

¿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)

Estas en: [Inicio](#) [Tutoriales](#) Axis2: Invocación de Servicios Web usando distintos MEP

Últimas Noticias

- » Historia de la informática. Capítulo 64. 1986
- » Autentia en la Sun Open Communities Forum
- » Comentario del libro: El economista naturalista de Robert Frank
- » Historia de la informática. Capítulo 63. 1985
- » Contratos ágiles: Vendiendo Scrum a tus clientes.
- » Resumen de la cuarta charla gratuita de Autentia: SCRUM (con video)
- » Si se pregunta ¿Qué ofrece este Web?
- » [Vota AdictosAlTrabajo.com en DZone](#)

+ Noticias Destacadas

- » [Contratos ágiles: Vendiendo Scrum a tus clientes.](#)
- » [Quinta charla Autentia + Proyectalis + Agile Spain: Contratos ágiles: Vendiendo Scrum a tus clientes](#)
- » [Lo mejor de esta semana: Curso de Scrum con Ángel Medinilla](#)
- » [Resumen de la cuarta charla gratuita de Autentia: SCRUM \(con video\)](#)

+ Comentarios Cómic

+ Enlaces

Catálogo de servicios Autentia (PDF 6,2MB)



En formato comic...

Tutorial desarrollado por



Carlos García Pérez

Creador de MobileTest, un complemento educativo para los profesores y sus alumnos.

Consultor tecnológico en el desarrollo de proyectos informáticos.

Técnico especialista en informática de empresa (CEU)

Ingeniero Técnico en Informática * (Politécnica)

Puedes encontrarme en [Autentia](#)

Somos expertos en Java/J2EE

Catálogo de servicios de Autentia

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

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

AdictosAlTrabajo.com es el Web de difusión de conocimiento de [Autentia](#).



[Catálogo de cursos](#)

Descargar este documento en formato PDF: [axis2_mep.pdf](#)

Fecha de creación del tutorial: 2009-07-02

Axis2: Invocación de Servicios Web usando distintos MEP

Introducción.

En este tutorial vamos a llevar a la práctica la **invocación de servicios web** a través de cada uno de los patrones de intercambios de mensajes (**MEP**) definidos en [WSDL 2.0](#). Para ello, vamos a realizar una aplicación de escritorio en donde utilizaremos el API de cliente de Axis2 para invocar la única operación que expone el servicio Web construido en un tutorial anterior » [Spring WS: Creación de Servicios Web con Spring](#) con cada uno de los MEP.

Manos a la obra con la aplicación cliente:

El siguiente ejemplo está construido bajo el siguiente entorno:

- o HP Pavilion.
- o Windows Vista Home Premium.
- o Eclipse Ganymede.
- o Java 6.
- o Tomcat 6
- o Maven 2.
- o Plugin Maven 4QE para Eclipse.
- o Axis2 1.4.1

Si después de leer este tutorial quieres probarlo tu mismo, en los siguientes enlaces te dejo lo necesario:

- El [servicio web](#) en formato WAR, por lo que simplemente deberás desplegarlo en tu servidor o contenedor favorito.
- La [parte cliente](#) (proyecto Maven2 y Eclipse) que vemos en este tutorial.

Para poder utilizar los servicios que ofrece un web service necesitamos conocer su descripción o WSDL. Esta casi siempre es publicada por el servicio web a través de una URL, por ejemplo: <http://localhost:8080/bibliotecaWS/bibliotecaWS.wsdl>)

WSDL del servicio web a invocar con distintos MEP:

Últimos tutoriales

2009-07-02
[Axis2: Invocación de Servicios Web usando distintos MEP](#)

2009-07-02
[Instalación OpenOffice](#)

2009-07-02
[Juegos 3D en Java: Blender y JMonkeyEngine](#)

2009-06-20
[StAX \(Xml Pull Parser\): Streaming API para XML](#)

2009-06-15
[Configuración de la desconexión de usuarios con ICEFaces](#)

2009-06-10
[LWUIT: Una librería gráfica tipo AWT o Swing para J2ME](#)

2009-06-10
[Mapas mentales con XMind](#)

2009-02-26
[Redimensionar Imagenes en Windows Vista](#)

2009-06-08
[UploadFile con Icefaces + Hibernate + Anotaciones](#)

2009-06-05
[Habilitar exportación en Liferay](#)

2009-06-01
[Registrar Liferay en Eclipse](#)

2009-05-29
[Liferay Social Office](#)



Web

www.adictosaltrabajo.com

Buscar

view plain print ?

```
01. <?xml version="1.0" encoding="utf-8"?>
02. <wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
03.   xmlns:sch="http://www.adictosaltrabajo.com/spring/ws/schemas"
04.   xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
05.   xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
06.   xmlns:tns="http://www.adictosaltrabajo.com/spring/ws/schemas"
07.   targetNamespace="http://www.adictosaltrabajo.com/spring/ws/schemas">
08.   <wsdl:types>
09.     <xsd:schema xmlns:schemas="http://www.adictosaltrabajo.com/spring/ws/schemas"
10.       xmlns:xsd="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
11.       targetNamespace="http://www.adictosaltrabajo.com/spring/ws/schemas">
12.       <xsd:element name="BooksInfoRequest">
13.         <xsd:complexType>
14.           <xsd:sequence>
15.             <xsd:element name="categoria" type="xsd:string" />
16.             <xsd:element name="nivel" type="schemas:nivelType" />
17.           </xsd:sequence>
18.         </xsd:complexType>
19.       </xsd:element>
20.       <xsd:simpleType name="nivelType">
21.         <xsd:restriction base="xsd:string">
22.           <xsd:enumeration value="basico" />
23.           <xsd:enumeration value="medio" />
24.           <xsd:enumeration value="avanzado" />
25.         </xsd:restriction>
26.       </xsd:simpleType>
27.       <xsd:element name="BooksInfoResponse">
28.         <xsd:complexType>
29.           <xsd:sequence>
30.             <xsd:element maxOccurs="unbounded" minOccurs="0" ref="schemas:libro" />
31.           </xsd:sequence>
32.         </xsd:complexType>
33.       </xsd:element>
34.       <xsd:element name="libro">
35.         <xsd:complexType>
36.           <xsd:sequence>
37.             <xsd:element name="editorial" type="xsd:string" />
38.             <xsd:element name="titulo" type="xsd:string" />
39.             <xsd:element name="paginas" type="xsd:positiveInteger" />
40.             <xsd:element name="precio" type="xsd:positiveInteger" />
41.           </xsd:sequence>
42.         </xsd:complexType>
43.       </xsd:element>
44.     </xsd:schema>
45.   </wsdl:types>
46.   <wsdl:message name="BooksInfoResponse">
47.     <wsdl:part element="tns:BooksInfoResponse" name="BooksInfoResponse"></wsdl:part>
48.   </wsdl:message>
49.   <wsdl:message name="BooksInfoRequest">
50.     <wsdl:part element="tns:BooksInfoRequest" name="BooksInfoRequest"></wsdl:part>
51.   </wsdl:message>
52.   <wsdl:portType name="bibliotecaWS">
53.     <wsdl:operation name="BooksInfo">
54.       <wsdl:input message="tns:BooksInfoRequest" name="BooksInfoRequest"></wsdl:input>
55.       <wsdl:output message="tns:BooksInfoResponse" name="BooksInfoResponse"></wsdl:output>
56.     </wsdl:operation>
57.   </wsdl:portType>
58.   <wsdl:binding name="bibliotecaWSSoap11" type="tns:bibliotecaWS">
59.     <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http" />
60.     <wsdl:operation name="BooksInfo">
61.       <soap:operation soapAction="" />
62.       <wsdl:input name="BooksInfoRequest">
63.         <soap:body use="literal" />
64.       </wsdl:input>
65.       <wsdl:output name="BooksInfoResponse">
66.         <soap:body use="literal" />
67.       </wsdl:output>
68.     </wsdl:operation>
69.   </wsdl:binding>
70.   <wsdl:binding name="bibliotecaWSSoap12" type="tns:bibliotecaWS">
71.     <soap12:binding style="document" transport="http://schemas.xmlsoap.org/soap/http" />
72.     <wsdl:operation name="BooksInfo">
73.       <soap12:operation soapAction="" />
74.       <wsdl:input name="BooksInfoRequest">
75.         <soap12:body use="literal" />
76.       </wsdl:input>
77.       <wsdl:output name="BooksInfoResponse">
78.         <soap12:body use="literal" />
79.       </wsdl:output>
80.     </wsdl:operation>
81.   </wsdl:binding>
82.   <wsdl:service name="bibliotecaWSService">
83.     <wsdl:port binding="tns:bibliotecaWSSoap11" name="bibliotecaWSSoap11">
84.       <soap:address location="http://localhost:8080/bibliotecaWS/services" />
85.     </wsdl:port>
86.     <wsdl:port binding="tns:bibliotecaWSSoap12" name="bibliotecaWSSoap12">
87.       <soap12:address location="http://localhost:8080/bibliotecaWS/services" />
88.     </wsdl:port>
89.   </wsdl:service>
90. </wsdl:definitions>
```

Si nos fijamos en el WSDL observamos que (de cara al código fuente que veremos a continuación):

1. El servicio Web tiene dos `wsdl:port` de comunicación (líneas 83 a 88)
2. Sólo hay definido un Servicio Web (líneas 82-89)
3. Sólo hay definida una operación `BooksInfo` (líneas 72 a 80)
4. El formato de la petición a la única operación es el siguiente (líneas 12-26):

2009-05-28
Broadcast con Ustream

2009-05-25
Tabla datos accesible con ordenación y paginación

2009-05-21
Primeros pasos con Audacity: Un editor de sonido libre y multiplataforma.

2009-05-11
Introducción a TortoiseSVN

2009-05-07
Hacer 'scp' de varios ficheros sin solicitud de clave

2009-05-02
Plugin Hibernate3 para Maven

2009-04-26
AgileDraw: una técnica rápida de modelado

2009-04-24
Spring AOP: Cacheando aplicaciones usando anotaciones y aspectos con Aspectj

2009-04-20
Modelos de conocimiento con CmapTools

2009-04-16
Informes Crosstab con iReport

2009-04-16
Registro de un fichero de datos personales con el formulario NOTA

2009-04-15
Estadísticas de www.adictosaltrabajo.com Abril 2009

2009-04-15
Iniciación a OSWorkflow con Spring

2009-04-14
Tests de Selenium con librerías de componentes JSF: Apache Tomahawk.

2009-04-13
JTAPI. El API de Telefonía para Java

2009-04-13
Registro de Web Services con Apache jUDDI. Configuración y ejemplo

2009-04-13
Cómo hacer UML con Eclipse y el plugin UML2

2009-04-09
Spring WS: Servicios Web a través del correo electrónico

```
<BooksInfoRequest xmlns="http://www.adictosaltrabajo.com/spring/ws/schemas">
  <categoria>Servicios Web</categoria>
  <nivel>avanzado</nivel>
</BooksInfoRequest>
```

Axis2MEPApp: Ejemplo del uso el API de cliente de Axis2 para conseguir invocar un método con distintos patrones de intercambio de mensajes (MEP).

A continuación contruiremos una aplicación de escritorio en donde al ejecutarla podremos elegir el método de invocación al servicio Web: In-Out sincrónico, In-Out asíncrónico, etc.
El código fuente está autocomentado.

2009-04-02
[Creación de cursos con Moodle](#)

2009-03-31
[Integrar Liferay Portal 5.2.1 con Pentaho BI 2.0.0 sobre MySQL 5.1](#)

2009-03-31
[Spring WS: Construcción de Clientes de Servicios Web con Spring](#)

2009-03-30
[Administración de sitios Moodle](#)

2009-03-29
[Empaquetamiento de aplicaciones de escritorio \(standalone\) con Maven](#)

2009-03-27
[Primeros pasos con Moodle](#)

2009-03-26
[Introducción a JSF Java](#)

2009-03-25
[A1 Website Analyzer](#)

2009-03-24
[Cómo ver el correo de Gmail sin conexión a Internet](#)

2009-03-20
[JasperReports Maven Plugin](#)

2009-03-16
[Creación de contenidos SCORM: eXe](#)

2009-03-15
[Spring WS: Creación de Servicios Web con Spring](#)

2009-03-13
[Instalación Alfresco \(Labs\)](#)

2009-02-26
[Maven JXR Plugin: publica el código fuente en el site](#)

2009-03-15
[Generación de XML Schema \(XSD\) y DTD a partir de documentos XML](#)

2009-03-04
[Persistencia con Spring](#)

2009-02-26
[Vistas materializadas](#)

2009-02-03
[Instalación de MySQL 5.1 en Windows](#)

2009-03-03
[Instalación de Java Virtual Machine](#)

```

view plain print ?
01. package com.autentia.tutoriales.axis2.cliente;
02.
03. import java.net.URL;
04. import javax.xml.namespace.QName;
05. import org.apache.axiom.om.*;
06. import org.apache.axis2.client.ServiceClient;
07. import org.apache.axis2.client.async.AxisCallback;
08. import org.apache.axis2.context.ConfigurationContext;
09.
10. /**
11.  * Ejemplo del uso el API de cliente de Axis2 para conseguir invocar un método
12.  * con distintos patrones de intercambio de mensajes (MEP).
13.  * @author Carlos García. Autentia
14.  */
15. public class Axis2MEPApp {
16.
17.     public static void main(String[] args) {
18.         char userOption = '0';
19.
20.         try {
21.             while ((userOption < '1') || (userOption > '4')){
22.                 System.out.println("Ejemplo de invocación a servicios web con distintos Message E
23.                 System.out.println("(1) MEP In-Out Sincrono");
24.                 System.out.println("(2) MEP In-Out Asíncrono");
25.                 System.out.println("(3) MEP In-Only. Recepción y procesamiento del mensaje asegur
26.                 System.out.println("(4) MEP In-Only. Recepción y procesamiento del mensaje no ase
27.                 System.out.println("Seleccione una opción (1, 2, 3, 4):");
28.                 userOption = (char) System.in.read();
29.             }
30.
31.             // Usamos la configuración por defecto de Axis2
32.             ConfigurationContext context = null;
33.
34.             // URL del WSDL
35.             URL wsdlURL = new URL("http://localhost:8080/bibliotecaWS/bibliotecaWS.wsdl");
36.
37.             // Podría ser Null pues sólo hay un servicio en el WSDL
38.             QName serviceName = new QName("http://www.adictosaltrabajo.com/spring/ws/schemas", "b
39.
40.             // El WSDL tiene 2 Port elegimos uno.
41.             String portName = "bibliotecaWSSoap12";
42.
43.             // Clase cliente básica (No tiene acceso a operaciones más avanzadas co
44.             ServiceClient wsClient = new ServiceClient(context, wsdlURL, serviceName, portName);
45.
46.             // Operación a invocar
47.             QName operation = new QName("http://www.adictosaltrabajo.com/spring/ws/schemas", "Boo
48.
49.             // Obtenemos la petición a enviar al servicio web
50.             OMElement request = Axis2MEPApp.getRequest("Servicios Web", "avanzado");
51.
52.
53.             switch (userOption){
54.                 case '1':
55.                     OMElement response = wsClient.sendReceive(operation, request);
56.
57.                     // Imprimimos la respuesta
58.                     System.out.println(response);
59.                     break;
60.                 case '2':
61.                     // CallBack que recibirá la respuesta
62.                     AxisCallback callback = new MyAxisCallback();
63.
64.                     wsClient.sendReceiveNonBlocking(operation, request, callback);
65.
66.                     for (int i = 0; i < 5; i++){
67.                         System.out.println("Continuamos trabajando en otras cosas mientras el ser
68.                         "(tiempos de red + proceso + configuraciones poco opti
69.                         Thread.sleep(100);
70.                     }
71.                     break;
72.                 case '3':
73.                     wsClient.sendRobust(operation, request);
74.                     break;
75.                 case '4':
76.                     wsClient.fireAndForget(operation, request);
77.
78.                     // Esperamos un poco para que le de tiempo a enviar la petición
79.                     Thread.sleep(200);
80.                     break;
81.             }
82.         } catch (Exception ex){
83.             System.out.println(ex); // Es sólo un tutorial, no me preocupo de gestionar
84.         }
85.
86.
87.         System.out.println("FIN DE LA APLICACION");
88.     }
89.
90.
91.     /**
92.     * @param categoria Categoría del libro a consultar
93.     * @param nivel Nivel del libro a consultar, valores permitidos: basico,
94.     * @return Devuelve la petición a enviar al servicio web.
95.     */
96.     private static OMElement getRequest(String categoria, String nivel){
97.         /**

```

2009-03-03
Primeros Pasos con Liferay
5.2.1

Últimas ofertas de empleo

2009-06-29
Atención a cliente - Call Center - BARCELONA.

2009-06-25
Atención a cliente - Call Center - BARCELONA.

2009-06-20
Comercial - Ventas - CASTELLON.

2009-06-19
Otras - Ingenieria (minas, puentes y puertos) - VALENCIA.

2009-06-17
Comercial - Ventas - ALICANTE.

MyAxisCallback: Implementación de un AxisCallback para las operaciones asíncronas.

```
view plain print ?
01. package com.autentia.tutoriales.axis2.cliente;
02.
03. import org.apache.axiom.om.OMElement;
04. import org.apache.axis2.context.MessageContext;
05. import org.apache.axis2.client.async.AxisCallback;
06.
07. /**
08.  * Implementación de un AxisCallback para las operaciones asíncronas.
09.  * @author Carlos García. Autentia.
10.  */
11. public class MyAxisCallback implements AxisCallback {
12.
13.     /**
14.      * @see org.apache.axis2.client.async.AxisCallback#onComplete()
15.      */
16.     public void onComplete() {
17.         System.out.println("onComplete");
18.     }
19.
20.     /**
21.      * @see org.apache.axis2.client.async.AxisCallback#onFault(org.apache.axis2.c
22.      */
23.     public void onFault(MessageContext context) {
24.         System.out.println("onFault");
25.
26.         OMElement payload = context.getEnvelope().getBody();
27.         System.out.println(payload);
28.     }
29.
30.     /**
31.      * @see org.apache.axis2.client.async.AxisCallback#onError(java.lang.Exceptio
32.      */
33.     public void onError(Exception ex) {
34.         System.out.println("onError " + ex.toString());
35.     }
36.
37.     /**
38.      * @see org.apache.axis2.client.async.AxisCallback#onMessage(org.apache.axis2
39.      */
40.     public void onMessage(MessageContext context) {
41.         System.out.println("onMessage");
42.         OMElement payload = context.getEnvelope().getBody();
43.         System.out.println(payload);
44.     }
45. }
```

Archivo de configuración de Maven 2: pom.xml:

Por si te sirve de plantilla para otros proyectos, te dejo el pom.xml del proyecto.

```
view plain print ?
01. <?xml version="1.0" encoding="UTF-8"?>
02. <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-in
03.     xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xs
04.     <modelVersion>4.0.0</modelVersion>
05.     <groupId>com.autentia.tutoriales</groupId>
06.     <artifactId>bibliotecaWSClienteAxis2</artifactId>
07.     <packaging>jar</packaging>
08.     <version>1.0-SNAPSHOT</version>
09.     <name>bibliotecaWSClienteAxis2</name>
10.     <url>http://www.adictosaltrabajo.com</url>
11.     <!-- Sintáxis Java 6 -->
12.     <build>
13.         <plugins>
14.             <plugin>
15.                 <artifactId>maven-compiler-plugin</artifactId>
16.                 <configuration>
17.                     <source>1.6</source>
18.                     <target>1.6</target>
19.                     <encoding>UTF-8</encoding>
20.                 </configuration>
21.             </plugin>
22.         </plugins>
23.     </build>
24.     <dependencies>
25.         <!-- Todas las dependencias de Axis2 en una -->
26.         <dependency>
27.             <groupId>org.apache.axis2</groupId>
28.             <artifactId>axis2-kernel</artifactId>
29.             <version>1.4.1</version>
30.         </dependency>
31.     </dependencies>
32. </project>
```

Referencias

- [Reference Guide to Apache Axis2 Client API Parameters.](#)
- [Web Services Description Language \(WSDL\) Version 2.0 Part 1: Core Language.](#)
- [Web Services Description Language \(WSDL\) Version 2.0 Part 2: Adjuncts](#)
- [Writing Your Own services.xml for Axis2 Web Services.](#)

Conclusiones

Como veis el API de cliente Axis2 es bastante facil de usar y comprender, siendo además bastante ligero en cuanto número de clases e interfaces.

Aunque con la clase `ServiceClient` podemos satisfacer la mayoría de nuestras necesidades de cliente, tenemos disponible la clase `OperationClient` para operaciones más avanzadas, como por ejemplo manipulación de cabeceras.

Existen otros patrones MEP en los que es el servidor el que inicia la comunicación Out-Only, etc. ver [WSDL 2.0](#), pero estos se consiguen de la misma forma que la que ya hemos visto, siempre y cuando el servidor conozca las direcciones de los clientes. Si estais interesados podeis mirar los ejemplos de la distribución para profundizar más....

Carlos García Pérez. Creador de [MobileTest](#), un complemento educativo para los profesores y sus alumnos.
cgpcosmad@gmail.com

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

Muy malo Malo Regular Bueno Muy bueno



Votar

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.

Nombre:

E-Mail:

Comentario:

Enviar comentario

[Texto Legal y condiciones de uso](#)

- 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
Registro de Web Services con Apache jUDDI. Configuración y ejemplo	Veamos como podemos catalogar y buscar web services bajo la especificación UDDI (Universal Description, Discovery and Integration)	2009-04-13	1865	Muy bueno	13	
Spring WS: Servicios Web a través del correo electrónico	En este tutorial aprenderemos a configurar un cliente y servicio web para que envíe/atienda peticiones por correo electrónico.	2009-04-09	1669	Muy bueno	2	
Spring WS: Construcción de Clientes de Servicios Web con Spring	En este tutorial veremos las características que nos ofrece Spring para la construcción de clientes de servicios Web	2009-03-31	1436	Muy bueno	5	
Spring WS: Creación de Servicios Web con Spring	En este tutorial veremos un completo ejemplo de creación de un servicio web Contract-First con Spring y Maven	2009-03-15	1994	Muy bueno	6	
Web Service Security	En este tutorial vamos a ver como crear un servicio web seguro con autenticación mediante usuario y contraseña. WS-Security define cómo utilizar los tokens de seguridad, XML Signature y Xml Encryption en los mensajes SOAP para proporcionar autenticación,	2009-02-09	2435	Muy bueno	4	
Soap Monitor: Monitorización de mensajes SOAP en Axis2	En este tutorial aprenderemos a usar la herramienta SoapMonitor para monitorizar el tráfico de mensajes SOAP sobre Axis2	2009-01-18	1632	Muy bueno	3	
Weblets y como servir recursos que están en el CLASSPATH	En éste tutorial os enseñamos el uso del proyecto de Weblets, destinado a solucionar los problemas más comunes cuando queremos crear un componente Web y necesitamos recursos adicionales de diverso tipo.	2008-12-01	921	Muy bueno	1	
Web Services con Estado	Este tutorial muestra un ejemplo de un web service con estado (stateful webservice)	2008-10-27	3012	Muy bueno	10	
Web Services con Axis2. Configuración y ejemplo	Este es un tutorial básico que introduce a los servicios web en Java, y muestra cómo configurar el equipo con Eclipse, Apache Tomcat y Axis2 para poder crear luego un web service de ejemplo	2008-10-24	7737	Muy bueno	37	
Jersey: la implemetación de RESTFull de Sun	En este tutorial Germán nos enseña cómo usar RESTFull con la tecnología de Sun.	2008-04-05	1902	-	-	

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.