
```

- Importar en eclipse este proyecto.
- este paso es para asegurarnos que anda todo bien. Con el eclipse abierto vuelven a la consola y ejecutan:

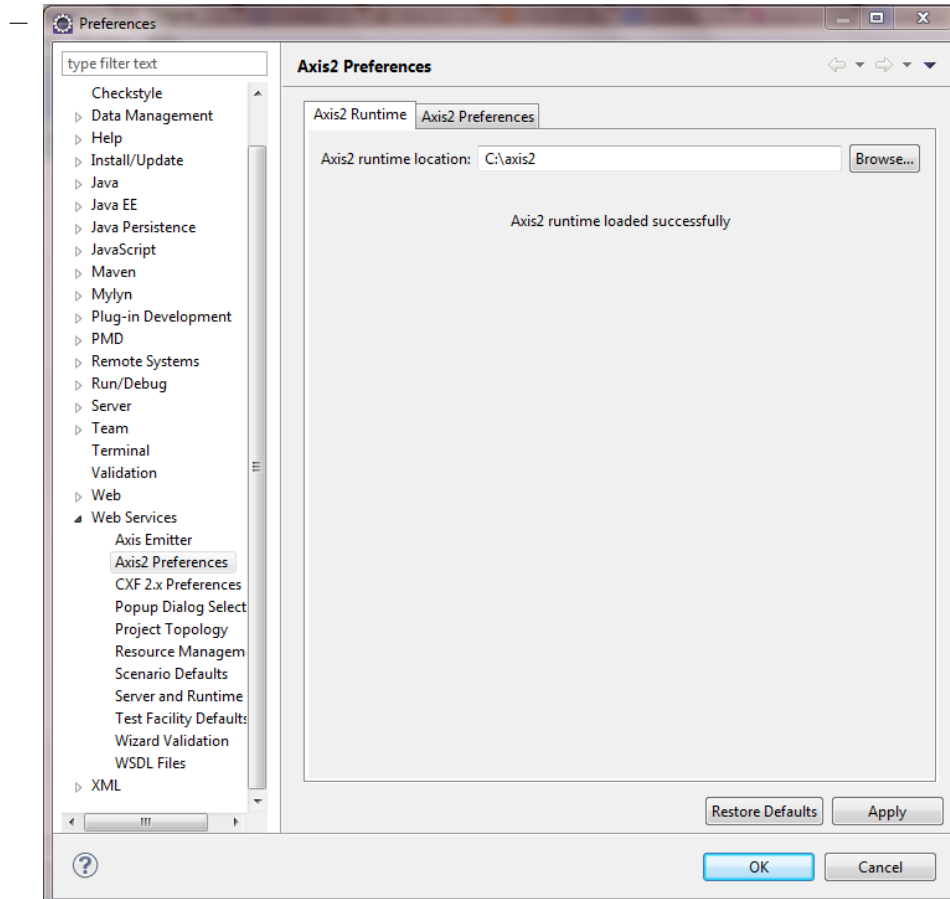
```
mvn eclipse:clean
```

Y en *eclipse* clic derecho sobre el proyecto *Configure* → *Convert to Maven Project*.

Luego, clic derecho sobre el proyecto *Maven* → *Update project..* (*Alt + F5*)

- (dami .. esto creo que yami ya lo tiene y con andy lo hicimos hace un rato) configuren axis2 en eclipse

- bajar
<http://apache.dattatec.com//axis/axis2/java/core/1.6.2/axis2-1.6.2-bin.zip>
- descomprimir
- seguir estas instrucciones
<http://efunctions.wordpress.com/2011/12/19/configurando-apache-axis2-en-eclipse-indigo/>

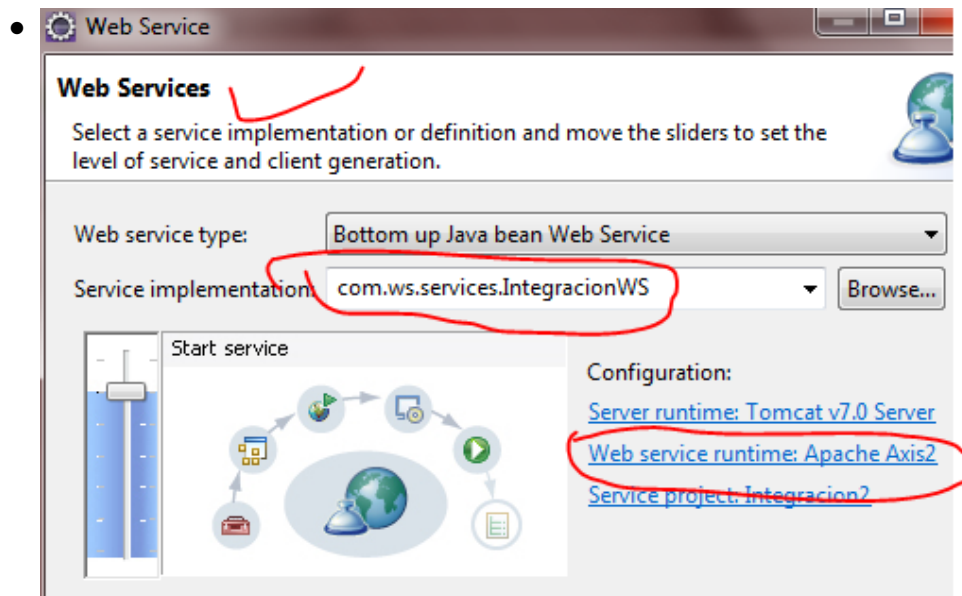


Así se ve el mío

- Crear un proyecto "dummy" para que eclipse se entere de que tiene axis2:
 - * File → New.. → Dynamic Web Project
 - * Project name: **"LoQueSea"**
 - * Dynamic web module version: **2.5**
 - * Configuration → Modify.. → tildar *Axis2 Web Services*
 - * Ok → Finish
- Detener (si existe y está corriendo por fuera del eclipse) el Tomcat, en linux se usa esto:

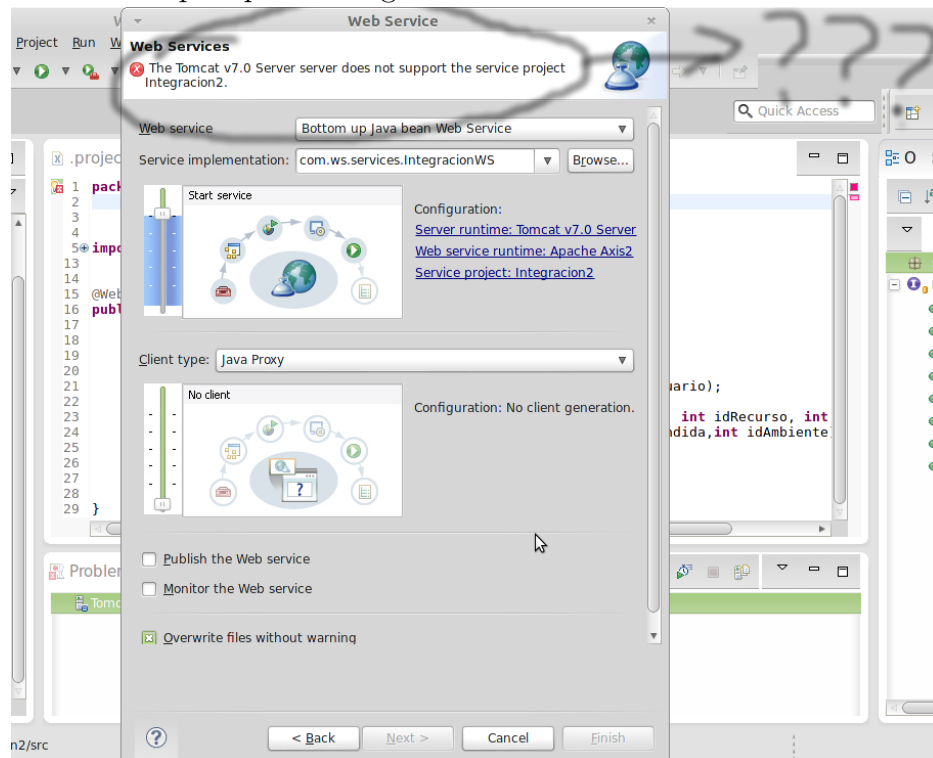

```
sudo /etc/init.d/tomcat7 stop
```
- Buscar dentro de *eclipse* el proyecto *SERVERS* y abrirlo.

- copien la carpeta Integracion2 que está en el repo dentro de su workspace de eclipse
- importen ese proyecto a eclipse
- en ese proyecto clic derecho sobre /com/ws/services/IntegracionWS.java
- New → Other..→ WebServices → Web Service



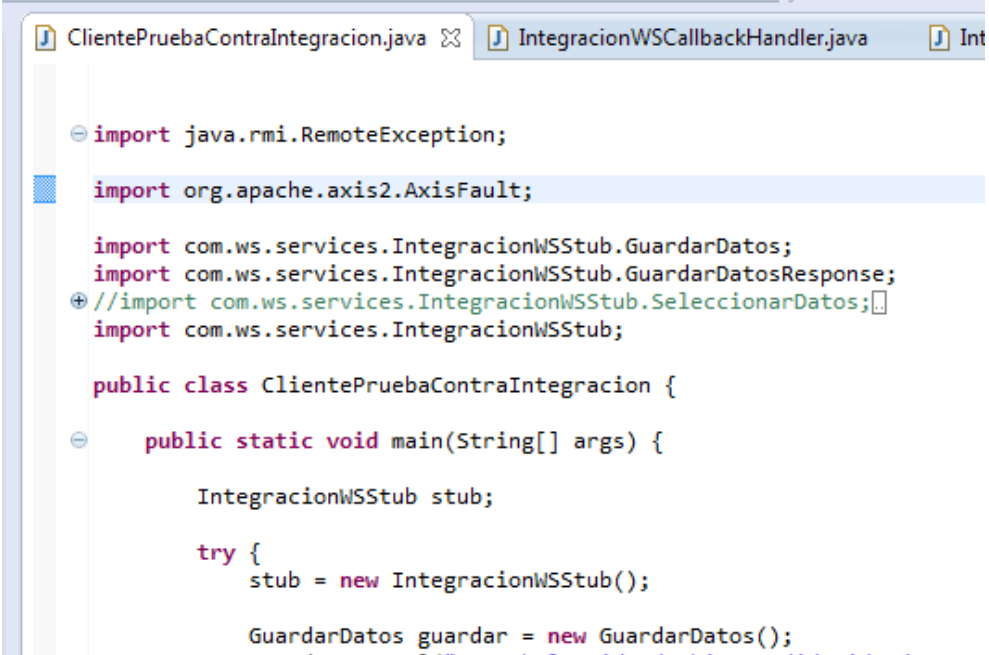
Así debería verse, tiene q decir *Apache Axis2!*
si no dice.. clic y seleccionar Axis2

- En caso de que aparezca algo asi:



Se deberán seguir los siguientes pasos: Click derecho sobre el proyecto de Integracion2 → properties → project facets→ dynamic web module y se reemplaza por 2.5

- dejar el resto como está y darle siguiente.. siguiente..
- probarlo...
http://localhost:8080/Integracion2/services/IntegracionWS?wsdl
- Ahora hay que probar el cliente



```
import java.rmi.RemoteException;
import org.apache.axis2.AxisFault;
import com.ws.services.IntegracionWSStub.GuardarDatos;
import com.ws.services.IntegracionWSStub.GuardarDatosResponse;
//import com.ws.services.IntegracionWSStub.SeleccionarDatos;
import com.ws.services.IntegracionWSStub;

public class ClientePruebaContraIntegracion {

    public static void main(String[] args) {

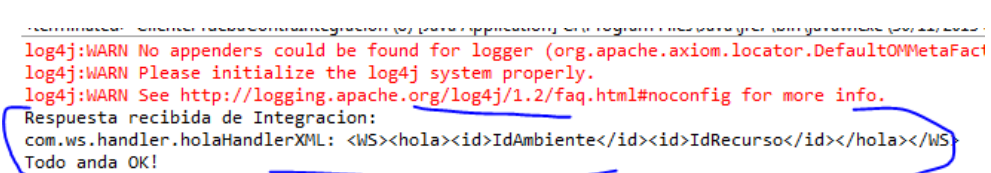
        IntegracionWSStub stub;

        try {
            stub = new IntegracionWSStub();

            GuardarDatos guardar = new GuardarDatos();
```

ejecutar como java application el archivo:

ClientePruebaContraIntegracion.java



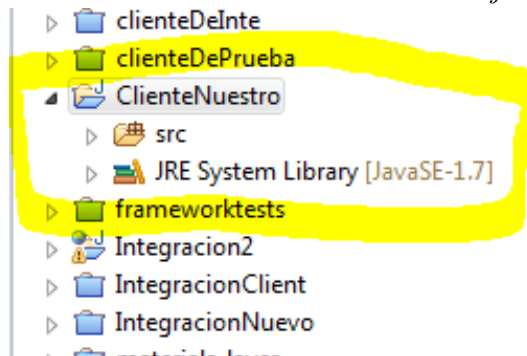
```
log4j:WARN No appenders could be found for logger (org.apache.axis2.transport.http.HTTPSender).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
Respuesta recibida de Integracion:
com.ws.handler.holaHandlerXML: <WS><hola><id>IdAmbiente</id><id>IdRecurso</id></hola></WS>
Todo anda OK!
```

les tiene que aparecer esto (*Caso contrario.. me avisan!*)

2 Generación de un cliente de nuestro Web Service

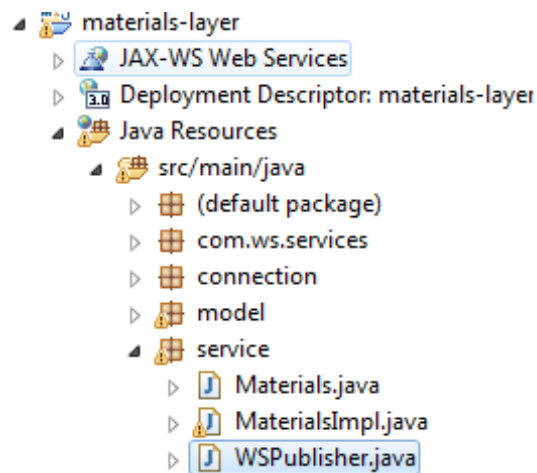
Se va a hacer un cliente simple en un *main* y se muestran algunas invocaciones de ejemplo.

- En *eclipse* se crea un proyecto común de Java y llamenlo "ClienteNuestro":
File → *New* → *Other..* → *Java Project*



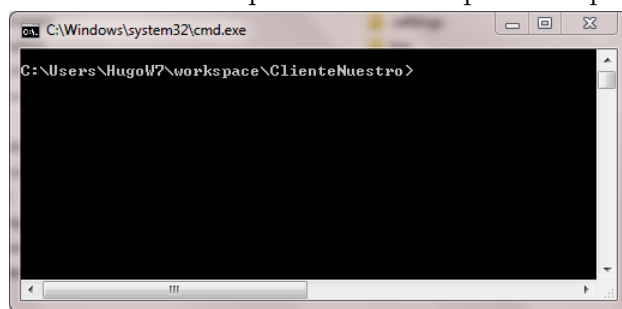
- Ir a nuestro proyecto y ejecutar como *Java application* la clase **WSPublisher** y tomar nota de la url en la cual se está publicando el servicio. En la consola se va a ver esto:

Ingresa "hola" para salir :



En caso de que dé algún error: puede ser que el puerto esté ocupado y haya que detener eso, aunque acá se puede poner manualmente cualquier puerto cambiando el parámetro del método `ENDPOINT.PUBLISH`.

- Dentro del workspace ir a la carpeta del proyecto y abrir un terminal ahí



- Ejecutar el siguiente comando para generar el código cliente (la url contiene parte de lo que dice el parámetro del método ENDPOINT.PUBLISH):

```
wsimport -s src -keep -verbose http://localhost:8081/WS2/Greeting2?wsdl
```

- En la pantalla se debe ver algo así

```
parsing WSDL...
Generating code...
```

```
service\AgregarEncuesta.java
service\AgregarEncuestaRespondida.java
service\AgregarEncuestaRespondidaResponse.java
service\AgregarEncuestaResponse.java
service\AgregarLink.java
service\AgregarLinkResponse.java
.... (todas las clases deben aparecer)
```

```
Compiling code...
```

```
javac -d ...
```

En caso de que dé algún error: puede ser que está mal la url o el servicio no está levantado

- En *eclipse* clic derecho sobre el proyecto "ClienteNuestro" → Refresh
- Crear una clase Client con este código (dejo la clase creada en la carpeta tutorial):

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.List;

import service.Encuesta;
import service.Materials;
import service.MaterialsImplService;
import service.Pregunta;

public class Client {

    public static void main(String[] args) {
        MaterialsImplService service = new MaterialsImplService();
        Materials port = service.getMaterialsImplPort();
        System.out.println("----->> Probando el servicio");
        System.out.println(port.sayHello("Programador"));
        System.out.println("----->> ESTA TODO OK!");

        System.out.println("Ingrese algo para continuar y probar las encuestas : ");

        try {
            BufferedReader bufferRead = new BufferedReader(new InputStreamReader(System.in));
            bufferRead.readLine();

        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

```

    }

    try {
        Encuesta enc = port.getEncuesta(0, 0);

        System.out.println("TITULO ENCUESTA: " + enc.getDescripcion());

        List<Pregunta> preguntas = enc.getPreguntas();
        for (Pregunta p : preguntas) {
            System.out.println("\t" + p.getEnunciado());
        }
        } catch (Exception e) {
            System.out.println("Encuesta inexistente!");
        }

    }

}

```