

# ¿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

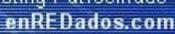
Tareas programadas (Quartz)  
 Gestor documental (Alfresco)  
 Inversión de control (Spring)

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)

	Powered by 	Hosting Patrocinado por  
---	--	--

[Home](#) | [Quienes Somos](#) | [Empleo](#) | [Tutoriales](#) | [Contacte](#)

	<p><b>Tutorial desarrollado por:</b> <a href="#">Carlos García Pérez</a></p> <p>Puedes encontrarme en <a href="#">Autentia</a> Somos expertos en Java/J2EE Contacta en <a href="mailto:carlos.garcia@autentia.com">carlos.garcia@autentia.com</a></p>	<p><a href="http://www.adictosaltrabajo.com">www.adictosaltrabajo.com</a> es el Web de difusión de conocimiento de <a href="http://www.autentia.com">www.autentia.com</a></p>  <p>real business solutions</p> <p><a href="#">Catálogo de cursos</a></p>
---	---	--

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

[Firma en nuestro libro de Visitas](#) <-----> [Asociarme al grupo AdictosAlTrabajo en eConozco](#)

#### Master Experto Java

100% alumnos trabajando Nuevo temario de Struts + Ajax  
[www.grupoatrium.com](http://www.grupoatrium.com)

#### Desarrollo Web con RPG400

Aplicaciones Gráficas HTML, Web y Cliente/Servidor en iSeries 400.  
[www.RpgForWeb.com](http://www.RpgForWeb.com)

[Anuncios Goooooogle](#)

[Anunciarse en este sitio](#)

## Introducción

Una necesidad común en el desarrollo de aplicaciones informáticas es el mapeo (conversión) entre objetos y documentos XML.

Normalmente esta tarea se realiza para poder utilizar otras tecnologías que trabajan con documentos XML como XSLT o para intercambiar información entre clientes y servidores.

En mi experiencia como programador he notado que esta tarea en la gran mayoría de los casos se realiza manualmente desde el código fuente de la aplicación o apoyándose en tecnologías como DOM y SAX.

En este tutorial realizo una presentación de una de las muchas APIS que nos facilitan esta tarea.

## ¿Qué es JOX?

JOX es una librería OpenSource con licencia LGPL de clases Java que nos permite convertir JavaBeans en documentos XML y viceversa.

Puedes obtener la librería JOX así como una documentación más extensa desde [aquí](#)

## ¿Qué puedo hacer con JOX?

- i. Crear JavaBeans desde InputStreams o Readers. (sockets, ficheros, buffers de memoria, etc)
- ii. Serializar JavaBeans con formato XML sobre OutputStream o Writers. (sockets, ficheros, buffers de memoria, etc)
- iii. Definición del tipo de mapping que deseamos mediante DTD. (Por ejemplo, que el elemento "firstname" de un xml se traduzca en la propiedad "nombre" de un JavaBean).

## ¿Ventajas del uso de JOX frente a otras librerías similares?

JOX es ideal para hacer mappings sencillos en donde no se quiere tener la sobrecarga (hablando en términos de Kilobytes) de otras librerías más potentes.

Por ejemplo, librerías como Apache Commons Digester son mucho más potentes y flexibles, pero tienen dependencias con otras librerías por lo que sobrecargan el proyecto y esto en determinadas circunstancias puede no ser deseable.

# Algunos ejemplos

Bueno, como el camino se hace al andar, vamos a ver unos **ejemplos autocomentados** del uso de la librería.

```
package com.autentia.tutoriales;

import com.wutka.jox.*;
import com.wutka.dtd.*;
import java.io.*;

/**
 * Ejemplo del uso de la librería JOX para el mapeo entre Beans y XML
 * @author Carlos Garcia. Autentia.
 */
public class JOXExampleApp {

    /**
     * @return Devuelve el XML a partir del cual creamos los objetos
     */
    public static String getXML() {
        StringBuffer xml = new StringBuffer(512);
        xml.append("<?xml version='1.0' standalone='yes'?>");
        xml.append("<Persona>");
        xml.append("<address><street>Honduras</street><number>30</number></address>");
        xml.append("<firstName>Joaquina</firstName>");
        xml.append("<lastNames>Pérez Guillén</lastNames>");
        xml.append("<weight>182</weight>");
        xml.append("</Persona>");
        return xml.toString();
    }

    /**
     * @return Devuelve el XML a partir del cual creamos el objeto
     */
    public static String getDTD() {
        StringBuffer dtd = new StringBuffer(512);
        dtd.append("<!ELEMENT Persona (firstName, lastNames)>");
        dtd.append("<!ELEMENT firstName (#PCDATA)>");
        dtd.append("<!ELEMENT lastNames (#PCDATA)>");
        dtd.append("<!ATTLIST Persona weight #REQUIRED>\n");
        return dtd.toString();
    }

    /**
     * Inicia el ejemplo
     */
    public static void main(String[] args) throws Exception {
        // Ejemplo 1: Construimos un objeto a partir de una definición XML
        String xml = JOXExampleApp.getXML();
        JOXBeanReader reader = new JOXBeanReader(new StringReader(xml));
        Persona person = (Persona) reader.readObject(Persona.class);

        /**
         * Imprimiré :
         * Nombre: Joaquina
         * Apellidos: Pérez
         * Peso: 182
         * Calle: Honduras
         * Número: 30
         */
        System.out.println("Nombre: " + person.getFirstName());
        System.out.println("Apellidos: " + person.getLastNames());
        System.out.println("Peso: " + person.getWeight());
        System.out.println("Calle: " + person.getAddress().getStreet());
        System.out.println("Número: " + person.getAddress().getNumber());

        // Ejemplo 2: Ahora pasamos de objeto a XML
        StringWriter writer = new StringWriter(1024);
        JOXBeanWriter joxOut = new JOXBeanWriter(writer);
        joxOut.writeObject("Persona", person);
        joxOut.close();

        /**
         * Se imprimirá:
         * <?xml version="1.0" encoding="ISO-8859-1"?>
         * <Persona>
         * <address>
         * <number>30</number>
         * <street>Honduras</street>
         * </address>
         * <firstName>Joaquina</firstName>
         * <lastNames>Pérez Guillén</lastNames>
         * <weight>182</weight>
         * </Persona>
         */
        System.out.println(writer.getBuffer().toString());
    }
}
```

```

writer.close();

// Ejemplo 3: Ahora pasamos de objeto a XML especificando el formato con
// una DTD
// El elemento Weight será un atributo de Persona
writer = new StringWriter(1024);
DTDParser parser = new DTDParser(new StringReader(getDTD()));
DTD dtd = parser.parse();

joxOut = new JOXBeanWriter(dtd, writer);
joxOut.writeObject("Persona", person);
joxOut.close();

/*
Imprimirá: (Observe el atributo weight)
<?xml version="1.0" encoding="ISO-8859-1"?>
<Persona weight="182">
    <firstName>Nombre como propiedad</firstName>
    <lastNames>Sanchez Luna</lastNames>
</Persona>
*/

System.out.println(writer.getBuffer().toString());
writer.close();
}
}

```

```

package com.autentia.tutoriales;

/**
 * Bean para el ejemplo.
 * @author Carlos García. Autentia
 */
public class Address {

    /**
     * Nombre de la calle
     */
    private String street;

    /**
     * Número de la calle
     */
    private int number;

    /**
     * Constructor por defecto
     */
    public Address() {}

    /**
     * @return Devuelve el número de la calle
     */
    public int getNumber() {
        return number;
    }

    /**
     * Establece el número de la calle
     * @param number Número
     */
    public void setNumber(int number) {
        this.number = number;
    }

    /**
     * @return Devuelve el nombre de la calle
     */
    public String getStreet() {
        return street;
    }

    /**
     * Establece el nombre de la calle
     * @param street Nombre de la calle
     */
    public void setStreet(String street) {
        this.street = street;
    }
}

```

```

package com.autentia.tutoriales;

/**
 * Bean para el ejemplo.
 * @author Carlos García. Autentia
 */
public class Persona {

    /**
     * Nombre de la persona
     */
    private String firstName;

    /**
     * Apellidos de la persona
     */
}

```

```

    */
    private String lastNames;

    /**
     * Peso en Kilogramos de la persona
     */
    private int     weight;

    /**
     * Dirección de la persona
     */
    private Address address;

    /**
     * Constructor por defecto
     */
    public Persona(){}

    /**
     * @return Devuelve el nombre de la persona
     */
    public String getFirstName() {
        return this.firstName;
    }

    /**
     * @return Devuelve los apellidos de la persona
     */
    public String getLastNames() {
        return this.lastNames;
    }

    /**
     * @return Devuelve el peso de la persona en gramos
     */
    public int getWeight() {
        return this.weight;
    }

    /**
     * Establece el nombre de la persona
     * @param firstName Nombre
     */
    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }

    /**
     * Establece los apellidos de la persona
     * @param lastNames Apellidos
     */
    public void setLastNames(String lastNames) {
        this.lastNames = lastNames;
    }

    /**
     * Establece el peso de la persona
     * @param weight Peso
     */
    public void setWeight(int weight) {
        this.weight = weight;
    }

    /**
     * @return Devuelve la dirección de la persona
     */
    public Address getAddress() {
        return address;
    }

    /**
     * Establece la dirección de la persona
     * @param addr Dirección
     */
    public void setAddress(Address addr) {
        this.address = addr;
    }
}

```

Bueno, espero que os haya sido de utilidad este tutorial.

En Autentia Real Business Solutions, nos gusta compartir el conocimiento. Aquí teneis un poquito más de nuestra aportación.

Si algún día necesitais ayuda con vuestros proyectos o necesitais formación, podéis encontraros en [Autentia](#)



This work is licensed under a [Creative Commons Attribution-NonCommercial-No Derivative Works 2.5 License](#).



[Puedes opinar sobre este tutorial aquí](#)

## Recuerda

que el personal de [Autentia](#) te regala la mayoría del conocimiento aquí compartido ([Ver todos los tutoriales](#))

¿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?**

[info@autentia.com](mailto:info@autentia.com)

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

**Autentia = Soporte a Desarrollo & Formación**

### Gestión de contenidos

[Autentia S.L.](#) Somos expertos en:

**J2EE, Struts, JSF, C++, OOP, UML, UP, Patrones de diseño ..**  
y muchas otras cosas

## Nuevo servicio de notificaciones

Si deseas que te enviemos un correo electrónico cuando introduzcamos nuevos tutoriales, inserta tu dirección de correo en el siguiente formulario.

<b>Subscribirse a Novedades</b>	
<b>e-mail</b>	
	<input type="button" value="Enviar"/>

## Otros Tutoriales Recomendados ([También ver todos](#))

### Nombre Corto

[Schemas XML. Introducción esquemas XML](#)

[XML básico](#)

[Técnicas básicas y poco comentadas en Java](#)

[Procesamiento XML en Java con JAXB y WSDP 1.6](#)

[Introducción a XPath](#)

[Transformación de XML y XSL en JSPs](#)

[Soporte XML en Eclipse con X-MEN](#)

[XML y XSL en Cliente](#)

[XMLEncryption en Java](#)

### Descripción

Los esquemas XML (schemas XML) son una evolución natural de las DTDs. Os mostramos como empezar con esta tecnología.

Si quieres ver de un modo visual como crear un documento XML, este es tu tutorial. Este es el primero de un conjunto de tutoriales que iremos publicando sobre esta fascinante y amplia tecnología

Os mostramos como realizar algunas cosas simples en Java: Formateo de decimales y enteros, gestión de preferencias y comparación entre objetos de nuevas clases

Os mostramos como instalar la versión 1.6 de WSDP y como procesar los ficheros XML con uno de sus componentes, JAXB

Este tutorial pretende ser una introducción a XPath mediante unos sencillos ejemplos.

Os mostramos como poder utilizar XML y XSL en JSPs, combinado con el Patrón MVC

Alejandro Perez nos enseña como potenciar el entorno eclipse para facilitarnos el trabajo con ficheros xml, gracias al plugin X-MEN

En este tutorial os enseñamos como formatear documentos XML directamente en vuestro navegador a través de Plantillas XSL. En cursos sucesivos veremos como hacerlo en el servidor, para no crear dependencias con el navegador del cliente.

En este magnífico tutorial, Alberto Carrasco nos enseña los fundamentos y un ejemplo práctico de XMLEncryption.

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](mailto:rcanales@adictosaltrabajo.com) para su resolución.

[Patrocinados por enredados.com](#) .... [Hosting en Castellano con soporte Java/J2EE](#)



www.AdictosAlTrabajo.com Optimizado 800X600