

¿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)



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

	<p>Tutorial desarrollado por: Roberto Canales Mora 2003-2005 Creador de AdictosAlTrabajo.com y</p> <p>Director General de Autentia S.L.</p> <p>Recuerda que me puedes contratar para echarte una mano:</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 rcanales@autentia.com.</p>	
---	--	---

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

[Curso Web J2EE](#)

Curso Avanzado en Desarrollo Web con J2EE

[XML to PDF in Java & J2EE](#)

Output PDF, PCL5, HTML in Java
J2EE Websphere, Weblogic, Tomcat, Jetty.

[JSP Tutorials](#)

JSP Made Easy With XMLSpy.
Syntax & Editing Help, Free D/L.

[Master Java J2ee Oracle](#)

Prácticas laborales 100% aseguradas
Nuevo temario de Struts. Trabaja ya

Anuncios Goooooogle

Anunciarse en este sitio

Caché avanzado de Objetos

Como podéis observar, poco a poco vamos subiendo el nivel de nuestros tutoriales. En uno de los últimos, os describíamos como empezar a pensar utilizando [patrones de diseño](#). Hoy vamos a continuar planteándonos como utilizar una solución de caché para Java.

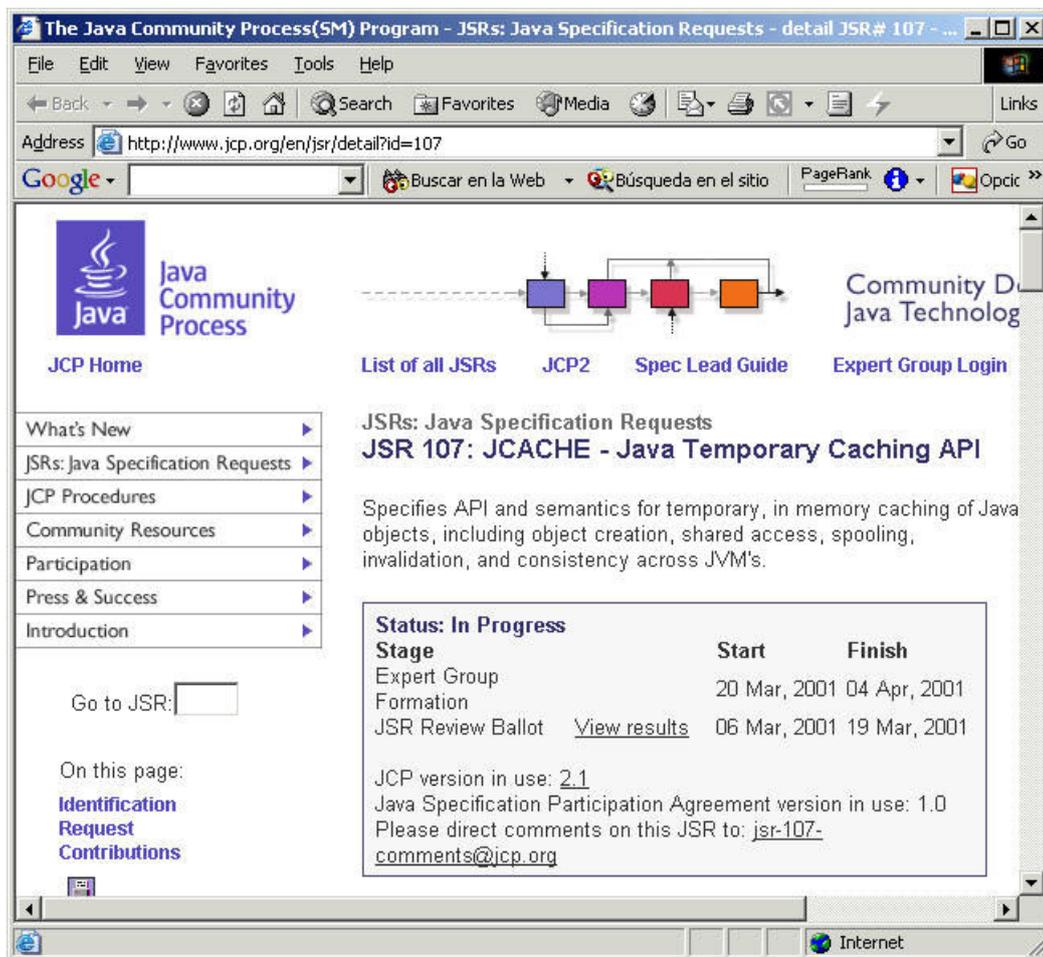
Antes de complicarnos la vida diseñando un sistema de caché, es muy posible que debiéramos buscar por Internet, en los sitios habituales, para ver que es lo que hay sobre el tema.

Sitios habituales son:

- Por supuesto, www.google.com (aunque es fácil perderse)
- www.aparache.org
- www.sourceforge.org

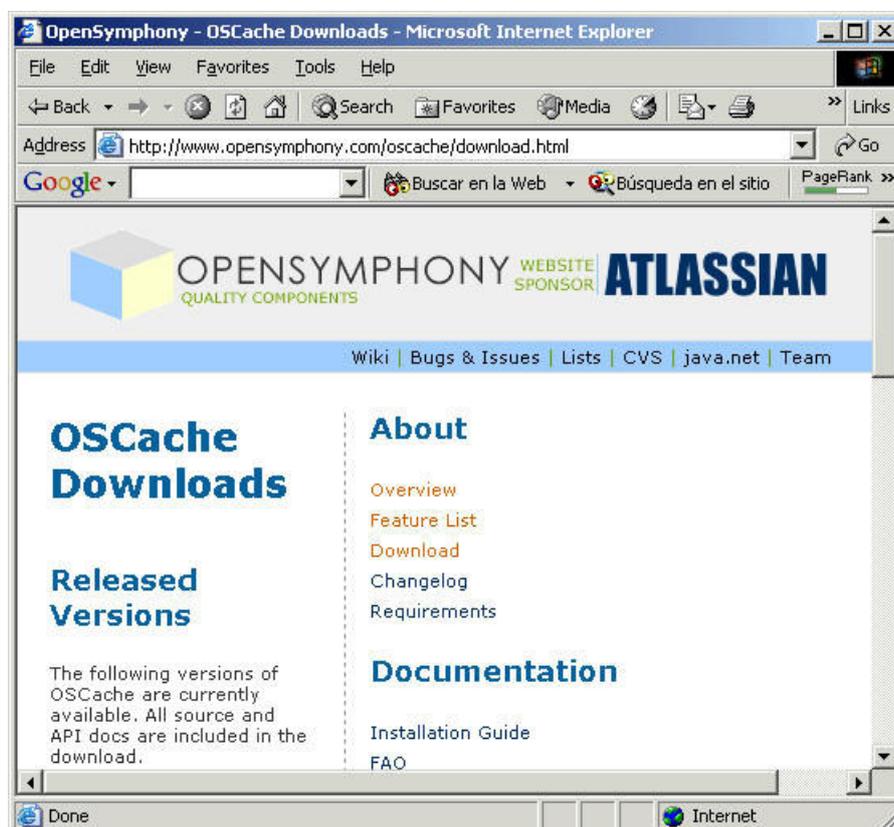
En el mundo Java, no debemos olvidar como se definen las especificaciones nuevas. Esto es a través de JCP (Proceso Comunitario Java)

Pues, que sorpresa ;-), que ya hay una especificación en marcha, la JSR 107....

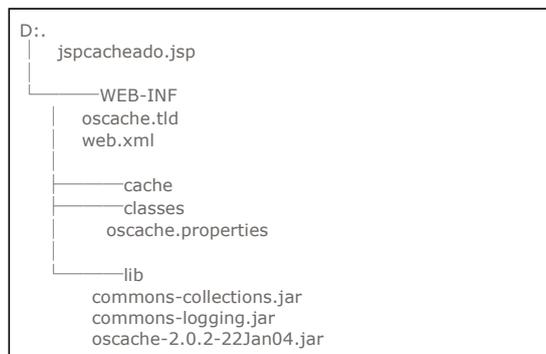


Buscando, podemos encontrar que hay un producto recomendado para obtener funcionalidades de caché, se llama OSCACHE.

Podemos encontrar este producto y otros con licencia gratuita en <http://www.opensymphony.com>



Una vez descargado el fichero ZIP con las librerías y documentación, para que nuestra prueba de concepto funcione, podemos establecer una estructura de ficheros como la descrita a continuación (cuesta un poquito llegar a esta conclusión siguiendo la documentación)



Debemos modificar el fichero web.xml para hacer referencia a la librería de TAGS:

```

<taglib>
  <taglib-uri>oscache</taglib-uri>
  <taglib-location>/WEB-INF/oscache.tld</taglib-location>
</taglib>
  
```

También tenemos que tocar el fichero de configuración **oscache.properties** e indicar el directorio donde cachear contenidos persistentes.

```

# CACHE DIRECTORY
#
# This is the directory on disk where caches will be stored by the DiskPersistenceListener.
# it will be created if it doesn't already exist. Remember that OSCache must have
# write permission to this directory.
#
# Note: for Windows machines, this needs \ to be escaped
# ie Windows:
# cache.path=c:\\myapp\\cache
# or *ix:
# cache.path=/opt/myapp/cache
#
cache.path=d:\\cacheadictos\\WEB-INF\\cache
  
```

Nuestro JSP, hace referencia a una librería de etiquetas para poder disponer del juego de TAGS que nos ayudarán en nuestro trabajo:

```
<%@ taglib uri="oscache" prefix="cache" %>
```

Declarar una área que queremos cachear es tan sencillo como incluir un bloque como este:

```

<cache:cache key="zona1" scope="application" duration="10">
    ...lo que sea
</cache:cache>
  
```

Dependiendo del tipo de aplicación y ámbito del elemento a cachear podemos jugar con los distintos modificadores: key, scope y duration (ver la documentación para más detalles y otras etiquetas relacionadas). Nosotros hemos especificado que queremos que todos los usuarios dispongan de este contenido cacheado y que se refresque cada 10 Segundos.

Para probar que se comporta como deseamos, vamos a poner un retardo en un área y ejecutar varias veces la consulta... el resultado os va a sorprender.

Como casi siempre en este Web, fijaos más en el fondo que en la forma.....

```

<%@ taglib uri="oscache" prefix="cache" %>
<%@page contentType="text/html"%>
<html>
<head><title>JSP Page</title></head>
<body>

<%! int contador = 0; %>

<center>
<h1>Ejemplos de Caches con OSCache</h1>

<% long inicio = System.currentTimeMillis();

contador ++;
%>
<br>

<cache:cache key="zona1" scope="application" duration="10">

<%

try
{
    Thread.currentThread().sleep(3000);
  
```

```
java.util.Date fecha = new java.util.Date();
out.println("Este es el mensaje" + fecha);
}
catch(Exception e){}
%>
</cache:cache>
<br>
<%
long fin = System.currentTimeMillis() - inicio;
out.println("El tiempo transcurrido completo es: " + fin
+ " milisegundos y el contador es: " + contador);
%>
</center>
</body>
</html>
```

Ésta es la salida de la página.....Podemos ver la primera ejecución



En una aplicación normal, el tiempo debería ser siempre superior a tres segundos pero vemos que efectivamente el tiempo de la segunda petición tiende a cero...



Conclusiones

Este ejemplo puede ser muy útil cuando trabajamos con el patrón de diseño MVC tipo 1 (JSPs que utiliza JavaBeans o etiquetas para acceder a estructuras de datos). Si seguimos investigando en el producto, pronto veremos que existe un modelo de objetos equivalente para aplicaciones que utilicen otro tipo de diseños como MVC tipo 2 o aplicaciones tradicionales.

Como comprobamos reiterativamente, necesitamos pararnos a probar y diseñar para que nuestro sistemas sean efectivos. Necesitamos tiempo ...

Yo tengo tiempo y tu empresa necesidades de formación o consultoría avanzada ¿hablamos? rcanales@autentia.com

[Sobre el Autor...](#)

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

J2EE, EJBs, Struts...

[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.

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

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

Nombre Corto	Descripción
Transformación de XML y XSL en JSPs	Os mostramos como poder utilizar XML y XSL en JSPs, combinado con el Patrón MVC
AspectJ, Programación con Aspectos	Os mostramos como configurar AspectJ (extensión Java para la programación basada en aspectos) y un pequeño ejemplo para medir la velocidad de una función sin alterar su código.
Forzar diálogo Guardar Como en JSPs	Os mostramos como afrontar tareas comunes en JSP: Forzar el diálogo Guardar Como al generar dinámicamente un fichero desde un JSP y asegurarnos que no se cachean nuestros ficheros (probado en IE Explorer 6)
CMP Entity Beans y MySql	Os mostramos como crear un Entity Bean con persistencia controlada por el servidor, configurado para usar MySql
Uso de JNDI, includes y cookies en Servlets	En este tutorial veremos como usar variables de entorno desde JNDI, incluir un servlet en otro (include) y como usar cookies en Servlets
Otra implementación JDO con TJDO	Os mostramos como montar un ejemplo simple de JDO, a través de la implementación gratuita TJDO
Novedades en Java 1.5	Ya está disponible la versión Beta del J2SDK 1.5. Os mostramos algunas de las nuevas características introducidas en el lenguaje Java: Clases genéricas, enumeraciones, bucles simplificados, etc.
JSP's y Modelo-Vista-Controlador	En este tutorial os enseñamos como crear un JSP, su relación con los servlets y como crear un ejemplo MVC en Tomcat
Generación automática de código JDBC	En este tutorial os enseñamos como, sin conocimiento de JDBC, crear vuestro programas en Java, gracias a JDBCTest.
Filtros de Servlets en Tomcat	En este tutorial os enseñamos la técnica (poco conocida) del encadenamiento de filtros en la activación de servlets, dentro del entorno Tomcat

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.

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

	<p>¿Buscas un hospedaje de calidad con soporte JAVA?</p>
---	---

www.AdictosAlTrabajo.com Optimizado 800X600