ANEXO B

Tutorial de Web Services JAX-WS en Java EE 5

Introducción

El presente tutorial es una adaptación del tutorial ubicado en el sitio web de NetBeans. Dirigido y adaptado para estudiantes de la Universidad del Cauca interesados en el la ejecución y consumo de servicios Web.

Objetivos

- Adquirir conocimientos básicos acerca de los Servicios Web
- Adquirir conocimientos básicos de cómo crear y consumir un servicio Web.

Prerrequisitos

- Conocimiento básico del lenguaje Java.
- Conocimiento básico de aplicaciones Web

Base Teórica

Los Servicios Web son componentes software independientes del lenguaje de implementación y por lo tanto aportan al usuario las funcionalidades que necesite a través de estándares web ampliamente conocidos

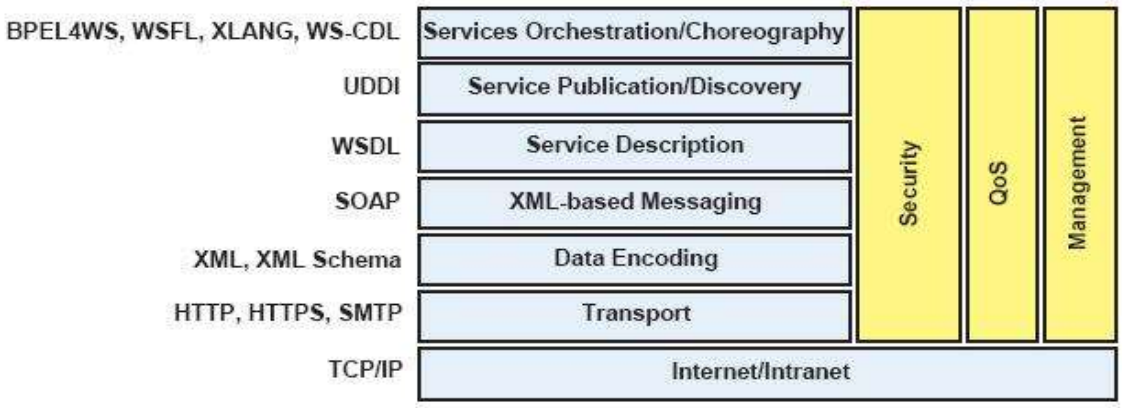


Figura 1. Arquitectura de los servicios web. Tomado de [YWL06]

En la figura se puede observar como los estándares SOAP (Protocolo Simple de Acceso a Objetos), WSDL (Lenguaje de Descripción de Servicios Web) y UDDI (Descripción, Descubrimiento e Integración Universal), los cuales están basados en XML (Lenguaje de Marcado Extensible) e Internet, constituyen el núcleo de los servicios web. Según lo dice [YWL06], SOAP define un protocolo simple basado en XML para el intercambio de estructuras de información en un entorno descentralizado y distribuido. Un mensaje SOAP es usado para que los usuarios accedan a los servicios web sin importar que plataforma ni que lenguaje estén utilizando. WSDL provee un formato XML para describir los servicios que están disponibles en la red exponiendo de ellos sus interfaces, las cuales poseen las operaciones que son capaces de realizar. UDDI es un conjunto de definiciones para un registro de servicios en donde la información de negocios, organizaciones, servicios Web disponibles e interfaces técnicas para su acceso es almacenada, incluyendo la definición de información para la publicación y descubrimiento de los mismos.

Un WSDL está constituido por 3 partes:

- **Los elementos types, message, y portType**: definen mensajes y tipos de datos intercambiados entre el cliente y el servidor. Un **message** es el elemento de comunicación básico de SOAP. Un mensaje puede constar de una o más partes, cada una de las cuales representa un parámetro con un tipo. Todos los mensajes se agrupan en operaciones en una entidad llamada portType. Un **portType** es la interfaz o conjunto concreto de operaciones soportadas por el Servicio Web. Un Servicio Web puede tener varias interfaces representadas por distintos portTypes.

- **Los elementos binding**: describen los detalles de la implementación técnica de un Servicio Web. Un binding permite unir un portType a un protocolo de comunicación como SOAP sobre HTTP. SOAP requiere de la especificación del estilo de comunicación utilizando para cada operación en el portType. SOAP soporta dos estilos de comunicación: RPC y Document. El estilo RPC soporta formación y deformación automática de mensajes, permitiendo a los desarrolladores expresar una solicitud como una llamada a método con un conjunto de parámetros, que devuelve una respuesta que contiene un valor de retorno. El estilo Document no soporta formación automática de mensajes. Asume que los contenidos del mensaje SOAP son datos XML bien-formados. SOAP requiere además de la especificación de la forma de los mensajes en XML. Se pueden utilizar valores tipo literal o tipos de datos encoded. El atributo *use='literal'* indica que el entorno de ejecución SOAP debería enviar el XML como se lo ha proporcionado. El atributo *use='encoded'* indica que el entorno de ejecución SOAP debería serializar los datos utilizando un estilo de codificación particular. Un estilo de codificación define un conjunto de reglas para expresar tipo de un lenguaje de programación en XML.

- **Los elementos service:** revisan el elemento service al final del documento WSDL. [PAL06]

En este tutorial se utilizará la especificación **JAX-WS** que corresponde al API de Java para Servicios Web basados en XML, tiene soporte para muchos protocolos tales como SOAP 1.1 y SOAP 1.2, XML y HTTP. Su especificación se encuentra descrita en el JSR 224. [JCP06]

Desarrollo del Tutorial

Software a utilizar

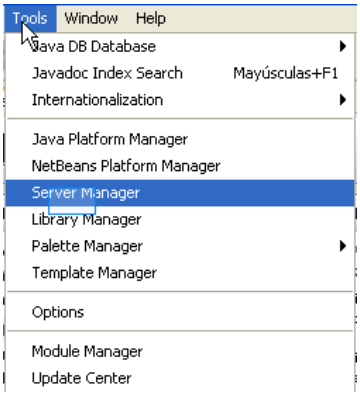
Para este tutorial se utilizará:

1. NetBeans 5.5 como IDE de desarrollo.
2. Java SDK Development Kit 6.0.
3. Sun Java System Application Server 9.0. (SJSAS de aquí en adelante).

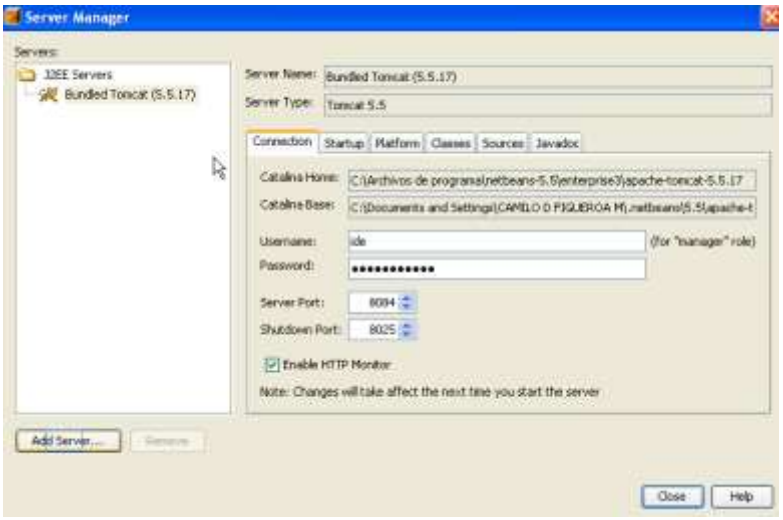
Adicionar un Servidor de Aplicaciones

Lo primero que se debe hacer es observar en NetBeans si ya está configurado el servidor de aplicaciones, en este caso el SJSAS, para se debe abrir NetBeans y seguir los siguientes pasos:

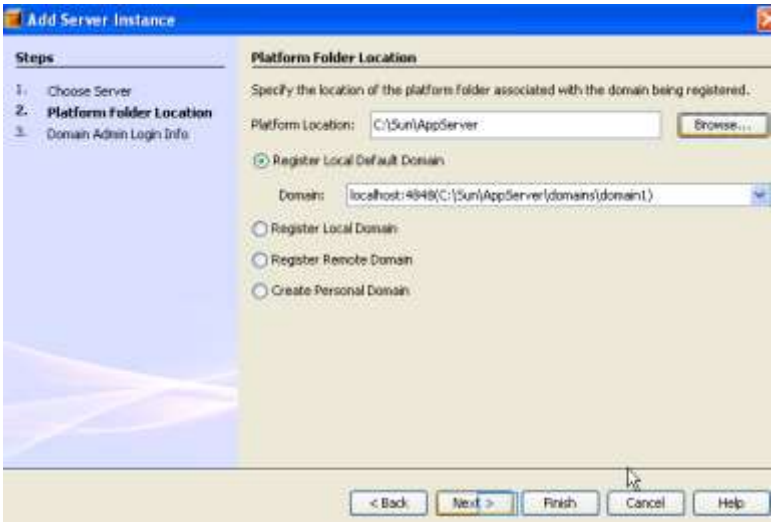
1. Elija el menú “Tools” y dentro de él haga clic en “Server Manager”.



2. Dentro del cuadro de diálogo “Server Manager” haga clic en el botón “Add Server” y seleccione “Sun Application Server”. Si lo desea coloque un nombre para esta instancia o deje el valor predeterminado. Luego haga clic en siguiente.



3. En este paso se abre la ventana “Add Server Instance” y luego haga clic en el botón “Browse” y seleccione el directorio donde se encuentra instalado el SJSAS.

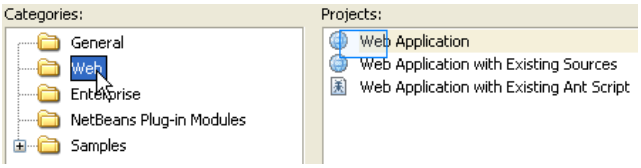


4. Haga clic en “Next” y agregue el nombre de usuario y la clave que le colocó al servidor cuando lo instaló. Haga clic en “Finish” y finalmente en “Close”.

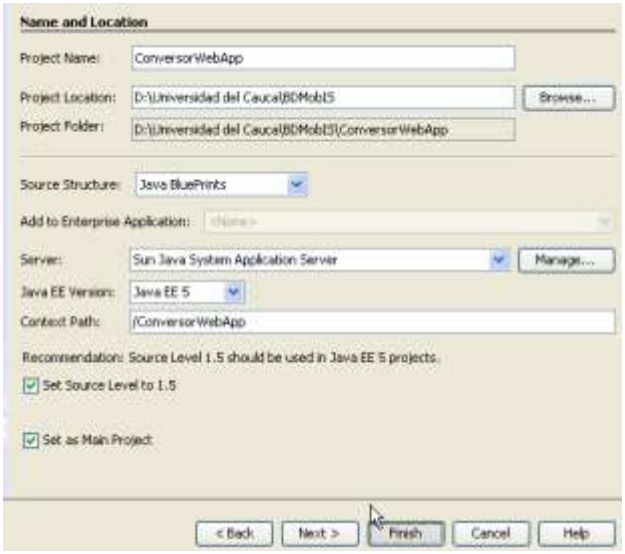
Crear un Servicio Web.

El primer paso para crear un Servicio Web es seleccionar un contenedor. En NetBeans para desplegar un servicio Web es necesario colocarlo dentro de un contenedor que bien puede ser un contenedor web o un contenedor EJB. En este tutorial se utilizará un contenedor Web.

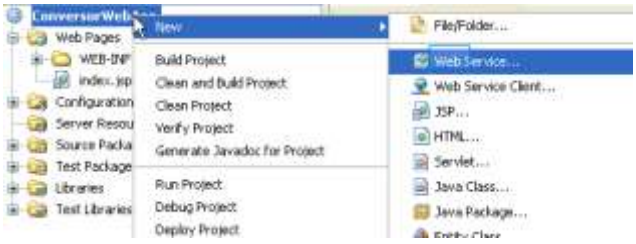
1. Haga clic en el menú “File”, seleccione “New Project” y en la ventana emergente en “categories” seleccione “Web” y en “Projects” seleccione “Web Application” y haga clic en “Next”.



2. En el campo de texto “Project Name” escriba “ConversorWebApp”. En “Server” seleccione el SJSAS. Seleccione como activos los cuadros de selección “Set Source Level 1.5” y “Set as Main Project” y haga clic en “Finish”.



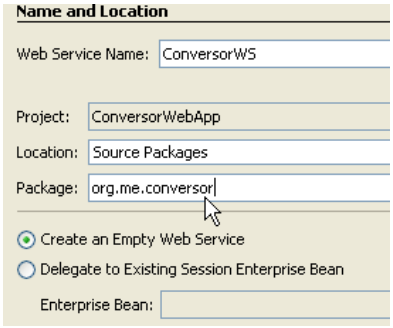
3. A continuación se crea el servicio Web dentro del contenedor, para esto haga clic derecho sobre el nodo “ConversorWebApp” y seleccione “New” y en el menú emergente haga clic sobre “Web Service”.



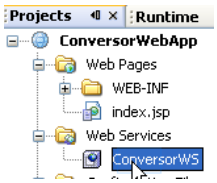
4. En la ventana emergente “New Web Service”:

    a. En el campo “Web Service Name” escriba “ConversorWS”.

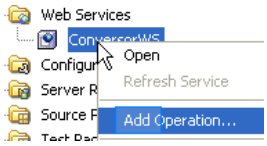
b. En el campo “package” escriba “org.me.conversor”. Y haga clic en el botón “Finish”.



5. La ventana del proyecto mostrará el nuevo servicio web. Dentro del árbol del proyecto expanda el nodo “Web Services” y observará que se encuentra el nuevo servicio web “ConversorWS”. A partir de este nodo del Servicio Web se crean las operaciones que realizará el servicio web, en este caso solo se crearan dos: una para convertir de grados Celsius a Fahrenheit y otra para realizar la operación inversa.



6. Haga clic derecho sobre el nodo “ConversorWS“y del menú emergente seleccione la opción “Add Operation”.



7. En el cuadro de diálogo emergente “Add Operation” seleccione:

   a. En el campo nombre escriba “celsiusToFarenheit”.

   b. En “Return Type” seleccione “float”.

   c. Dentro de la pestaña “Parameters” haga clic en el botón “Add”.

d. En el cuadro emergente “Enter Method Parameter”, en “Type” seleccione “float” y en “Name” escriba “celsius" y haga clic en “ok”. Finalmente haga clic en “OK”.

8. La ventana de código se actualizará, en la nueva pestaña “ConversosrWS.java”, dentro del código del método *celsiusToFahrenheit* copie el siguiente código.

```
@WebMethod
        public float celsiusToFahrenheit(@WebParam(name =
        "celcius") float celcius) {
            return (celcius*9/5)+32;}
```

9. Repita los pasos 6 a 8 para la operación *fahrengeithToCelcius*, y en el código coloque.

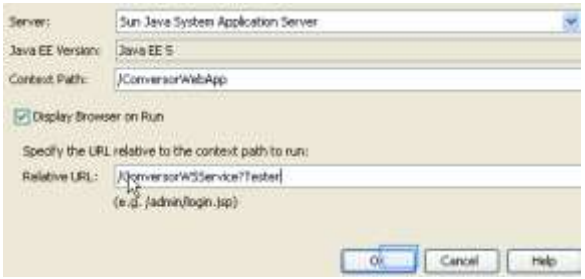
```
@WebMethod
      public float fahrenheitToCelcius(@WebParam(name =
      "farenheit") float farenheit) {
        return (farenheit - 32) * 5 / 9;}
```

Desplegar y Probar el Servicio Web

Ahora que ya se tiene el servicio Web creado solo resta desplegarlo sobre el servidor de aplicaciones, en este caso lo haremos sobre un contenedor web.

1. Haga clic derecho sobre el nodo del proyecto y seleccione “Properties” y a continuación “Run”.

2. En el campo “Relative URL” escriba “/ConversorWSService?Tester”. Finalmente Haga clic en “OK”



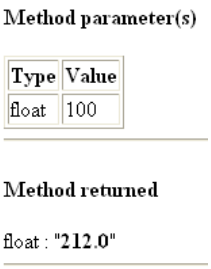
3. Si todo sale bien en la pantalla “Output” saldrá el siguiente mensaje.

```
All operations completed successfully
run-deploy:
Browsing:
http://localhost:8080/ConversorWebApp/ConversorWSService?Tester
run-display-browser:
run:
BUILD SUCCESSFUL (total time: 3 seconds)
```

4. Y en el navegador se desplegará una página de pruebas en la dirección: http://localhost:8080/ConversorWebApp/ConversorWSService?Tester

   En la figura se puede observar que la página de prueba tiene dos campos de texto con sus respectivos botones que corresponden a cada una de las funciones del Servicio Web. Para probarla ingrese un valor de 100 en el campo de “elciusToFarenheit” y haga clic en el botón que lleva ese nombre.

5. Observe que a continuación aparece el resultado de la operación junto con los mensajes SOAP intercambiados por el servicio web. A continuación se muestran esos mensajes.



Method parameter(s)

Type	Value
float	100

Method returned

float : "212.0"

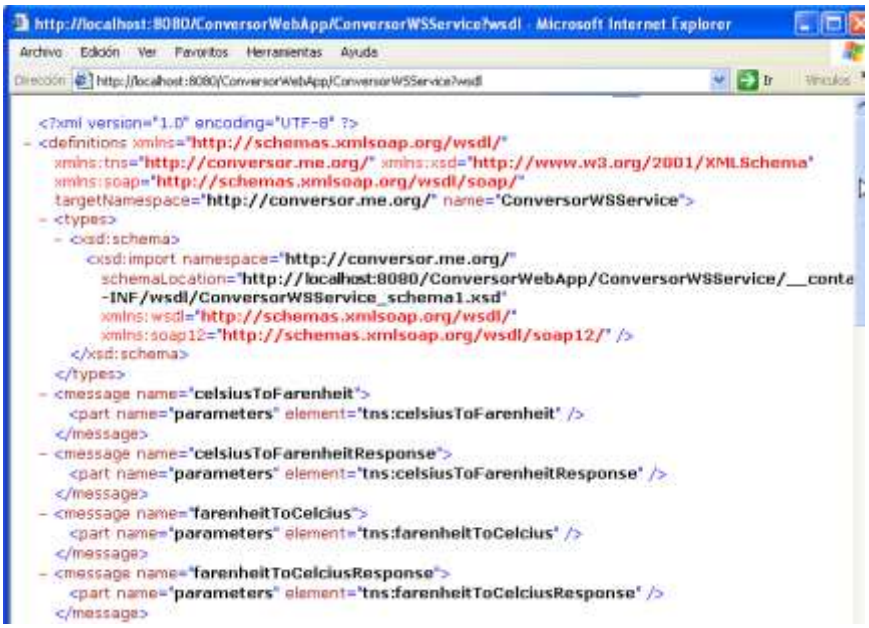
6. Los mensajes SOAP intercambiados son los siguientes:
   a. **SOAP Request.**

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:ns1="http://conversor.me.org/">
    <soapenv:Body>
        <ns1:celsiusToFarenheit>
            <celsius>100.0</celsius>
        </ns1:celsiusToFarenheit>
    </soapenv:Body>
</soapenv:Envelope>
```

   b. SOAP Response.

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:ns1="http://conversor.me.org/">
    <soapenv:Body>
        <ns1:celsiusToFarenheitResponse>
            <return>212.0</return>
        </ns1:celsiusToFarenheitResponse>
    </soapenv:Body>
</soapenv:Envelope>
```

7. Si se desea observar el WSDL del servicio Web se puede hacer a través de la **dirección:** http://localhost:8080/ConversorWebApp/ConversorWSService?wsdl



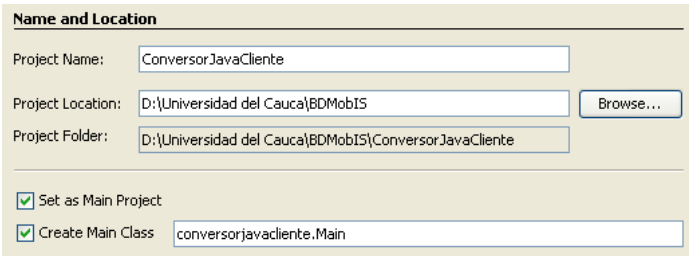
8. De igual manera se procede para la operación “farenheitToCelcius”.

Consumir el Servicio Web.

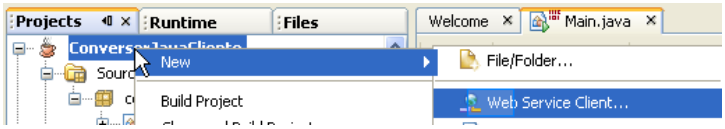
En este tutorial se desarrollarán tres clientes que consumirán el servicio Web, y corresponderán a tres aplicaciones diferentes que se ejecutan separadamente.

Cliente 1: Aplicación Cliente desarrollada en Java J2SE.

1. Haga clic en el menú “File” y seleccione “New Project”.En la ventana emergente seleccione “General” y seleccione “Java Application” en la sección “Projects” y haga clic en “Next”. En el campo de texto “Project Name” escriba “ConversorClienteJava” y haga clic en “Finish”.

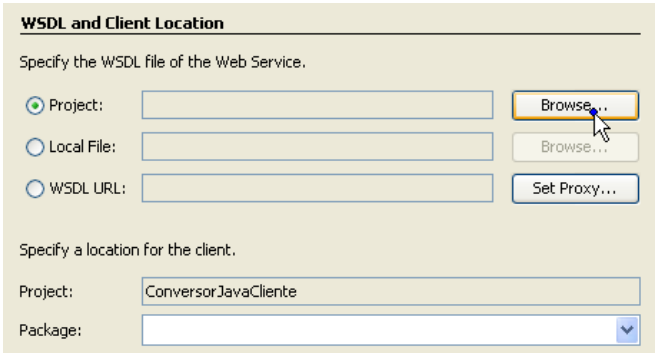


2. Haga clic derecho sobre el nodo del proyecto “ConversorJavaCliente”, seleccione “New” y en el menú emergente haga clic en “Web Service Client”.



3. En el cuadro emergente “New Web Service Cliente”, se debe especificar el WSDL del servicio a consumir lo cual se puede realizar de tres maneras.

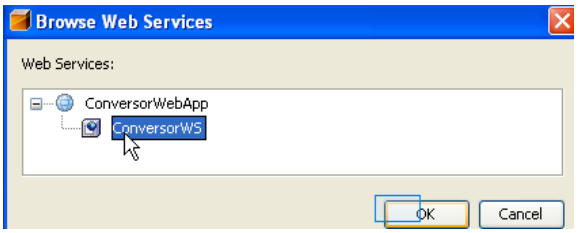
    a. *Seleccionando el archivo WSDL de uno de los proyectos activos*: En este caso seleccione el radio botón “Proyect” haga clic en el botón “Browse” y a continuación expanda el nodo “ConversorWebApp” y seleccione el servicio Web “ConversorWS” y haga clic en “OK”



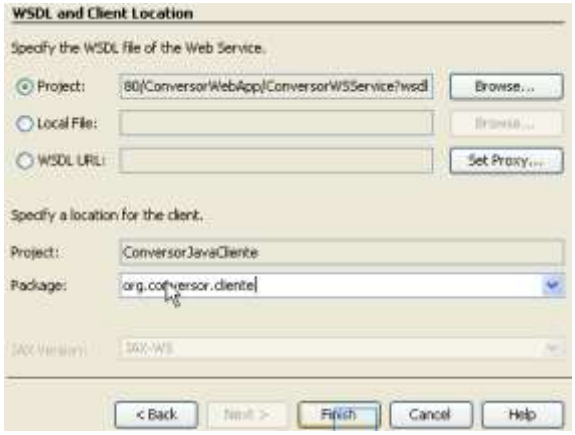
    b. Como un archivo Local, es decir que pertenece al proyecto activo. Seleccione el botón “Browse” y busque el archivo WSDL en su sistema de archivos.

    c. Con la URL del servicio Web: En este caso seleccione el radio botón “WSDL URL” y copie la URL del servicio web en el campo de texto. http://localhost:8080/ConversorWebApp/ConversorWSService?wsdl

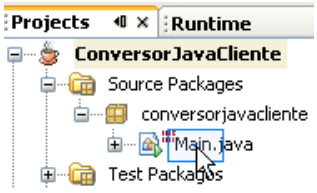
4. Para el presente proyecto en todos los clientes que se desarrollarán para consumir el servicio web se utilizará la primera forma de agregar la referencia web es decir la del literal “a” del anterior paso. De acuerdo a esto aparece una ventana “Browse Web Services” que indica los servicios web del proyecto. Expanda “ConversorWebApp”, seleccione ConversorWS y haga clic en OK.



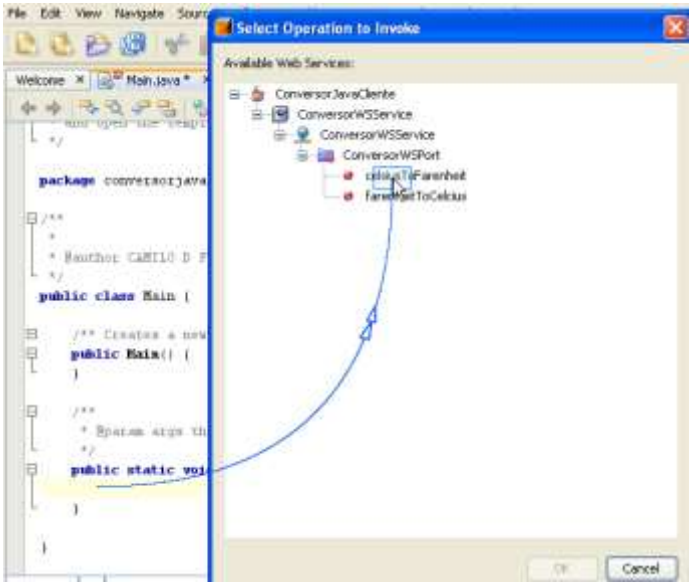
5. En “package” escriba “org.conversor.cliente” y haga clic en Finish.



6. Ahora en el proyecto “ConversorClienteJava” abra la clase “Main”.



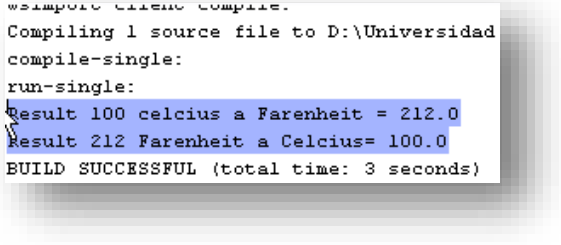
7. Busque el método “main” y una vez ahí haga clic derecho y seleccione “Web Services Cliet Resources” y a continuación haga clic en “Call Web Service Operation” y expanda el árbol respectivo al servicio Web hasta encontrar las operaciones y finalmente seleccione la operación que desea probar en la aplicación java. Haga clic en OK.



8. Haga clic en “OK”, y ahora se tiene listo un código que invoca a una operación del servicio web, solo resta que modifique los ítems que considere necesario. Ahora su código del método main debe quedar así.

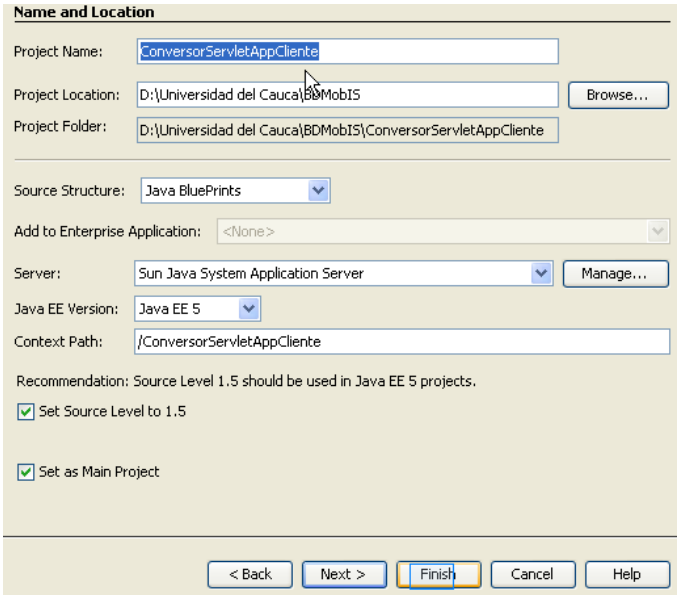
```
public static void main(String[] args) {
        try { // Call Web Service Operation
              org.conversor.cliente.ConversorWSService service = new
              org.conversor.cliente.ConversorWSService();
              org.conversor.cliente.ConversorWS port =
              service.getConversorWSPort();
              // TODO initialize WS operation arguments here
              float celsius = 100;
              // TODO process result here
              float result = port.celsiusToFarenheit(celsius);
              System.out.println("Result 100 celcius a Farenheit =
              "+result);
              } catch (Exception ex) {
              // TODO handle custom exceptions here
              }
              try { // Call Web Service Operation
              org.conversor.cliente.ConversorWSService service = new
              org.conversor.cliente.ConversorWSService();
              org.conversor.cliente.ConversorWS port =
              service.getConversorWSPort();
              // TODO initialize WS operation arguments here
              float farenheit = 212;
              // TODO process result here
              float result = port.farenheitToCelcius(farenheit);
              System.out.println("Result 212 Farenheit a Celcius=
              "+result);
              } catch (Exception ex) {
              // TODO handle custom exceptions here
              }
      }
```

9. Finalmente dentro del árbol del proyecto expanda el paquete org.conversor.cliente y haga clic derecho sobre la clase Main y luego clic en “Run”. Haga clic en la consola de eclipse y observe los resultados de la invocación del servicio Web.



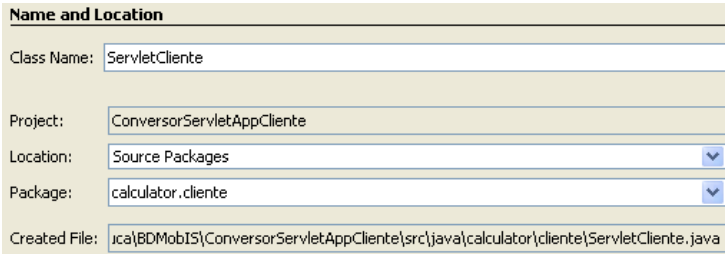
Cliente 2: Un Servlet en una Aplicación Web.

1. De manera similar al anterior cliente, Haga clic en el menú “File” y seleccione “New Project”. En la ventana emergente seleccione “Web” y seleccione “Web Application” en la sección “Projects” y haga clic en “Next”. En el campo de texto “Project Name” escriba “ConversorServletAppCliente” y haga clic en “Finish”.

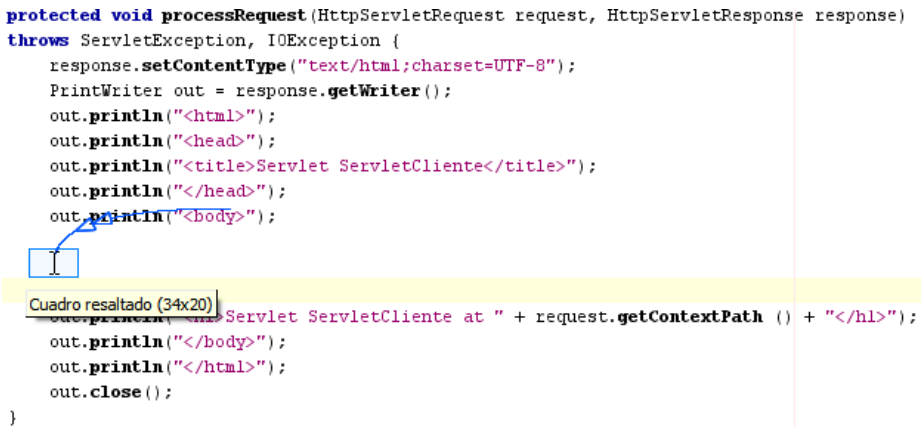


2. Realice los pasos 2-5 del anterior cliente.
3. Una vez que se tiene las referencias web agregadas al proyecto haga clic derecho en el nodo del proyecto y seleccione “New” y a continuación “Servlet”.

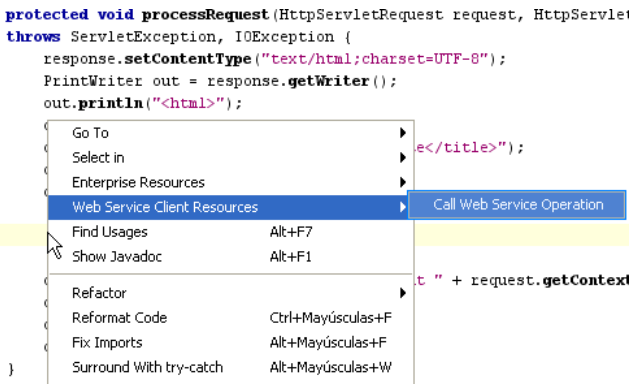
4. En el cuadro de diálogo emergente, en el En el campo “name” escriba “ServletCliente” y en el paquete coloque “org.servlet.cliente y haga clic en “Finish”.



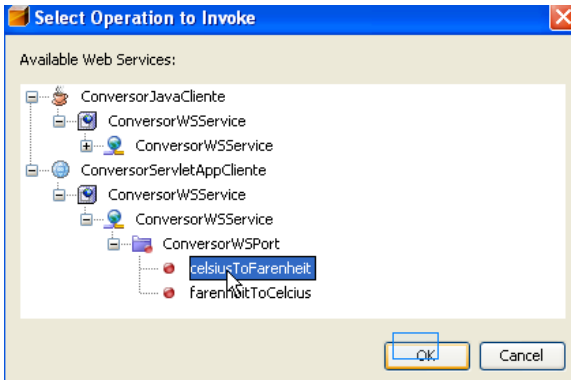
5. En la clase ServletCliente.java, elimine la línea de inicio de comentario del cuerpo de “processRequest”. Es decir la línea “/* TODO output your page here” y luego la línea de fin de comentario “*/”.Agregue algunos espacios en blanco antes de la línea “out.println("<h1>Servlet ClientServlet at " + request.getContextPath () + "</h1>");”



6. Ahora haga clic derecho sobre una de esas líneas vacías, agregadas en el paso anterior, y elija “”Web Service Client” y luego “Call Web Service Operation”.



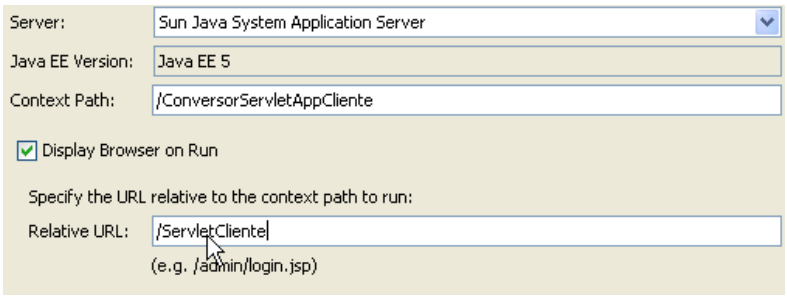
7. En el cuadro de diálogo emergente seleccione la operación que desea invocar. No olvide cambiar los valore float Celsius a 100 y float Fahrenheit a 212



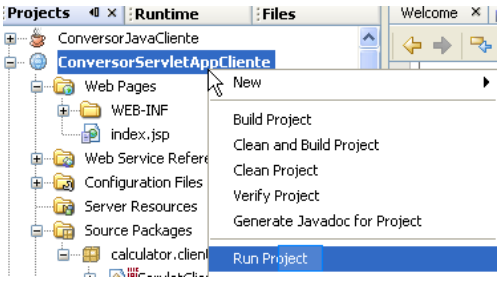
8. Su código debe quedar así:

```
protected void processRequest(HttpServletRequest request,
HttpServletResponse response)
    throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Servlet ServletCliente</title>");
        out.println("</head>");
        out.println("<body>");
        try { // Call Web Service Operation
            org.conversor.cliente.ConversorWS port =
service.getConversorWSPort();
            // TODO initialize WS operation arguments here
            float celsius = 100;
            // TODO process result here
            float result = port.celsiusToFarenheit(celsius);
            out.println("Result = "+result);
        } catch (Exception ex) {
            // TODO handle custom exceptions here
        }
        try { // Call Web Service Operation
            org.conversor.cliente.ConversorWS port =
service.getConversorWSPort();
            // TODO initialize WS operation arguments here
            float farenheit = 212;
            // TODO process result here
            float result = port.farenheitToCelcius(farenheit);
            out.println("Result = "+result);
        } catch (Exception ex) {
            // TODO handle custom exceptions here
        }
        out.println("<h1>Servlet ServletCliente at " +
request.getContextPath () + "</h1>");
        out.println("</body>");
        out.println("</html>");
        out.close();
    }
```

9. Haga clic derecho sobre el nodo del proyecto y seleccione properties, luego en run y en el campo “relative URL” escriba /ServletCliente.y clic en “OK”.



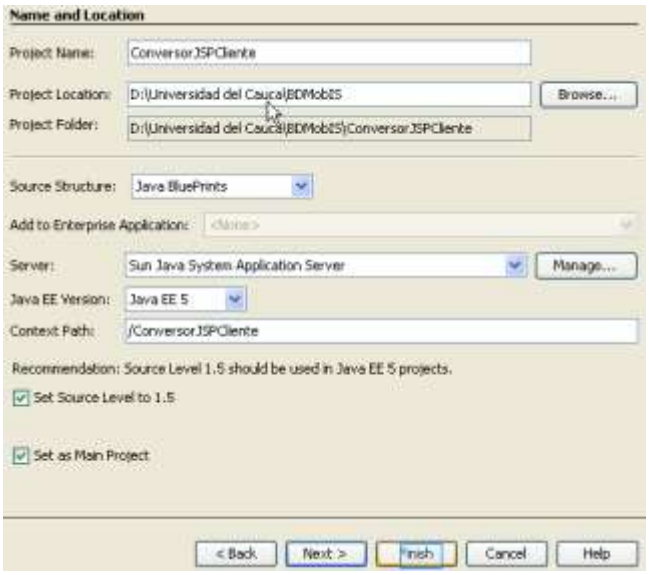
10. Finalmente haga clic derecho sobre el nodo del proyecto y luego haga clic en Run Project.



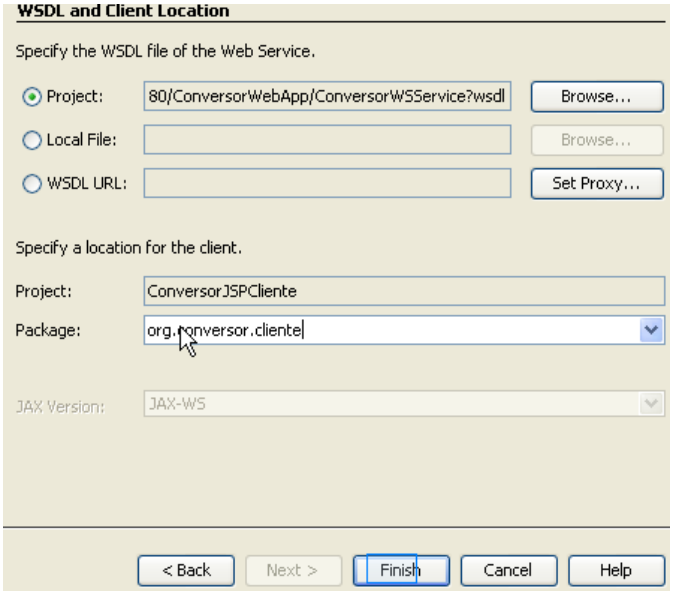
11. Abra el navegador y escriba la siguiente dirección: http://localhost:8080/ConversorServletAppCliente/ServletCliente obtendrá la siguiente página donde se pueden visualizar los resultados.

Cliente 3. Página JSP en una Aplicación Web.

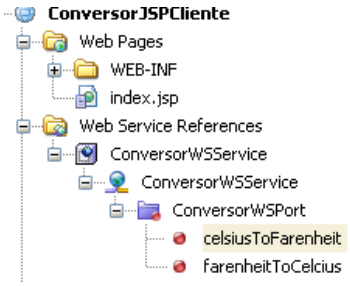
1. Como en los anteriores clientes: seleccione “File” luego “New Project”. Seleccione Web Application en la categoría Web category. Nombre el proyecto como ConversorJSPCliente. Haga clic en Finish.



2. Haga clic derecho sobre el nodo del proyecto ConversorJSPCliente y seleccione New > Web Service Client. En el proyecto haga clic en Browse y seleccione el servicio Web que va a consumir de la misma manera que se hizo en los anteriores clientes, haga clic en OK. En package escriba “org.calculator.cliente” y haga clic en Finish Y finalmente haga clic “Finish”.



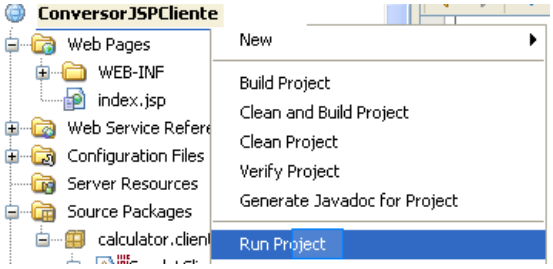
3. La ventana del proyecto mostrará el Nuevo cliente del servicio Web.



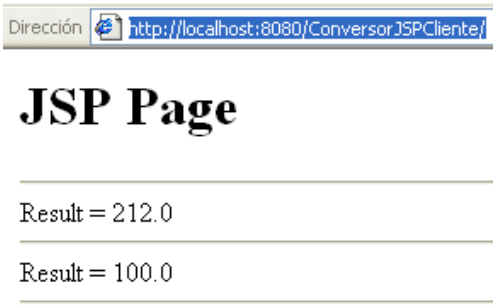
4. En la carpeta Web Pages haga doble clic en index.jsp para abrir el editor. En el nodo Web Services References, busque el nodo que representa al servicio Web. Luego arrastre la operación que desee justo bajo el tag H1 de la página index.jsp. Modifique el código añadido como en el caso anterior.

```
        org.conversor.cliente.ConversorWSService se
        org.conversor.cliente.ConversorWS port = se
         // TODO initialize WS operation arguments
        float celsius = 100;
        // TODO process result here
        float result = port.celsiusToFarenheit(cels
        out.println("Result = "+result);
    } catch (Exception ex) {
        // TODO handle custom exceptions here
    }
    %>
        <%-- start web service invocation --%><hr/>
    <%
    try {
        org.conversor.cliente.ConversorWSService se
        org.conversor.cliente.ConversorWS port = se
         // TODO initialize WS operation arguments
        float farenheit = 212;
        // TODO process result here
        float result = port.farenheitToCelcius(fare
        out.println("Result = "+result);
    } catch (Exception ex) {
```

5. Haga clic derecho sobre el proyecto y elija Run Project.



6. Abra un navegador web e ingrese la siguiente URL: http://localhost:8080/ConversorJSPCliete/.



BIBLIOGRAFÍA

[YWL06] Yushi, C; Wah, L; Limbu, D. Web Services Composition – An Overview of Standards. Singapore Institute of Manufacturing Technology Lee Eng Wah, Chairman of the Information Exchange Technical Committee

[PAL06] Palos J. Introducción a los Servicios Web. En internet:

http://www.programacion.com/java/tutorial/servic_web/1/

[JCP06] JSR 224: Java API for XML-Based Web Services (JAX-WS)2.0. Enl internet:

http://jcp.org/en/jsr/detail?id=224

[NET07] Documentación IDE NetBeans en internet: http://www.netbeans.org/kb/index.html