

# ¿Qué ofrece Autentia Real Business Solutions S.L?

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

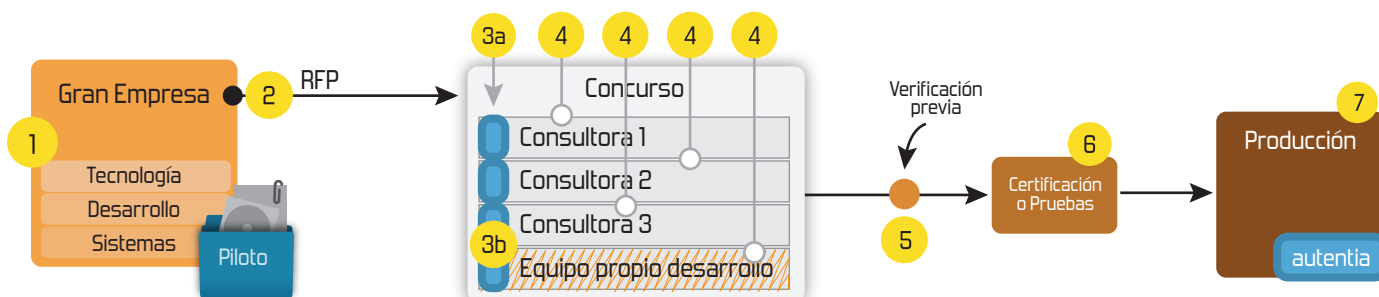
## 1. Desarrollo de componentes y proyectos a medida



## 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](#) | [Tutoriales](#) | [Contacte](#)

## Lanzamiento TNTConcept

[Autentia](#) da un paso más en su evolución: Lanzamiento de software propio. Ponemos a vuestra disposición el software que hemos construido para nuestra gestión interna, llamado TNTConcept (auTeNTia).

Construida con las últimas tecnologías de desarrollo Java/J2EE (Spring, JSF, Hibernate, Maven, Subversion, etc.) y disponible en licencia GPL, seguro que a muchos profesionales independientes y PYMES os ayudará a organizar mejor vuestra operativa.

**Las cosas grandes empiezan siendo algo pequeño .....** Saber más en: <http://tntconcept.sourceforge.net/>

|                                                                                                                                                                                                                                                                                     |                                                                                                                                                                                                                                                                                                                                                                |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><b>Tutorial desarrollado por:</b> <a href="#">Francisco Javier Martínez Páez</a></p> <p><b>Puedes encontrarme en <a href="#">Autentia</a></b><br/> <b>Somos expertos en Java/J2EE</b><br/> <b>Contacta en <a href="mailto:fjmpaez@autentia.com">fjmpaez@autentia.com</a></b></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><b>autentia</b><br/>real business solutions</p> <p><a href="#">Catálogo de cursos</a></p> |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

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

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

### Programación a medida

Desarrollo de Aplicaciones Java C, .Aplicaciones Web,etc, Tecsur  
[www.tecsur.es](http://www.tecsur.es)

### IT Mill Toolkit

a web user interface framework for developing quality web applications  
[www.itmill.com](http://www.itmill.com)

### Portal + BPM + ECM

Gestión unificada de personas, procesos y contenidos  
[www.polymita.com](http://www.polymita.com)

### Hosting Web - 4,90 €/mes

¡Prueba Gratuita sin compromiso!  
 .PHP, MySQL, PostgreSQL y Tomcat  
[silicontower.net](http://silicontower.net)

[Anuncios Google](#)

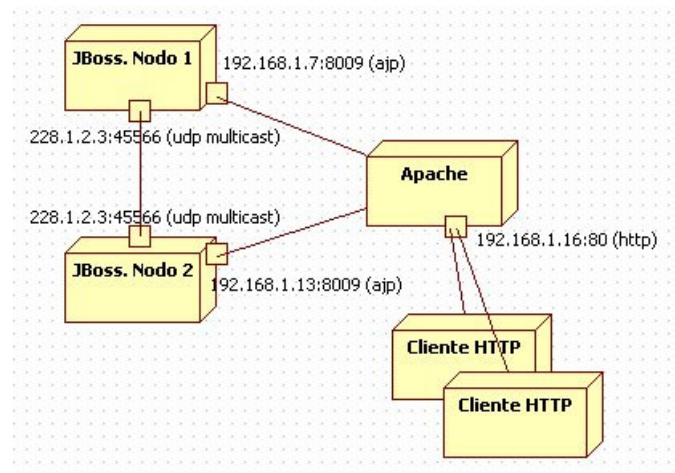
[Anunciarse en este sitio](#)

### APACHE, MODULO JK (mod\_jk) y JBOSS

Los ejemplos de este tutorial están hechos con el siguiente entorno de desarrollo:

- Jboss Eclipse IDE.
- JDK 1.5
- JBoss 4.0.5 GA
- Apache 2.2.4

En este tutorial os vamos a enseñar a instalar un servidor apache, instalar el módulo JK para comunicar con dos servidores JBoss en cluster, para conseguir balanceo de carga y replicación de sesión (el balanceo de carga nos lo proporcionará el módulo JK de apache, y la replicación de la sesión JBoss). Suponemos que disponemos de tres máquinas. En dos de ellas tendremos instaladas un servidor JBoss 4.0.5 que arrancaremos en cluster. En la otra máquina tenemos instalado el servidor apache 2.2.4 . La arquitectura es la siguiente:



En este tutorial no enseñaré a instalar ni Apache ni JBoss, ya que no requiere su instalación ningún conocimiento especial. Os muestro los enlaces donde descargar ambas aplicaciones:

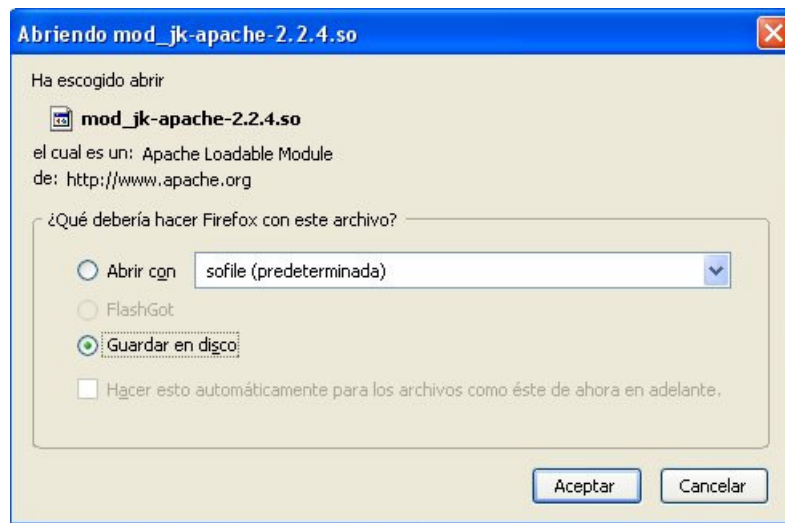
<http://labs.jboss.com/portal/jbossas/download>

<http://httpd.apache.org/download.cgi>

Lo que si necesitaremos descargar es el mod\_jk y os enseñaré a instalarlo:

### INSTALACIÓN Y CONFIGURACIÓN DEL MÓDULO JK

Desde <http://www.apache.org/dist/tomcat/tomcat-connectors/jk/binaries/> seleccionaremos la carpeta en función de donde esté instalado nuestro servidor apache (en mi caso win32) seleccionaremos la versión 1.2.21 (versión estable en el momento del tutorial) y elegiremos la librería a descargar en función de la versión de nuestro apache (en nuestro caso: mod\_jk-apache-2.2.4.so )



A continuación, modificaremos el nombre del fichero que nos acabamos de descargar y le llamaremos: **mod\_jk.so** y lo copiaremos a la ruta de nuestro servidor apache: <APACHE\_HOME>/modules

Una vez copiado, abriremos el fichero de configuración de apache: <APACHE\_HOME>/conf/httpd.conf e incluiremos la siguiente línea al final del fichero:

```
# Include mod_jk configuration file
Include conf/mod-jk.conf
```

Crearemos el fichero **mod-jk.conf** tal y como hemos indicado en el fichero de configuración anterior en: <APACHE\_HOME>/conf . Debajo muestro el contenido del fichero.

```
# Load mod_jk module
# Specify the filename of the mod_jk lib
LoadModule jk_module modules/mod_jk.so

# Where to find workers.properties
JkWorkersFile conf/workers.properties

# Where to put jk logs
JkLogFile logs/mod_jk.log

# Set the jk log level [debug/error/info]
JkLogLevel info

# Select the log format
JkLogStampFormat "[%a %b %d %H:%M:%S %Y]"

# JkOptions indicates to send SSK KEY SIZE
JkOptions +ForwardKeySize +ForwardURISCompat -ForwardDirectories

# JkRequestLogFormat
JkRequestLogFormat "%w %V %T"

# Mount your applications
JkMount /application/* loadbalancer

# You can use external file for mount points.
# It will be checked for updates each 60 seconds.
# The format of the file is: /url=worker
# /examples/*=loadbalancer
JkMountFile conf/uriworkmap.properties

# Add shared memory.
# This directive is present with 1.2.10 and
# later versions of mod_jk, and is needed for
# for load balancing to work properly
JkShmFile logs/jk.shm

# Add jkstatus for managing runtime data
<Location /jkstatus/>
JkMount status
Order deny,allow
Deny from all
Allow from 127.0.0.1
</Location>
```

Ahora, crearemos el fichero **workers.properties** donde configuraremos los nodos a los que vamos a balancear la carga en <APACHE\_HOME>/conf:

```
# Define list of workers that will be used
# for mapping requests
# The configuration directives are valid
# for the mod_jk version 1.2.18 and later
worker.list=loadbalancer,status

# Definimos el nodo Nod01
# Puerto del conector ajp de nuestro tomcat (JBoss)
worker.nod01.port=8009
# Ip del nodo 1.
worker.nod01.host=192.168.1.7
worker.nod01.type=ajp13
# Peso de nuestro nodo. A más peso, mas peticiones recibe.
worker.nod01.lbfactor=1

# Definimos el nodo Nod02
worker.nod02.port=8009
worker.nod02.host=192.168.1.13
worker.nod02.type=ajp13
worker.nod02.lbfactor=1

# Load-balancing behaviour
worker.loadbalancer.type=lb
worker.loadbalancer.balance_workers=nod01,nod02

# Status worker for managing load balancer
worker.status.type=status
```

Ahora, crearemos el fichero **uriworkermap.properties** donde configuraremos las aplicaciones a balancear en <APACHE\_HOME>/conf:

```
# Simple worker configuration file
# Mount the Servlet context to the ajp13 worker
/jmx-console=loadbalancer
/jmx-console/*=loadbalancer
/web-console=loadbalancer
/web-console/*=loadbalancer
/ejemplo=loadbalancer
/ejemplo/*=loadbalancer
```

Hemos configurado para balancear las consolas de JBoss, así como una aplicación que crearemos posteriormente cuyo context-path es **ejemplo**.

El conector mod\_jk usa sticky-sessions por defecto para balancear la carga, lo que quiere decir que siempre intentará enviar al mismo nodo todas las peticiones con la misma sesión (esto es configurable).

Ya tenemos instalado y configurado Apache y el módulo jk. Arrancaremos apache para comprobar que no hemos cometido ningún error:



Podemos ver en la parte posterior como nos indica que el mod\_jk está cargado.

### CONFIGURACIÓN DE LOS NODOS EN JBOSS

Para información sobre temas de cluster en JBoss podéis ver el tutorial:

<http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=HAScheduler>

Lo primero que haremos es configurar los nodos de JBoss para arrancar en cluster. Para trabajar en cluster, lo haremos en la configuración *all* (*run.bat -c all*) de JBoss.

Configuraremos el cluster en ambas máquinas. Lo primero que haremos será editar el fichero: <JBoss\_HOME>/server/all/deploy/**cluster-service.xml**, para configurar el cluster (usaremos UDP multicast). Si no lo hemos tocado nunca o la instalación es nueva en ambos casos es probable que no haga falta tocar nada. Os muestro a continuación la parte que nos interesa del fichero en ambas máquinas del cluster:

```
....
....
<!-- The JGroups protocol configuration -->
<attribute name="PartitionConfig">
  <Config>
    <UDP mcast_addr="{jboss.partition.udpGroup:228.1.2.3}" mcast_port="45566"
      ip_ttl="{jgroups.mcast.ip_ttl:8}" ip_mcast="true"
      mcast_rcv_buf_size="2000000" mcast_snd_buf_size="640000"
      ucast_rcv_buf_size="2000000" ucast_snd_buf_size="640000"
```

```

        loopback="false"/>
<PING timeout="2000" num_initial_members="3"
  up_thread="true" down_thread="true"/>
<MERGE2 min_interval="10000" max_interval="20000"/>
<FD_SOCKET down_thread="false" up_thread="false"/>
<FD shun="true" up_thread="true" down_thread="true"
  timeout="10000" max_tries="5"/>
<VERIFY_SUSPECT timeout="3000" num_msgs="3"
  up_thread="true" down_thread="true"/>
<pbcast.NAKACK gc_lag="50" retransmit_timeout="300,600,1200,2400,4800"
  max_xmit_size="8192"
  up_thread="true" down_thread="true"/>
<UNICAST timeout="300,600,1200,2400,4800" window_size="100" min_threshold="10"
  down_thread="true"/>
<pbcast.STABLE desired_avg_gossip="20000" max_bytes="400000"
  up_thread="true" down_thread="true"/>
<FRAG frag_size="8192"
  down_thread="true" up_thread="true"/>
<pbcast.GMS join_timeout="5000" join_retry_timeout="2000"
  shun="true" print_local_addr="true"/>
<pbcast.STATE_TRANSFER up_thread="true" down_thread="true"/>
</Config>

<!-- Alternate TCP stack: customize it for your environment, change bind_addr and initial_hosts -->
<!--
<Config>

```

```

....
....

```

Comprobad que tenéis comentada la parte de TCP y descomentada la parte de UDP. Comprobad que ambas máquinas tienen la misma IP y puerto multicast:

```
mcast_addr="{jboss.partition.udpGroup:228.1.2.3}"           mcast_port="45566"
```

Comprobad en la parte superior del fichero que ambas máquinas pertenecen al mismo grupo del cluster (mismo nombre):

```

....
<!-- Name of the partition being built -->
<attribute name="PartitionName">
  ${jboss.partition.name:DefaultPartition}
</attribute>
....

```

Ahora configuraremos el tomcat de JBoss de ambos nodos para indicarle que vamos a usar mod-jk. Además, vamos a desactivar los conectores http:8080 de ambos nodos para impedir que se acceda directamente a los servidores de aplicaciones.

Editaremos el fichero:

JBOSS\_HOME/server/all/deploy/jbossweb-tomcat50.sar/**server.xml**  
y modificaremos lo siguiente:

En ambos nodos:

Comentaremos esta parte:

```

<!-- A HTTP/1.1 Connector on port 8080
  <Connector port="8080" address="{jboss.bind.address}"
    maxThreads="250" strategy="ms" maxHttpHeaderSize="8192"
    emptySessionPath="true"
    enableLookups="false" redirectPort="8443" acceptCount="100"
    connectionTimeout="20000" disableUploadTimeout="true"/>
-->

```

Comprobaremos que esta parte está descomentada:

```

<!-- A AJP 1.3 Connector on port 8009 -->
  <Connector port="8009" address="{jboss.bind.address}"
    emptySessionPath="true" enableLookups="false" redirectPort="8443"
    protocol="AJP/1.3"/>

```

En el nodo 1, añadiremos lo siguiente a la etiqueta *engine*:

```

....
<Engine name="jboss.web" defaultHost="localhost" jvmRoute="nodo1">
....

```

En el nodo 2:

```

....
<Engine name="jboss.web" defaultHost="localhost" jvmRoute="nodo2">
....

```

Ahora editaremos el fichero:

JBOSS\_HOME/server/all/deploy/jbossweb-tomcat50.sar/META-INF/**jboss-service.xml**  
y le indicaremos que vamos a usar mod-jk en ambas máquinas:

```

....
<attribute name="UseJK">true</attribute>
....

```

A continuación, vamos a configurar el cluster en el tomcat de JBoss de ambas máquinas.

Editaremos el fichero:

JBOSS\_HOME/server/all/deploy/tc5-cluster.sar/META-INF/**jboss-service.xml**

y comprobaremos que la parte de configuración del cluster está igual en ambos nodos y tiene comentada la parte de TCP y descomentada la de UDP:

```

....
<attribute name="ClusterConfig">
  <config>
    <UDP mcast_addr="{jboss.partition.udpGroup:230.1.2.7}"
      mcast_port="45577"
      ucast_rcv_buf_size="20000000"
      ucast_snd_buf_size="640000"
      mcast_rcv_buf_size="25000000"
      mcast_snd_buf_size="640000"
      loopback="false"
      max_bundle_size="64000"
      max_bundle_timeout="30"
      use_incoming_packet_handler="true"
      use_outgoing_packet_handler="true"
    >
  </config>
</attribute>

```

```

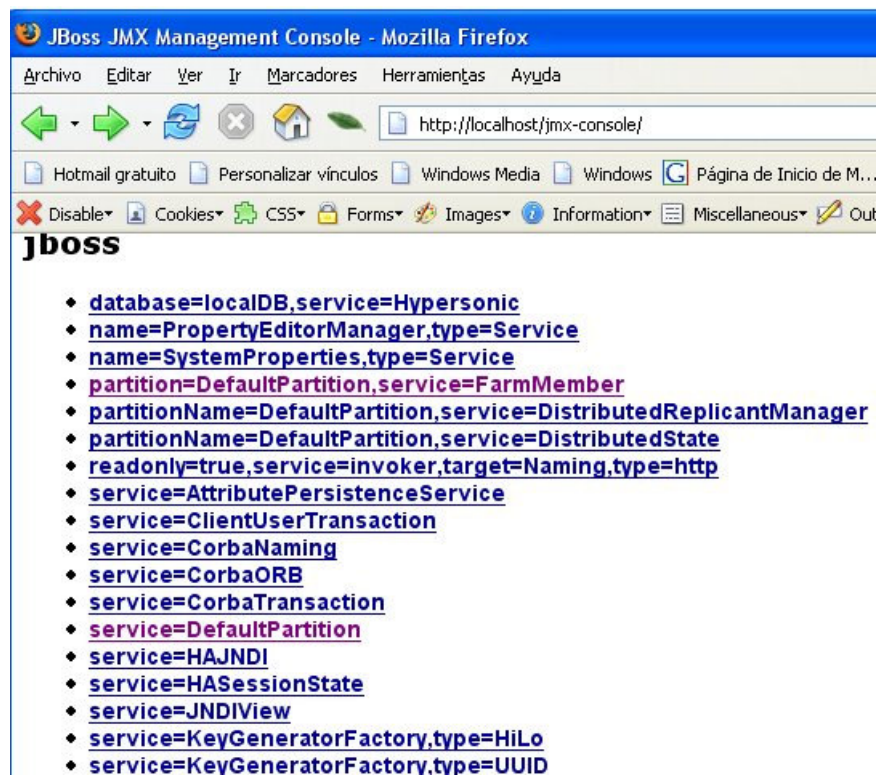
    ip_ttl="${jgroups.mcast.ip_ttl:2}"
    down_thread="false" up_thread="false"
    enable_bundling="true"/>
<PING timeout="2000"
    down_thread="false" up_thread="false" num_initial_members="3"/>
<MERGE2 max_interval="100000"
    down_thread="false" up_thread="false" min_interval="20000"/>
<FD_SOCK down_thread="false" up_thread="false"/>
<FD shun="true" up_thread="false" down_thread="false"
    timeout="20000" max_tries="5"/>
<VERIFY_SUSPECT timeout="1500"
    up_thread="false" down_thread="false"/>
<pbcast.NAKACK max_xmit_size="60000"
    use_mcast_xmit="false" gc_lag="50"
    retransmit_timeout="300,600,1200,2400,4800"
    down_thread="false" up_thread="false"
    discard_delivered_msgs="true"/>
<UNICAST timeout="300,600,1200,2400,3600"
    down_thread="false" up_thread="false"/>
<pbcast.STABLE stability_delay="1000" desired_avg_gossip="50000"
    down_thread="false" up_thread="false"
    max_bytes="400000"/>
<pbcast.GMS print_local_addr="true" join_timeout="3000"
    down_thread="false" up_thread="false"
    join_retry_timeout="2000" shun="true"/>
<FC max_credits="2000000" down_thread="false" up_thread="false"
    min_threshold="0.10"/>
<FRAG2 frag_size="60000" down_thread="false" up_thread="false"/>
<pbcast.STATE_TRANSFER down_thread="false" up_thread="false"/>
</config>

```

....

En este mismo fichero se configura la replicación de sesión, en principio no tocaremos nada.

Ahora no nos queda más que arrancar los servidores JBoss. Comprobaremos que las máquinas del cluster se ven. Para empezar, entraremos en la consola jmx de alguno de ellos (no sabemos cual será ya que el balanceador nos llevará a alguno de ellos)



Seleccionaremos el MBean jboss:service=DefaultPartition para ver la configuración del cluster:

| Type                                          | Access | Value                                                             |
|-----------------------------------------------|--------|-------------------------------------------------------------------|
| java.lang.String                              | RW     | DefaultPartition                                                  |
| int                                           | R      | 3                                                                 |
| long                                          | RW     | 30000                                                             |
| boolean                                       | RW     | <input type="radio"/> True <input checked="" type="radio"/> False |
| java.util.Vector                              | R      | [192.168.1.13:1099, 192.168.1.7:1099]                             |
| java.lang.String                              | RW     | UDP(ip_mcast=true;ip_#                                            |
| org.jboss.ha.framework.interfaces.HAPartition | R      | org.jboss.ha.framework.server.HAPartition                         |
| java.lang.String                              | R      | Started                                                           |
| java.net.InetAddress                          | RW     | /0.0.0.0                                                          |
| java.lang.String                              | RW     | 192.168.1.7:1099                                                  |
| long                                          | RW     | 60000                                                             |

Podemos ver las dos máquinas del cluster. Podéis comprobar que no podemos acceder directamente al puerto 8080 de nuestros Jboss porque los hemos deshabilitado (probamos por ejemplo la máquina 192.168.1.7:8080:



Hemos comprobado que el balanceo está funcionando. Vamos ahora a probar la replicación de sesión.

#### **APLICACIÓN WEB CON REPLICACIÓN DE SESIÓN.**

Vamos a crearnos una pequeña aplicación web que sólo va a contener un index.jsp como página de inicio. La aplicación la empaquetaremos en un fichero llamado ejemplo.war.

Os muestro el código de **index.jsp**:

**METO EN SESION UN CONTADOR.**<br>

```
<%
Integer contador = (Integer)session.getAttribute("CONTADOR");
if(contador==null) {
    contador = new Integer(0);
}
contador = new Integer(contador.intValue()+1);
session.setAttribute("CONTADOR",contador);
System.out.println("Salgo en la consola del servidor que sirve la jsp");
%>
<br>
El contador vale: <%=session.getAttribute("CONTADOR")%>
```

Os muestro el web.xml:

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<!DOCTYPE web-app
PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.2//EN"
"http://java.sun.com/j2ee/dtds/web-app_2_4.dtd">

<web-app>
  <distributable/>
  <display-name>Ejemplo</display-name>
  <welcome-file-list>
    <welcome-file>index.jsp</welcome-file>
  </welcome-file-list>
</web-app>
```

Indicamos con el tag distributable que queremos habilitar la replicación de sesión para nuestra aplicación.

Necesitamos también el descriptor jboss-web.xml para configurar la manera de la replicación:

```
<jboss-web>
  <replication-config>
    <replication-trigger>SET_AND_NON_PRIMITIVE_GET</replication-trigger>
    <replication-granularity>SESSION</replication-granularity>
    <replication-field-batch-mode>true</replication-field-batch-mode>
  </replication-config>
</jboss-web>
```

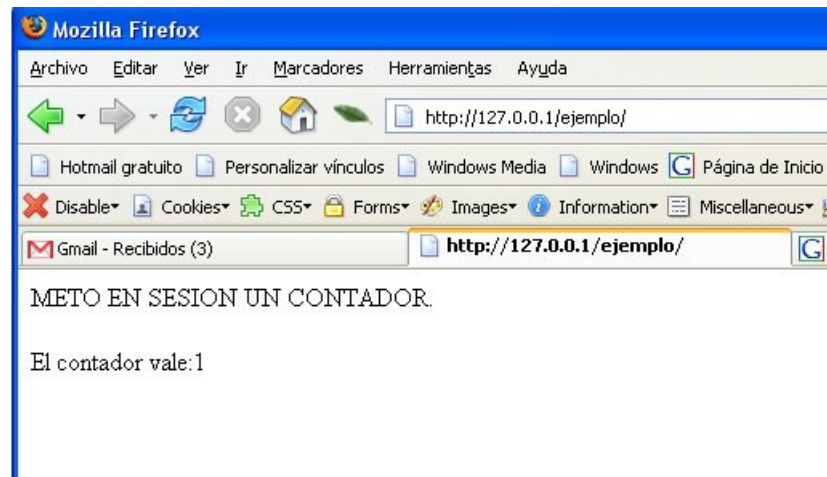
Empaquetamos la aplicación en ejemplo.war:

```
- ejemplo.war
+ WEB-INF/
  - web.xml
  - jboss-web.xml
- index.jsp
```

La desplegamos en la granja de servidores jboss, es decir, la copiamos en el directorio JBOSS\_HOME/server/all/**farm** de alguno de ellos:

```
C:\WINDOWS\system32\cmd.exe - run.bat -c all
10:50:51.970 INFO [ConnectionFactoryBindingService] Bound ConnectionManager 'jboss.jca:service=ConnectionFactoryBinding,name=JmsXA' to JNDI name 'java:JmsXA'
10:50:52.267 INFO [TomcatDeployer] deploy, ctxPath=/jmx-console, warUrl=.../deploy/jmx-console.war/
10:50:52.704 INFO [FarmMemberService] **** pullNewDeployments ****
10:50:52.704 INFO [ClusterFileTransfer] Start pull of file cluster-examples-service.xml from cluster.
10:50:52.751 INFO [ClusterFileTransfer] Finished cluster pull of file cluster-examples-service.xml to cluster-examples-service.xml
10:50:53.064 INFO [ChannelSocket] JK: ajp13 listening on /0.0.0.0:8009
10:50:53.095 INFO [JkMain] Jk running ID=0 time=0/94 config=null
10:50:53.220 INFO [Server] JBoss (MX MicroKernel) [4.0.5.GA (build: CVSTag=Branch_4_0 date=200610162339)] Started in 55s:187ms
11:35:33.001 INFO [TomcatDeployer] deploy, ctxPath=/ejemplo, warUrl=.../tmp/deploy/tmp5443ejemplo-exp.war/
11:35:33.329 INFO [JBossCacheManager] init(): replicationGranularity_ is 0 and invalidateSessionPolicy is 2
11:35:33.439 INFO [JBossCacheManager] Starting JBossManager
11:35:33.454 INFO [JBossCacheManager] We are using mod_jk(2) for load-balancing. Will add JvmRouteValue.
11:35:33.517 INFO [ClusterFileTransfer] Start push of file ejemplo.war to cluster.
11:35:33.626 INFO [ClusterFileTransfer] Finished push of file ejemplo.war to cluster.
```

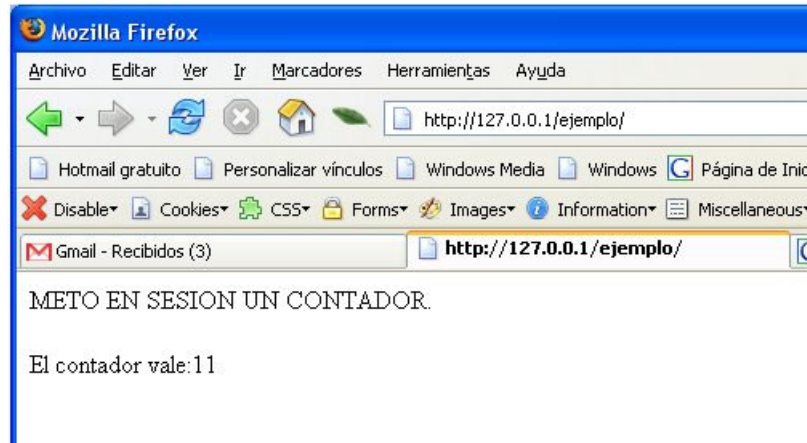
Vemos que se ha desplegado en el cluster. Vamos a acceder a la aplicación a través de apache:



En la consola del servidor que elija el balanceador se mostrará el mensaje que sacamos con System.out.

Accedemos varias veces para aumentar el contador hasta 10 (veréis como el nodo elegido siempre es el mismo)

Parad ahora el servidor que estaba sirviendo las peticiones. Y probad a entrar de nuevo a la página:



Comprobaréis dos cosas:

1. Ahora el balanceador detecta que el servidor al que dirigía las peticiones está caído y lo redirige al otro nodo (mirad la consola del que está levantado)
2. El contador no empieza en uno, sino que se mantiene la sesión valiendo 11.

Bueno, pues parece que no es tan difícil conseguir una arquitectura medianamente profesional. Espero que os haya gustado.

No obstante si necesitáis ayuda, ya sabéis: <http://www.autentia.com>



This work is licensed under a [Creative Commons Attribution-Noncommercial-No Derivative Works 2.5 License](http://creativecommons.org/licenses/by-nc-nd/2.5/).  
[Puedes opinar sobre este tutorial aquí](#)



## Recuerda

que el personal de [Autentia](http://www.autentia.com) te regala la mayoría del conocimiento aquí compartido ([Ver todos los tutoriales](#))

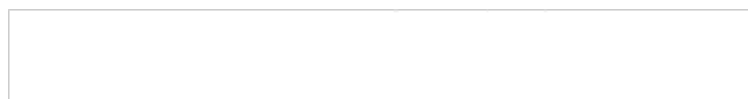
¿Nos vas a tener en cuenta cuando necesites consultoría o formación en tu empresa?

**¿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**



[Autentia S.L.](http://www.autentia.com) 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.

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

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

Nombre Corto

Descripción

[mod\\_jk en Ubuntu / Apache2-JBoss](#)

Os mostramos como instalar el conector mod\_jk sobre la distribución linux Ubuntu

|                                                      |                                                                                                                                                 |
|------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                      | utilizando Apache2 y JBoss                                                                                                                      |
| <a href="#">Apache Commons Configuration</a>         | En este tutorial os vamos a enseñar a utilizar una API de Apache para gestionar las configuraciones de vuestras aplicaciones de manera avanzada |
| <a href="#">Activación de la seguridad en Apache</a> | Alejandro Pérez nos enseña como securizar Apache a través de autenticación básica y certificados de seguridad SSL.                              |
| <a href="#">mod_jk en WindowsXP / Apache2-JBoss</a>  | Os mostramos como instalar el conector mod_jk sobre WindowsXP utilizando Apache2 y JBoss                                                        |
| <a href="#">Instalar JBoss</a>                       | Os mostramos como instalar en servidor gratuito de aplicaciones JBOSS así como a automatizar su arranque y parada.                              |
| <a href="#">mod_jk en WindowsXP / ISS-JBoss</a>      | Os mostramos como instalar el conector mod_jk sobre WindowsXP utilizando ISS y JBoss                                                            |
| <a href="#">mod_jk en Linux / Apache2-JBoss</a>      | El conector mod_jk se encarga de enviar las peticiones dinámicas de Apache2 a un servidor de aplicaciones JBoss                                 |
| <a href="#">Apache, MySQL y PHP</a>                  | Os mostramos como configurar Apache, MySQL y PHP en vuestra máquina                                                                             |

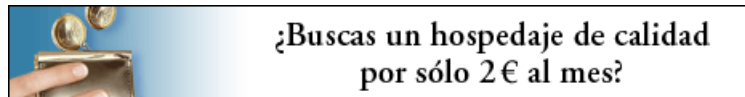
Nota: Los tutoriales mostrados en este Web tienen como objetivo la difusión del conocimiento.

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](#)



www.AdictosAlTrabajo.com Optimizado 800X600