

¿Qué ofrece Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.
 Ese apoyo que siempre quiso tener...

1. Desarrollo de componentes y proyectos a medida



2. Auditoría de código y recomendaciones de mejora

3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



4. Cursos de formación (impartidos por desarrolladores en activo)

Spring MVC, JSF-PrimeFaces /RichFaces,
 HTML5, CSS3, JavaScript-jQuery

Gestor portales (Liferay)
 Gestor de contenidos (Alfresco)
 Aplicaciones híbridas

Tareas programadas (Quartz)
 Gestor documental (Alfresco)
 Inversión de control (Spring)

Control de autenticación y
 acceso (Spring Security)
 UDDI
 Web Services
 Rest Services
 Social SSO
 SSO (Cas)

JPA-Hibernate, MyBatis
 Motor de búsqueda empresarial (Solr)
 ETL (Talend)

Dirección de Proyectos Informáticos.
 Metodologías ágiles
 Patrones de diseño
 TDD

BPM (jBPM o Bonita)
 Generación de informes (JasperReport)
 ESB (Open ESB)



Empieza el jaleo: publicamos el libro
INFORMÁTICA PROFESIONAL
 LAS REGLAS NO ESCRITAS PARA TRIUNFAR EN LA EMPRESA

-¿Crees que la informática es sólo programar?
 -¿Sabrías organizar eficientemente un equipo?
 -¿Quién te resuelve las dudas sobre esta profesión?..... o te siembra más?

[Inicio](#) [Quienes somos](#) [Tutoriales](#) [Formación](#) [Comparador de salarios](#) [Comentar libro](#) [Charlas](#) [Más](#)

Estas en: [Inicio](#) [Tutoriales](#) Transformación de mensajes en SOA con OpenESB

Últimas Noticias

- » Nuestros tutoriales alcanzan la cifra de 10 millones de visitas!!!
- » Publicado el primer libro de TOD en castellano de Carlos Blé.
- » Comentando: Esta no es mi empresa: El desapego de los profesionales del siglo XXI de Ignacio Muro Benayas.
- » Impresiones y video de asistentes al primer coding-dojo de agilismo.es
- » Tarta y 7º aniversario de AdictosAITrabajo.com
- » Formación por calendario
- » Comentando el libro: Eneagrama y éxito personal de Ginger Lapid-Bogda
- » "Informática profesional: Las reglas no escritas para triunfar en la empresa." [Pincha para comprar.](#)
- » "¿?": Disponible la primera versión de los plugins para integrar Maven y Bugzilla.

+ Noticias Destacadas

- » Impresiones y video de asistentes al primer coding-dojo de agilismo.es
- » "Informática profesional: Las reglas no escritas para triunfar en la empresa." [Pincha para comprar.](#)
- » Autentia patrocina el primer Agile Open Spain
- » Lanzamiento del nuevo Web de Autentia

+ Comentarios Cómic

+ Enlaces

Tutorial desarrollado por



Iván García Puebla

Consultor tecnológico de desarrollo de proyectos informáticos.

Puedes encontrarme en [Autentia](#)

Somos expertos en Java/JEE

Catálogo de servicios de Autentia

[Descargar \(6,2 MB\)](#)

[Descargar en versión comic \(17 MB\)](#)

AdictosAITrabajo.com es el Web de difusión de conocimiento de Autentia.



[Catálogo de cursos](#)

Descargar este documento en formato PDF: [transformacion-mensajes-openESB.pdf](#)

Fecha de creación del tutorial: 2010-01-29

Transformación de mensajes en SOA con OpenESB

Índice

1. **Transformación de mensajes en SOA con OpenESB**
 1. Introducción
 2. Ejemplo: transformación sintáctica de una misma entidad
 3. Implementación con XSLT Module de OpenESB
 4. Proceso de transformación con XSLT
 5. Desplegar y probar el proyecto en OpenESB
 6. Conclusión

Introducción

En este tutorial vamos a mostrar con un ejemplo práctico una de las capacidades esenciales de todo sistema de integración, y por tanto característica del Enterprise Service Bus de SOA: la transformación de mensajes.

Los mensajes que viajan entre distintas partes de una infraestructura pueden ser transformados y manipulados durante su enrutamiento para adaptarse a las necesidades lógicas y físicas de la misma. Así, y en arquitecturas orientadas a servicios, a un mensaje se le pueden aplicar mecanismos (estándar o propietarios) de seguridad, direccionamiento, composición, filtrado, etc., y transformación, que es nuestro caso. Partiremos del entorno descrito en el tutorial [OpenESB 2.1. Instalación e introducción al entorno](#) y se recomienda tener conocimientos de OpenESB, definición de WSDL, XML Schema y XSLT.

El código fuente del tutorial y el proyecto de prueba soapUI están disponibles aquí: [xsitmodule_adictosaltrabajo.com.zip](#)

Ejemplo: transformación sintáctica de una misma entidad

En el ejemplo que vamos a desarrollar utilizaremos dos representaciones de una misma entidad de un pedido. Cada pedido tiene asociado una persona de contacto y unas especificaciones de material solicitado. Una representación se corresponde con las unidades de medida y nombres de familia de un país como España:

- Pedido
 - Contacto del proveedor:
 - Nombre y apellidos - tipo cadena
 - Teléfono de contacto - tipo cadena
 - Material de construcción
 - Identificador - tipo cadena
 - Peso máximo en kilogramos - tipo decimal, unidades de peso: kilogramos

Esta estructura la definimos en un fichero de XML Schema `pedidoSchema.xsd`:

Hosting patrocinado por [enredados](#)



Catálogo de servicios Autentia (PDF 6,2MB)



[En formato comic...](#)

Acceso de usuarios registrados:

E-mail:

Contraseña:

[Deseo registrarme](#)

[He olvidado mis datos de acceso](#)

Registra tu empresa:

Descubre las ventajas de registrar tu empresa en AdictosAITrabajo...

[Registrar mi empresa](#)

[Listado de empresas ya registradas](#)



Web

[www.adictosaltrabajo.com](#)

Últimos tutoriales

2010-01-29

[Transformación de mensajes en SOA con OpenESB](#)

2010-01-26

[JMeter. Uso de funciones.](#)

2010-01-18

[Autenticando los usuarios de Sonar contra un LDAP](#)

2010-01-18

[Introducción a jQuery UI.](#)

2010-01-18

[jQuery: cómo crear nuestros propios plugins.](#)

2010-01-18

[Cómo consumir un servicio web RESTful con el soporte de Ajax y JSON de jQuery.](#)

2010-01-18

[Introducción a jQuery.](#)

2010-01-17

Ads by Google

```

view plain print ?
01. <?xml version="1.0" encoding="UTF-8"?>
02. <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
03. targetNamespace="http://tutorial.adictosaltrabajo.com/schema/pedido
04. xmlns:tns="http://tutorial.adictosaltrabajo.com/schema/pedido
05. elementFormDefault="qualified">
06. <xsd:element name="pedido">
07. <xsd:complexType>
08. <xsd:sequence>
09. <xsd:element name="contacto" type="tns:contacto"/>
10. <xsd:element name="material" type="tns:material"/>
11. </xsd:sequence>
12. </xsd:complexType>
13. </xsd:element>
14. <xsd:complexType name="contacto">
15. <xsd:sequence>
16. <xsd:element name="nombre" type="xsd:string"/>
17. <xsd:element name="apellidos" type="xsd:string"/>
18. <xsd:element name="telefono" type="xsd:string"/>
19. </xsd:sequence>
20. </xsd:complexType>
21. <xsd:complexType name="material">
22. <xsd:sequence>
23. <xsd:element name="ID" type="xsd:int"/>
24. <xsd:element name="pesoMaximo" type="xsd:float"/>
25. </xsd:sequence>
26. </xsd:complexType>
27. </xsd:schema>
    
```

La otra representación se corresponde con un país anglosajón. La entidad se define de la siguiente forma:

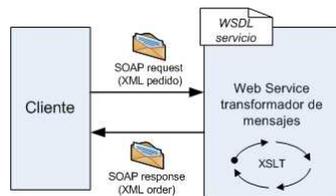
- Order:
 - Supplier:
 - First name - tipo cadena
 - Last name - tipo cadena
 - Phone number - tipo cadena
 - Building material
 - Identificador - tipo cadena
 - Maximum weight - tipo decimal, unidades de peso: libras

Esta estructura la definimos en order:Schema.xsd:

```

view plain print ?
01. <?xml version="1.0" encoding="UTF-8"?>
02. <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
03. targetNamespace="http://tutorial.adictosaltrabajo.com/schema/order
04. xmlns:tns="http://tutorial.adictosaltrabajo.com/schema/order
05. elementFormDefault="qualified">
06. <xsd:complexType name="supplier">
07. <xsd:sequence>
08. <xsd:element name="firstName" type="xsd:string"/></xsd:element>
09. <xsd:element name="lastName" type="xsd:string"/></xsd:element>
10. <xsd:element name="phoneNo" type="xsd:string"/></xsd:element>
11. </xsd:sequence>
12. </xsd:complexType>
13. <xsd:complexType name="buildingMaterial">
14. <xsd:sequence>
15. <xsd:element name="ID" type="xsd:int"/></xsd:element>
16. <xsd:element name="maximumWeight" type="xsd:float"/></xsd:element>
17. </xsd:sequence>
18. </xsd:complexType>
19. <xsd:element name="order">
20. <xsd:complexType>
21. <xsd:sequence>
22. <xsd:element name="supplier" type="tns:supplier"/></xsd:element>
23. <xsd:element name="buildingMaterial" type="tns:buildingMaterial"/></xsd:element>
24. </xsd:sequence>
25. </xsd:complexType>
26. </xsd:element>
27. </xsd:schema>
    
```

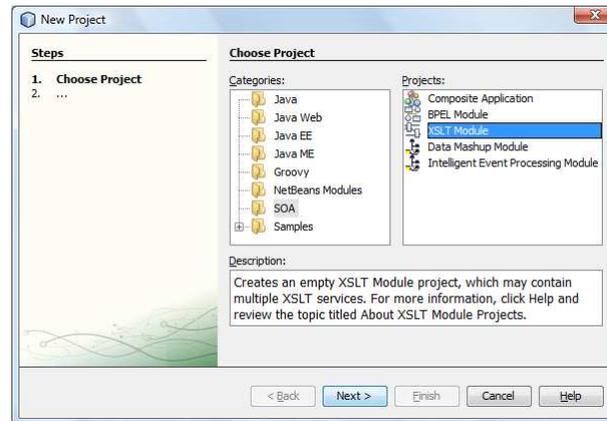
Nuestro servicio realizará la transformación necesaria para representar la información del pedido en la estructura anglosajona partiendo de la española. Un esquema representativo de la operación puede ser:



Implementación con XSLT Module de OpenESB

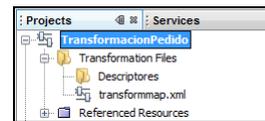
Creamos en NetBeans un nuevo proyecto de la categoría SOA y de tipo XSLT Module:

Introducción a Tapestry 5
2010-01-14 JMeter. Gestión de usuarios
2010-01-14 Patrón Visitor con commons-collections y sus Closures
2010-01-12 Creación de servicios web RestFul, con soporte a persistencia, en NetBeans.
2010-01-11 JMeter y JSF. Extracción del parámetro ViewState
2010-01-07 Importar el correo de Microsoft Outlook al cliente de correo de Mac OS.
2010-01-07 Monitor de Hudson para Eclipse.
2010-01-07 Patrones de diseño de XML Schema
2010-01-04 Procesador Inteligente de Eventos (IEP) con OpenESB
2010-01-04 PHP Vs Java
2009-12-29 Tutorial de BPEL con OpenESB (II)
2009-12-29 Tutorial de BPEL con OpenESB (I)
2009-12-28 Pruebas funcionales de servicios web con soapUI
2009-12-28 SoapUI: jugando con web services
2009-12-17 ¿Qué son el cloud computing y google app engine?
2009-12-14 JavaBean Datasource Ireport
2009-12-11 Contract-First web services con Visual Studio 2008
2009-12-09 Integrando Sonar con Hudson
2009-12-09 Apache + Tomcat: Balanceo de carga y alta disponibilidad
2009-12-08 MySQL: Replicación de bases de datos en MySQL
2009-12-07 Analizando la calidad del código Java con Sonar
2009-12-03 Instalar OpenESB 2.1 e Introducción



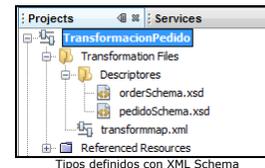
Proyecto de tipo XSLT Module en OpenESB

ponemos como nombre de proyecto `TransformacionPedido` y Target Namespace: `http://tutorial.adictosaltrabajo.com/transformmap/TransformacionPedido` y creamos a continuación una subcarpeta de `Transformation Files` que llamaremos `Descriptores`:



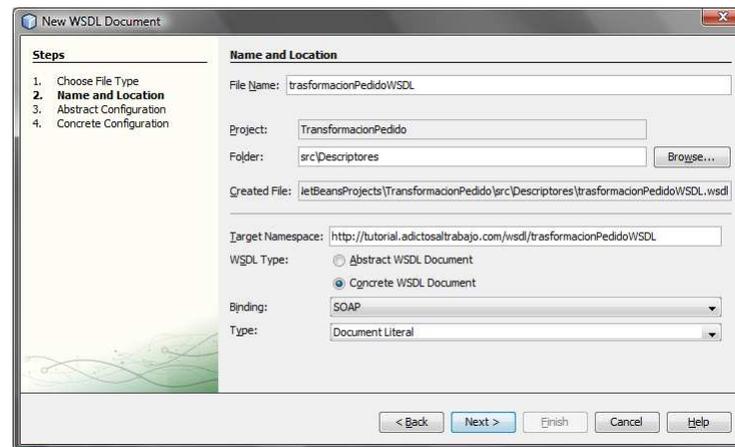
Proyecto inicial de XSLT Module en NetBeans

En la carpeta `descriptores` creamos los ficheros `pedidoSchema.xsd` y `orderSchema.xsd` con el contenido mostrado en el apartado anterior:



Tipos definidos con XML Schema

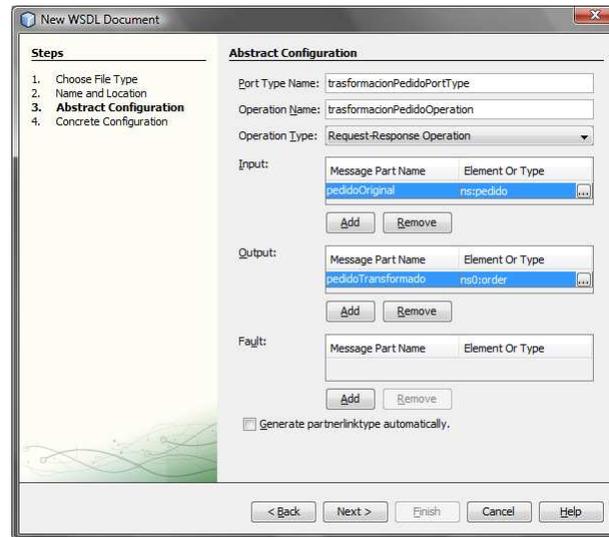
De esta manera ya tenemos la definición de los tipos de entrada y salida del servicio de transformación de mensajes, de manera que la conversión podrá describirse en un fichero XSLT. Este módulo de transformación de mensajes lo desplegaremos en el ESB, y será accesible de una manera estandarizada al igual que cualquier otro servicio; esto es, deberá disponer de un descriptor WSDL que contenga los detalles del servicio para poder vincularnos al mismo. Así pues, nuestro siguiente paso es crear en NetBeans el descriptor mediante el asistente del WSDL Document, en la misma carpeta de `Descriptores`, con las siguientes propiedades:



Crear un descriptor para el servicio de transformación

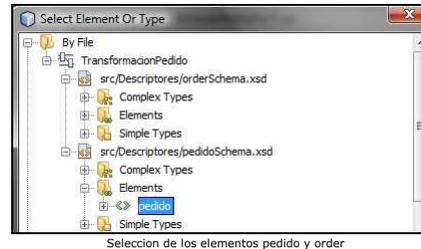
Definimos el contenido del mensaje y los tipos:

2009-11-25	Tutorial de Google Forms
2009-11-23	Alfresco - Modificando y eliminando contenido desde nuestras aplicaciones Java
2009-11-23	Alfresco - Añadiendo contenido desde nuestras aplicaciones Java
2009-11-23	Haciendo funcionar Google Chromium Operating System sobre Windows XP
2009-11-23	Redimensionar el tamaño de una partición de VirtualBox
2009-11-20	El Arte de las Presentaciones. Siguiendo la Senda Zen.
2009-11-18	Tutorial basico de google wave bots
2009-11-13	Introducción a Escritorios Animado (Winamp y MilkWave)
2009-11-12	Maven Assembly Plugin: empaquetando aplicaciones con Maven para la ejecución de procesos batch.
2009-11-12	Tutorial básico sobre Google Application Engine
2009-11-11	Instalación de Glassfish 2.1
2009-11-26	Rational Software Architect y Rational RequisitePro
2009-10-27	Cómo conseguir que Subversion avise a Hudson para lanzar una build
2009-10-26	Cómo instalar Hudson en Apache Tomcat
2009-10-26	Estilos externos en iReport
2009-10-14	Echo en MySQL
2009-09-28	iReport : Solución al problema de los parámetros dinámicos.
2009-09-27	Mavenizar Liferay SDK
2009-09-27	Crear un plugin con Liferay SDK
2009-09-23	EJB 3.0 y pruebas unitarias con Maven, JUnit 4 y Apache Open EJB.
2009-09-21	Primeros pasos con JBoss Seam



Configuración abstracta del descriptor

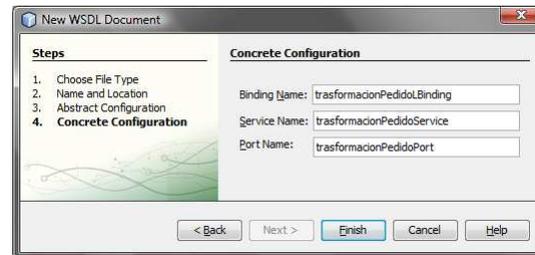
Para la elección de los tipos `pedido` y `order` de la imagen anterior, debemos escogerlos bajo el nodo `Elements` del explorador de tipos:



Selección de los elementos `pedido` y `order`

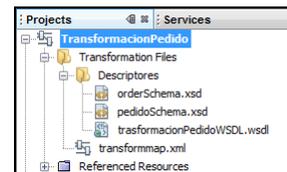
En caso de no hacerlo y seleccionarlos bajo el nodo de `Complex Types`, no estaremos respetando la regla [R2204](#) del WS-I (ya que hemos definido el tipo del descriptor como `Document-Literal`) y el asistente no nos permitirá continuar (más información en el tutorial [Patrones de diseño de XML Schema](#)).

Por último definimos la sección concreta del WSDL:



Configuración concreta del WSDL

Al finalizar, dispondremos en el proyecto de todos los descriptores necesarios:



Últimas ofertas de empleo

2009-07-31
[T. Información - Operador \(día / noche\) - BARCELONA.](#)

2009-06-25
[Atención a cliente - Call Center - BARCELONA.](#)

2009-06-19
[Otras - Ingeniería \(minas, puentes y puertos\) - VALENCIA.](#)

2009-06-17
[Comercial - Ventas - ALICANTE.](#)

2009-06-03
[Comercial - Ventas - VIZCAYA.](#)

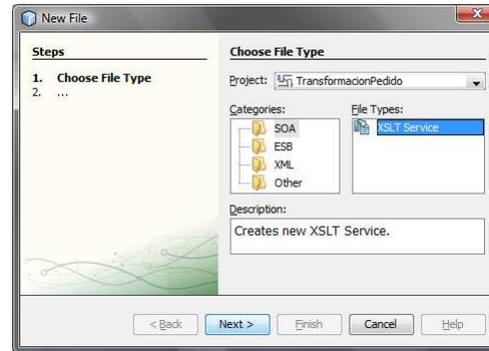
Ads by Google

Proyecto con los descriptores necesarios

Proceso de transformación con XSLT

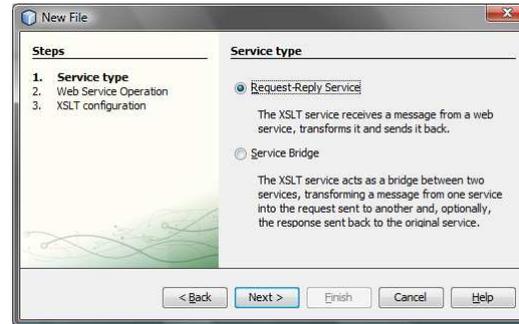
En este punto ya podemos implementar la lógica de transformación. Puesto que vamos a transformar mensajes XML, usaremos XSLT.

Pulsamos con el botón derecho sobre el nombre de proyecto TransformacionPedido | New | Other | XSLT Service:



Creamos un servicio de transformación con XSLT

El tipo de servicio de transformación será petición-respuesta (la otra opción corresponde a unmessage broker de tipo proxy):



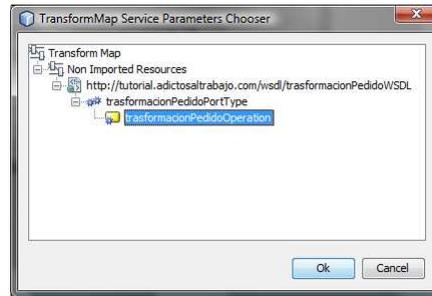
Servicio XSLT petición-respuesta

A continuación definimos cómo se aplica la transformación. Introducimos como nombre del web service que publicaremos transformacionPedidoService y la operación la definida en el descriptoe del proyecto, transformacionPedidoOperation:



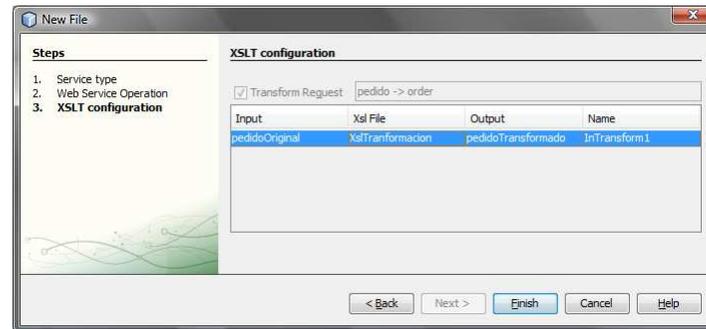
Nombre y operación del web service público

Como podemos observar, la operación corresponde a la que hemos definido en WSDL del proyecto:



Operación de transformación definida en el WSDL

En el siguiente paso de configuración XSLT cambiamos el nombre del fichero XSLT por `XslTransformacion`:



Configuración XSLT

En el proyecto se ha creado el fichero `XslTransformacion.xsl`, cuyo código de conversión para nuestro propósito debe ser el siguiente:

```

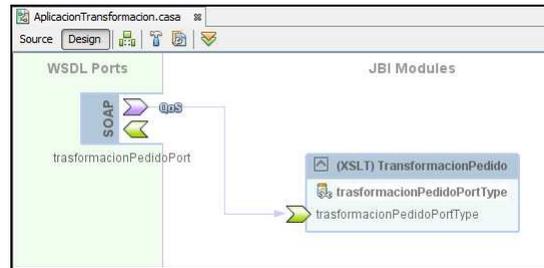
view plain print ?
01. <?xml version="1.0" encoding="UTF-8"?>
02. <xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform version="1.0"
03. xmlns:nspedido="http://tutorial.adictosaltrabajo.com/schema/pedido"
04. xmlns:nsorder="http://tutorial.adictosaltrabajo.com/schema/order"
05. <xsl:template match="/">
06.   <nsorder:order>
07.     <nsorder:supplier>
08.       <nsorder:firstName>
09.         <xsl:value-of select="/nspedido:pedido/nspedido:contacto/nspedido:nombre"/>
10.       </nsorder:firstName>
11.       <nsorder:lastName>
12.         <xsl:value-of select="substring-before(/nspedido:pedido/nspedido:contacto/nspedido:apellidos, ' ')/>
13.       </nsorder:lastName>
14.       <nsorder:phoneNo>
15.         <xsl:value-of select="concat('+34 ', /nspedido:pedido/nspedido:contacto/nspedido:telefono)/>
16.       </nsorder:phoneNo>
17.     </nsorder:supplier>
18.     <nsorder:buildingMaterial>
19.       <nsorder:ID>
20.         <xsl:value-of select="/nspedido:pedido/nspedido:material/nspedido:ID/>
21.       </nsorder:ID>
22.       <nsorder:maximumWeight>
23.         <xsl:value-of select="/nspedido:pedido/nspedido:material/nspedido:pesoMaximo * 2.20462262/>
24.       </nsorder:maximumWeight>
25.     </nsorder:buildingMaterial>
26.   </nsorder:order>
27. </xsl:template>
28. </xsl:stylesheet>

```

Con esto ya hemos terminado el proyecto, vamos a probarlo.

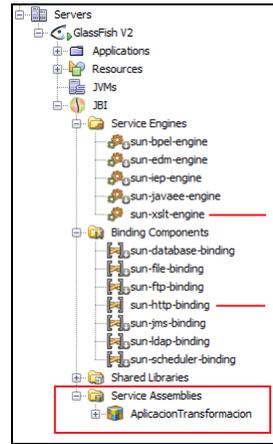
Desplegar y probar el proyecto en OpenESB

Tal como se explica en otros tutoriales (e.g. [Tutorial de BPEL con OpenESB \(II\)](#)), para probar un módulo SOA de OpenESB es necesario que pertenezca a una aplicación compuesta de JBI. Para ello creamos en NetBeans un `projectoComposite Application` desde el menú `File | New Project | SOA | Composite Application`, y le damos el nombre `AplicacionTransformacion`. Sobre la carpeta `JBI Modules` pulsamos botón derecho | `Add JBI Module` y en la ventana emergente seleccionamos el `projectoTransformacionPedido` y pulsamos en `Add Project JAR Files`. Finalmente compilamos el proyecto, pulsando con botón derecho sobre su nombre `AplicacionTransformacion` | `Clean and Build`. El aspecto final será:



Composite Application de SOA para desplegar el XSL Module

Teniendo el servidor GlassFish V2 previamente arrancado, desplegamos el proyecto AplicacionTransformacion pulsando de nuevo botón derecho | Deploy. El proyecto se habrá desplegado en el contenedor JBI:



Estructura de módulos JBI

En este momento la operación del proxy de transformación estará expuesto como una operación habitual de webservice, y podrá ser invocados como tales. Su WSDL se encuentra publicado en <http://localhost:9080/transformacionPedidoService/transformacionPedidoPort?WSDL>

```
<?xml version="1.0" encoding="UTF-8" ?>
<definitions name="transformacionPedidoWSDL"
targetNamespace="http://tutorial.adictosaltrabajo.com/wsdl/transformacionPedidoWSDL"
xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:tns="http://tutorial.adictosaltrabajo.com/schema/pedido"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:tns2="http://tutorial.adictosaltrabajo.com/wsdl/transformacionPedidoWSDL"
xmlns:nsd="http://tutorial.adictosaltrabajo.com/schema/order"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" />
<types>
<xs:schema targetNamespace="http://tutorial.adictosaltrabajo.com/wsdl/transformacionPedidoWSDL">
<xs:import namespace="http://tutorial.adictosaltrabajo.com/schema/order"
schemaLocation="http://Vista:9080/AplicacionTransformacion-sun-http-binding/TransformacionPedido/Descriptores/orderSchema.xsd" />
<xs:import namespace="http://tutorial.adictosaltrabajo.com/schema/pedido"
schemaLocation="http://Vista:9080/AplicacionTransformacion-sun-http-binding/TransformacionPedido/Descriptores/pedidoSchema.xsd" />
</xs:schema>
</types>
<message name="transformacionPedidoOperationRequest">
<part name="pedidoOriginal" element="ns:pedido" />
</message>
<message name="transformacionPedidoOperationResponse">
<part name="pedidoTransformado" element="ns:order" />
</message>
<portType name="transformacionPedidoPortType">
<operation name="transformacionPedidoOperation">
<input name="input1" message="tns:transformacionPedidoOperationRequest" />
<output name="output1" message="tns:transformacionPedidoOperationResponse" />
</operation>
</portType>
<binding name="transformacionPedidoDotBinding" type="tns:transformacionPedidoPortType">
<soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http" />
<operation name="transformacionPedidoOperation">
```

Descriptor del servicio de transformación desplegado en el ESB

Vamos a invocar al servicio web utilizando, por ejemplo, soapUI (tutorial de uso en [SoapUI: Jugando con web services](#)). El SOAP request será:

```

<?xml version='1.0' encoding='UTF-8'>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <ped:pedido>
      <ped:contacto>
        <ped:nombre>Raul</ped:nombre>
        <ped:apellidos>Fernandez Perez</ped:apellidos>
        <ped:telefono>674839049</ped:telefono>
      </ped:contacto>
      <ped:material>
        <ped:ID>18892883</ped:ID>
        <ped:pesoMaximo>20.5</ped:pesoMaximo>
      </ped:material>
    </ped:pedido>
  </soapenv:Body>
</soapenv:Envelope>
    
```

Mensaje SOAP request

El mensaje de respuesta obtenido corresponde con el esperado:

```

<?xml version='1.0' encoding='UTF-8'>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <nsorder:order xmlns:msgns="http://tutorial.adictosaltrabajo.com/msgns">
      <nsorder:supplier>
        <nsorder:firstName>Raul</nsorder:firstName>
        <nsorder:lastName>Fernandez</nsorder:lastName>
        <nsorder:phoneNo>+34 674839049</nsorder:phoneNo>
      </nsorder:supplier>
      <nsorder:buildingMaterial>
        <nsorder:ID>18892883</nsorder:ID>
        <nsorder:maximumWeight>45.19476371</nsorder:maximumWeight>
      </nsorder:buildingMaterial>
    </nsorder:order>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
    
```

Mensaje SOAP response

El mensaje ha sido transformado en tiempo de ejecución correctamente.

Conclusión

La heterogeneidad habitual en los sistemas que componen la infraestructura de una empresa que implementa SOA requieren soluciones que permitan salvar las diferencias entre la representación de la información. La ejecución de procesos en los que intervienen aplicaciones legacy requieren habitualmente una adaptación de los mensajes, al menos hasta un estado más avanzado de integración. Pero lo realmente importante es que los usuarios de negocio tienen que poder trabajar con una visión única de las entidades que manejan, y progresivamente los sistemas tienen que adecuarse para tener una sola representación, mediante normalización sintáctica o uso de estándares.

Para todo ello, la industria ofrece una amplia variedad de soluciones tecnológicas, y en este tutorial hemos probado la que OpenESB nos ofrece.

¿Qué te ha parecido el tutorial? Déjanos saber tu opinión y vota!

Muy malo Malo Regular Bueno Muy bueno

(Sólo para usuarios registrados)

[» Regístrate y accede a esta y otras ventajas «](#)

Anímate y coméntanos lo que pienses sobre este tutorial

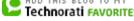
Puedes opinar o comentar cualquier sugerencia que quieras comunicarnos sobre este tutorial; con tu ayuda, podemos ofrecerte un mejor servicio.

(Sólo para usuarios registrados)

[» Regístrate y accede a esta y otras ventajas «](#)

Autor

Mensaje de usuario registrado

- Puedes inscribirte en nuestro servicio de notificaciones [haciendo clic aquí](#).
- Puedes firmar en nuestro libro de visitas [haciendo clic aquí](#).
- Puedes asociarte al grupo AdictosAlTrabajo en XING [haciendo clic aquí](#).
- **Añadir a favoritos Technorati.** 



Esta obra está licenciada bajo [licencia Creative Commons de Reconocimiento-No comercial-Sin obras derivadas 2.5](#)

Recuerda

Autentia te regala la mayoría del conocimiento aquí compartido ([Ver todos los tutoriales](#)). Somos expertos en: J2EE, Struts, JSF, C++, OOP, UML, UP, Patrones de diseño ... y muchas otras cosas.

¿Nos vas a tener en cuenta cuando necesites consultoría o formación en tu empresa?, ¿Vas a ser tan generoso con nosotros como lo tratamos de ser con vosotros?

Somos pocos, somos buenos, estamos motivados y nos gusta lo que hacemos ...

Autentia = Soporte a Desarrollo & Formación.

info@autentia.com



Tutoriales recomendados

Nombre	Resumen	Fecha	Visitas	Valoración	Votos	Pdf
Transformación de mensajes en SOA con OpenESB	En este tutorial vamos a mostrar con un ejemplo práctico una de las capacidades esenciales de todo sistema de integración, y por tanto característica del Enterprise Service Bus de SOA: la transformación de mensajes.	2010-01-29	5	-	-	
Creación de servicios web RestFul, con soporte a persistencia, en NetBeans.	En este tutorial vamos a analizar el soporte que nos proporciona NetBeans para la creación de servicios web RESTful, que den soporte a las operaciones típicas que se pueden realizar contra una tabla de base de datos. El objetivo es la creación de un proyec	2010-01-12	515	Muy bueno	1	
Patrones de diseño de XML Schema	En este tutorial vamos a analizar y mostrar con ejemplos los cuatro principales patrones de diseño de XML Schema: Russian Doll, Salami Slice, Venetian Blind y Garden of Eden.	2010-01-07	583	Muy bueno	1	
Tutorial de BPEL con OpenESB (I)	En este tutorial vamos a aprender a crear procesos BPEL practicando con un ejemplo: un proceso de negocio de venta online de libros.	2009-12-29	1063	-	-	
Tutorial de BPEL con OpenESB (II)	Continuación del Tutorial de BPEL con OpenESB (I).	2009-12-29	822	-	-	
SoapUI: jugando con web services	SoapUI es una aplicación muy versátil que nos permite probar, simular y generar código de servicios web de forma ágil. En este tutorial introducimos esta herramienta, mostrando su instalación y un primer ejemplo de uso con un web service público	2009-12-28	1151	Muy bueno	1	
Pruebas funcionales de servicios web con soapUI	Las pruebas unitarias en cualquier paradigma de programación son, más que una buena práctica, una garantía para obtener un software robusto y (más) fácilmente mantenible. Como responsables de diseño o desarrollo de web services hemos de aplicar estas buen	2009-12-28	809	-	-	
Contract-First web services con Visual Studio 2008	Utilizaremos Visual Studio 2008 con el addin WSCF.blue para generar contract-first web services (servicios web dirigidos por contrato)	2009-12-11	829	-	-	
Instalar OpenESB 2.1 e Introducción	En este tutorial veremos como descargar e instalar OpenESB y explicaremos sus funcionalidades	2009-12-03	1450	-	-	
Tutorial de Google Forms	En este breve tutorial vamos a ver cómo podemos usar la herramienta Google Forms, para crear rápida y cómodamente un formulario de recogida de datos online (y de paso opinar sobre el Real Madrid).	2009-11-25	1569	-	-	

Nota:

Los tutoriales mostrados en este Web tienen como objetivo la difusión del conocimiento. Los contenidos y comentarios de los tutoriales son responsabilidad de sus respectivos autores. En algún caso se puede hacer referencia a marcas o nombres cuya propiedad y derechos es de sus respectivos dueños. Si algún afectado desea que incorporemos alguna reseña específica, no tiene más que solicitarlo. Si alguien encuentra algún problema con la información publicada en este Web, rogamos que informe al administrador rcanales@adictosaltrabajo.com para su resolución.