

# ¿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)



Hosting Patrocinado por  
enREDados.com



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

	<p><b>Tutorial desarrollado por:</b>  <b>Roberto Canales Mora 2003-2005</b>  <a href="#">Creador de AdictosAlTrabajo.com</a> y</p> <p><b>Director General de Autentia S.L.</b></p> <p><b>Recuerda que me puedes contratar para echarte una mano:</b></p> <p>Desarrollo y arquitectura Java/J2EE  Asesoramiento tecnológico Web  Formación / consultoría integrados en tu proyecto</p> <p>No te cortes y contacta: 655 99 11 72 <a href="mailto:rcanales@autentia.com">rcanales@autentia.com</a>.</p>	 <p>real business solutions</p>
---	--	--

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

**Master Java J2ee Oracle**

Prácticas laborales 100% aseguradas Nuevo temario de Struts. Trabaja ya  
[www.grupoatrium.com](http://www.grupoatrium.com)

**RPG apps to Java 2EE**

eCube Systems transforms AS/400 RPG legacy apps to Java and J2EE.  
[www.ecubesystems.com](http://www.ecubesystems.com)

**Java Servlet & XML Tool**

Edit JSP, XML, DTD, Schema, SOAP Easy-to-use, Download Free Trial!  
[www.altova.com](http://www.altova.com)

Anuncios Goooooogle

Anunciarse en este sitio

## Mejora de rendimiento en la serialización

En el desarrollo de las aplicaciones Java, de un modo habitual, escribimos en disco o transmitimos objetos completos.

En otro tutorial, os comentamos que una de las opciones de [comunicación entre applets y servlets](#) puede ser la serialización.

El modo de realizar ésta puede condicionar el rendimiento.

Os vamos a mostrar en un programa sencillo el modo tradicional y un modo optimizado a través del uso de buffer en memoria.

```

import java.io.*;
/**
 *
 * @author Roberto Canales
 */
public class serializacion
{
    static int veces = 10;

    public static void main(String[] args)
    {
        Persona array[] = new Persona[10000];

        for(int i=0;i< array.length; i++ )
        {
            array[i] = new Persona(i , "Nombre " + i);
        }

        long tiempoInicial = System.currentTimeMillis();

        ejecutaProceso1(array);

        long total = System.currentTimeMillis() - tiempoInicial;
        System.out.println("El tiempo a es " + total);

        tiempoInicial = System.currentTimeMillis();
        ejecutaProceso2(array);

        total = System.currentTimeMillis() - tiempoInicial;
        System.out.println("El tiempo b es " + total);
    }

    static void ejecutaProceso1(Persona [] array)
    {
        try
        {
            FileOutputStream outfisico = new FileOutputStream("salida1.txt");
            ObjectOutputStream salida = new ObjectOutputStream(outfisico);
        }
    }
}

```

```

        for(int i=0; i < veces; i++)
        {
            salida.writeObject(array);
            salida.write(i);
        }
    }
    catch(Exception e)
    {
    }
}

static void ejecutaProceso2(Persona [] array)
{
    try
    {
        FileOutputStream outfisico = new FileOutputStream("salida2.txt");
        ByteArrayOutputStream presalida = new ByteArrayOutputStream();
        ObjectOutputStream salida = new ObjectOutputStream(presalida);

        for(int i=0; i < veces; i++)
        {
            salida.writeObject(array);
            salida.write(i);
        }

        presalida.writeTo(outfisico);

    }
    catch(Exception e)
    {
    }
}
}

class Persona implements Serializable
{
    int id;
    String nombre;

    public Persona(){ }

    public Persona(int pId, String pNombre)
    {
        this.id = pId;
        this.nombre = pNombre;
    }
}
}

```

Podemos ver que la mejora de rendimiento es de un factor de 3

El tiempo a es 401  
El tiempo b es 130

Como norma general, siempre realizaremos entradas y salidas a través mediante streams que gestionen buffers.

A partir de la versión Java 1.4, aparece NIO (nuevo API de Input Output) que nos permite mejorar el rendimiento de nuestros procesos de entrada y salida.

A la hora de desarrollar aplicaciones Java, el uso de técnicas simples puede ayudarnos a mejorar el comportamiento de nuestro programas ....

Existen otras muchas técnicas que os iremos contando en otras ocasiones .....

---

Si desea contratar formación, consultoría o desarrollo de piezas a medida puede contactar con

## Formación en nuevas tecnologías

[Autentia S.L.](#) Somos expertos en:  
**J2EE, C++, OOP, UML, Vignette, Creatividad ..**  
 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.

e-mail	
	Enviar

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

Nombre Corto	Descripción
<a href="#">Medida del Rendimiento en aplicaciones J2EE</a>	Os mostramos como medir el rendimiento de vuestras aplicaciones Java J2E
<a href="#">Escritura log con Fichero UDP y JMS</a>	Os mostramos ejemplos para cuantificar el coste de escritura de Logs por pantalla, fichero, UDP y JMS (describiendo como configurar el entorno)
<a href="#">Analizar ejecución de programa Java</a>	Os mostramos como investigar el comportamiento de vuestros programas Java, en ejecución, a través del profiling.
<a href="#">Serialización Servlet-Applet</a>	Os mostramos un ejemplo para serializar una respuesta en la comunicación servlet-applet
<a href="#">Rendimiento de aplicaciones Web</a>	En este tutorial veremos una introducción al funcionamiento de la Suite e-Test de Empirix.

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