

¿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: Carlos León Villamayor</p> <p>Puedes encontrarme en Autentia</p>	 <p>autentia real business solutions</p>
--	--

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

CCS

Centro de Cálculo de Sabadell 1ª empresa
española de software
www.ccs.biz

Nexedi ERP5

Consulting and development services Open
Source ERP5, CRM, CPS, Zope
www.nexedi.com

Eniac S.A. desde 1985

Software hardware y servicios para empresas
y profesionales
www.eniac.es

Anuncios Goooooogle

Anunciarse en este sitio

Instalación de mod_jk en la plataforma WindowsXP Apache2-JBoss

En [Autentia](#) estamos preparados para cualquier tipo de solución referente a nuevas tecnologías, en este caso, la instalación del **mod_jk** con **Apache2** como servidor web y **JBoss** como servidor de aplicaciones sobre el Sistema Operativo **Windows**.

Introducción:

El conector "[mod_jk](#)" se encarga de enviar las peticiones que nosotros configuremos desde un servidor web a un servidor de aplicaciones. Esta estrategia se suele utilizar para liberar de carga al servidor de aplicaciones; para ello dejaremos que la parte estática de nuestra aplicación web sea procesada por el servidor [Apache2](#), mientras que la parte dinámica sea gestionada por el servidor de aplicaciones [JBoss](#). También se podría aplicar para dejar nuestro servidor de aplicaciones oculto. En este caso de ejemplo tan solo tendremos acceso a las consolas de administración de **JBoss**.

Necesitamos la descarga del siguiente software (ver los enlaces a pie de página):

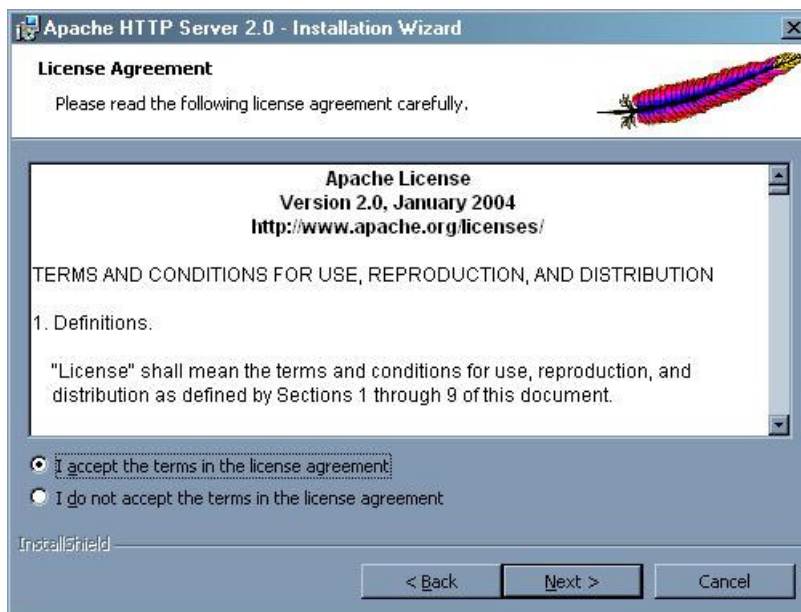
- Distribución de **J2SE 1.4.X** (no es recomendable usar todavía la 1.5 con JBoss).
- Conector **mod_jk** (en este tutorial usaremos la versión 1.2.15).
- Servidor Web **Apache2** (en este tutorial utilizaremos la versión 2.0.55).
- Servidor de aplicaciones **JBoss** (en este tutorial utilizaremos la versión 4.0.3 SP1).

Seguiremos el siguiente orden de instalación:

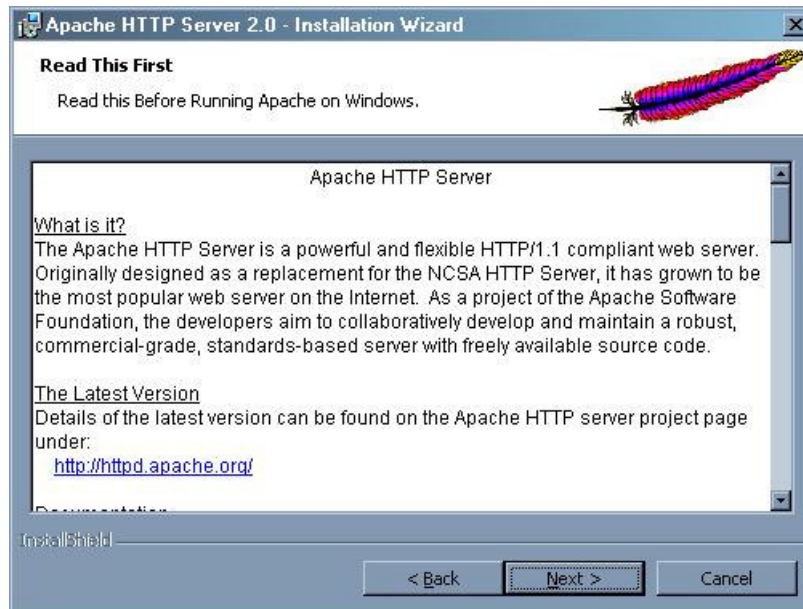
1. **Instalación del servidor Web Apache2 para windows:**
Una vez descargada la distribución (buscar debajo en enlaces), ejecutamos el archivo "**apache_2.0.55-win32-x86-no_ssl.msi**" y obtendremos una ventana como esta:



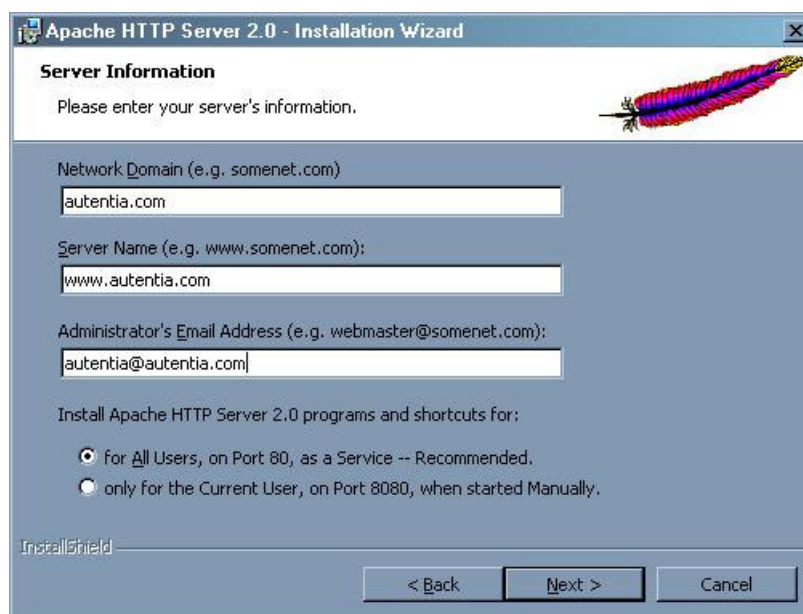
Pulsamos sobre **"Next"** y aparecerá una ventana como esta:



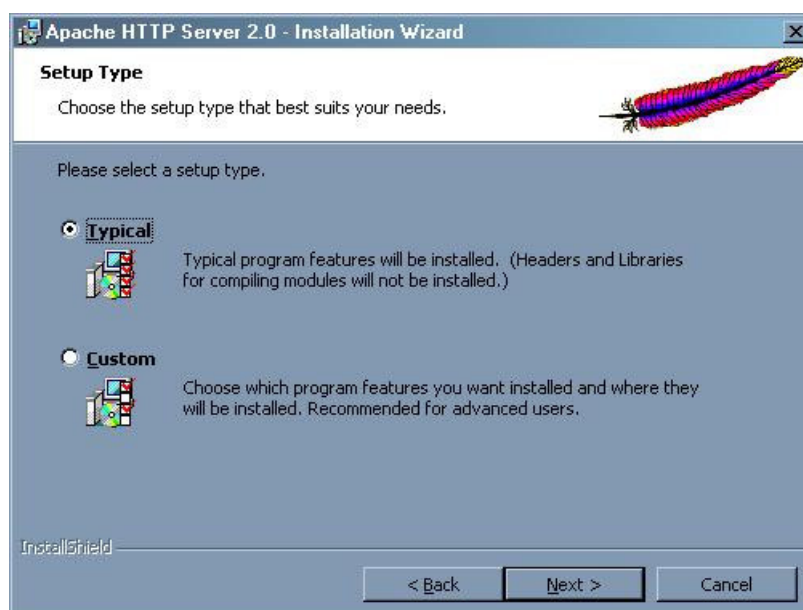
Aceptamos los términos y pulsamos sobre **"Next"**.



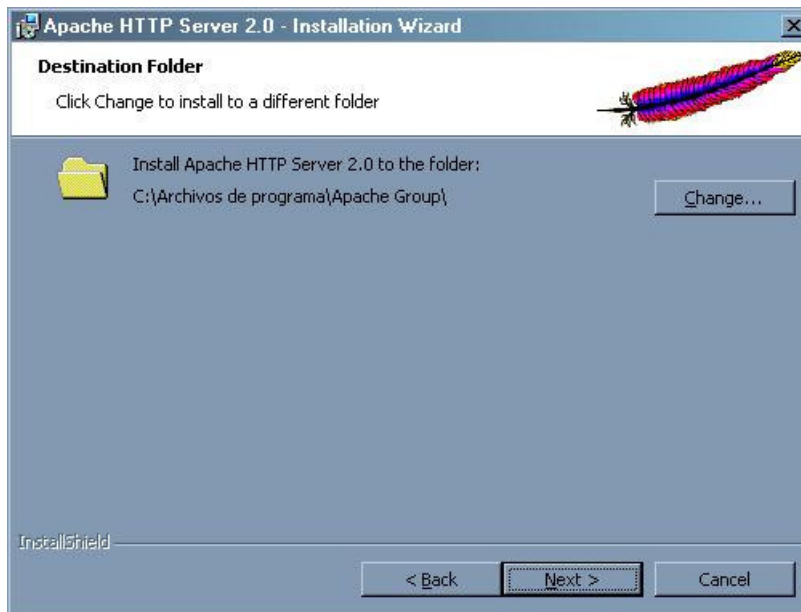
Nos aparece una documentación en la cual nos hacen una breve descripción del servidor. Pulsamos sobre **"Next"**.



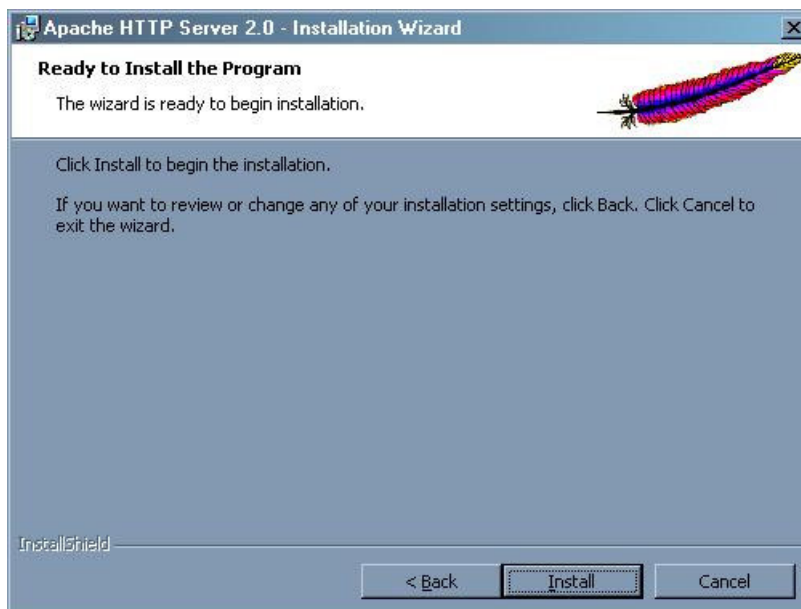
Rellenamos los datos según nuestro entorno y pulsamos sobre **"Next"**.



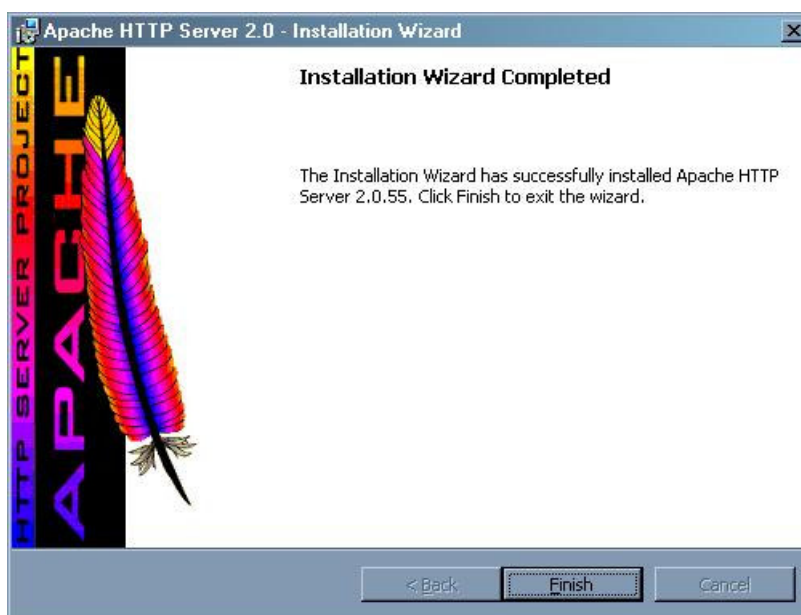
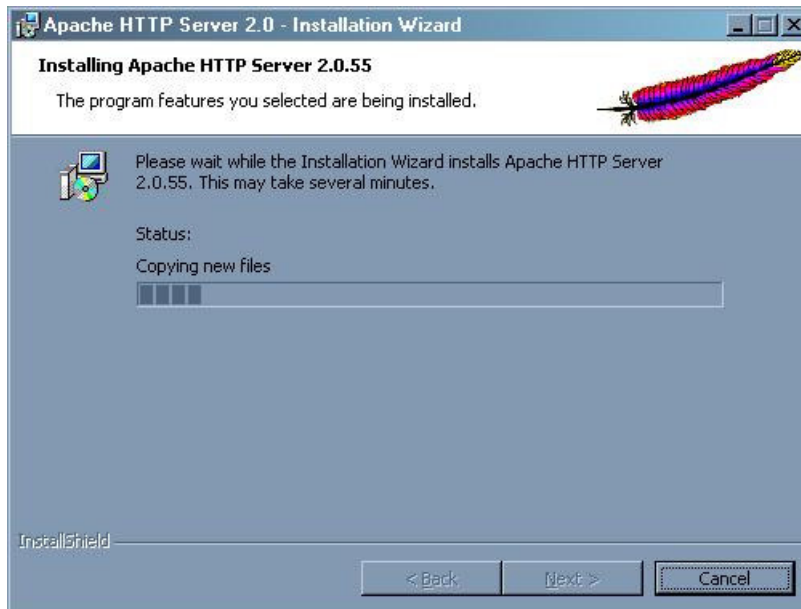
Seleccionamos el tipo de instalación y pulsamos sobre **"Next"**.



En esta ventana se selecciona el directorio de instalación, en nuestro caso el directorio por defecto. Pulsamos sobre **"Next"**.

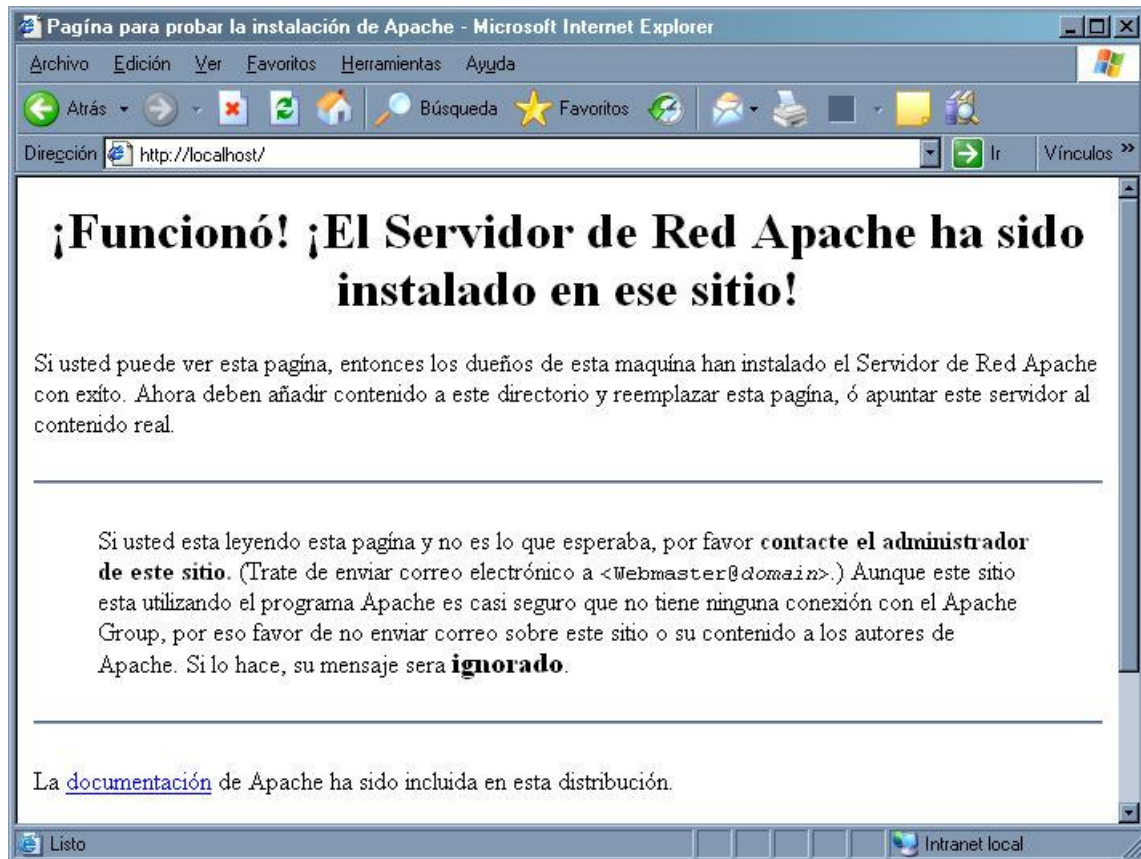


Confirmamos la instalación y pulsamos sobre **"Install"**.



Para finalizar la instalación pulsamos sobre **"Finish"**.

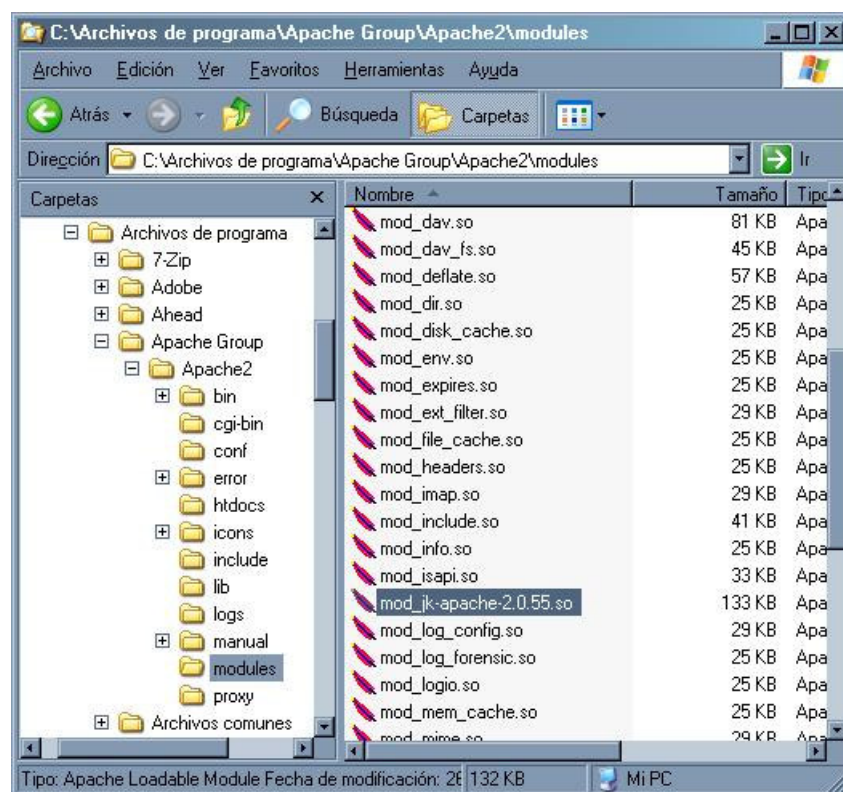
Una vez instalado el servidor web nos aseguramos que se encuentra escuchando poniendo esto en nuestro navegador: <http://localhost/>. Si todo ha ido correctamente, debemos visualizar la siguiente página:



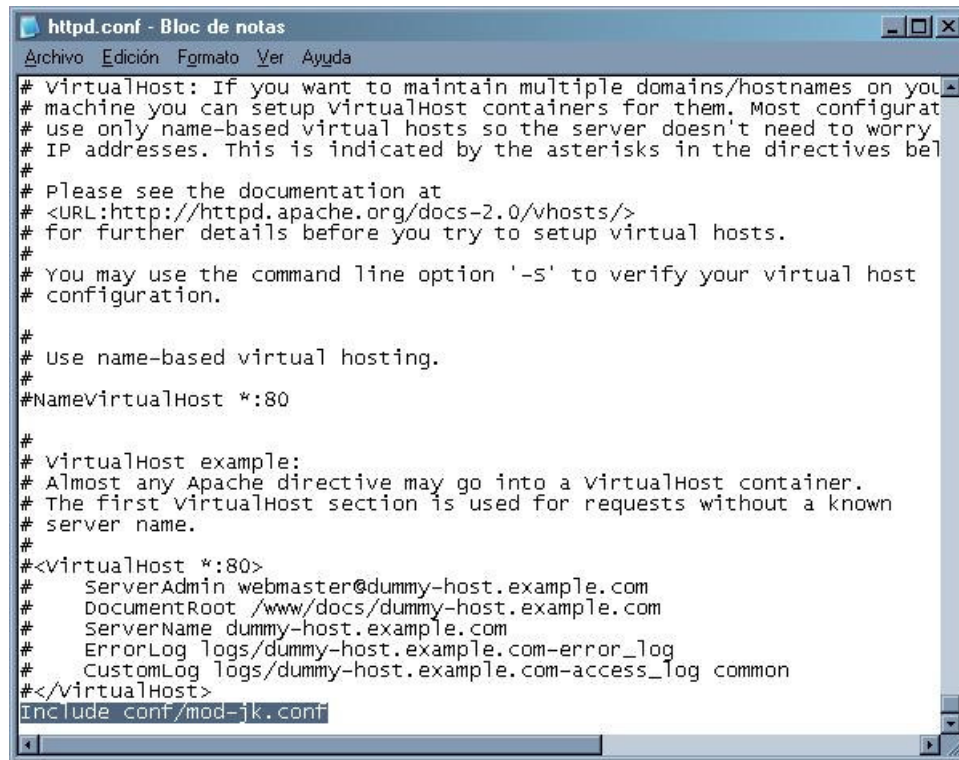
2. Instalación del conector mod_jk:

Una vez descargado el archivo "**mod_jk-apache-2.0.55.so**" debemos seguir los siguientes pasos:

1. Copiamos el archivo "**mod_jk-apache-2.0.55.so**" a "**C:\Archivos de programa\Apache Group\Apache2\modules**" :



2. Modificamos el archivo "**C:\Archivos de programa\Apache Group\Apache2\conf\httpd.conf**" con el contenido seleccionado:

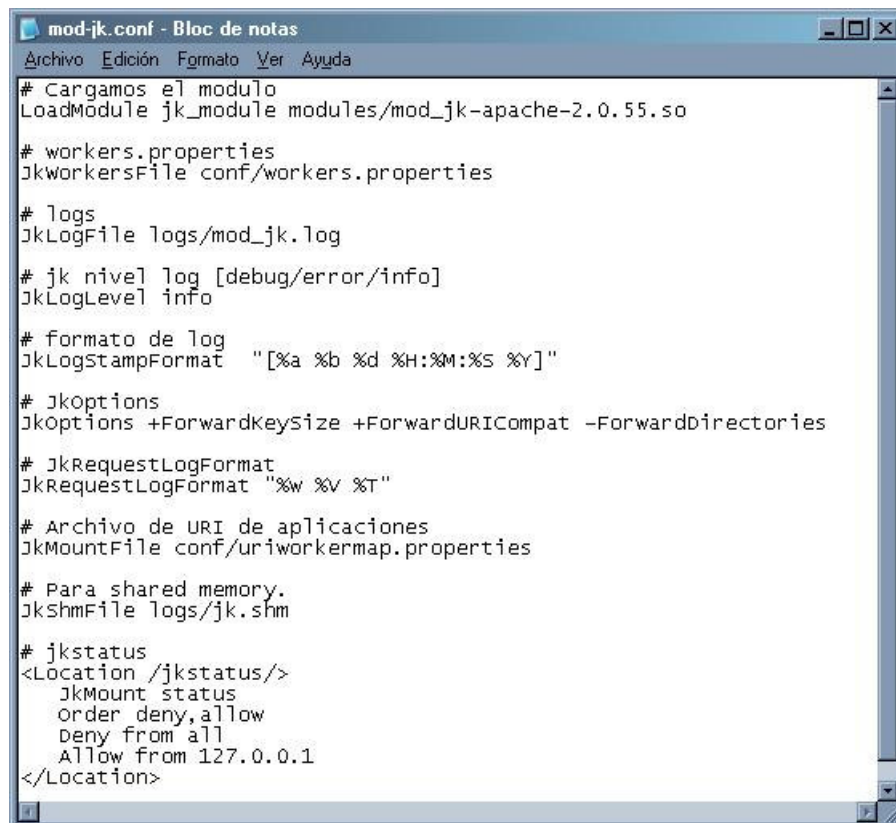


```

# VirtualHost: If you want to maintain multiple domains/hostnames on your
# machine you can setup VirtualHost containers for them. Most configurations
# use only name-based virtual hosts so the server doesn't need to worry about
# IP addresses. This is indicated by the asterisks in the directives below.
#
# Please see the documentation at
# <URL:http://httpd.apache.org/docs-2.0/vhosts/>
# for further details before you try to setup virtual hosts.
#
# You may use the command line option '-S' to verify your virtual host
# configuration.
#
# Use name-based virtual hosting.
#
#NameVirtualHost *:80
#
# VirtualHost example:
# Almost any Apache directive may go into a VirtualHost container.
# The first VirtualHost section is used for requests without a known
# server name.
#
#<VirtualHost *:80>
#    ServerAdmin webmaster@dummy-host.example.com
#    DocumentRoot /www/docs/dummy-host.example.com
#    ServerName dummy-host.example.com
#    ErrorLog logs/dummy-host.example.com-error_log
#    CustomLog logs/dummy-host.example.com-access_log common
#</VirtualHost>
include conf/mod-jk.conf

```

3. Creamos el archivo "**C:\Archivos de programa\Apache Group\Apache2\conf\mod-jk.conf**" con el siguiente contenido:



```

# Cargamos el modulo
LoadModule jk_module modules/mod_jk-apache-2.0.55.so

# workers.properties
JkWorkersFile conf/workers.properties

# logs
JkLogFile logs/mod_jk.log

# jk nivel log [debug/error/info]
JkLogLevel info

# formato de log
JkLogStampFormat "[%a %b %d %H:%M:%S %Y]"

# Jkoptions
JkOptions +ForwardKeySize +ForwardURISCompat -ForwardDirectories

# JkRequestLogFormat
JkRequestLogFormat "%w %V %T"

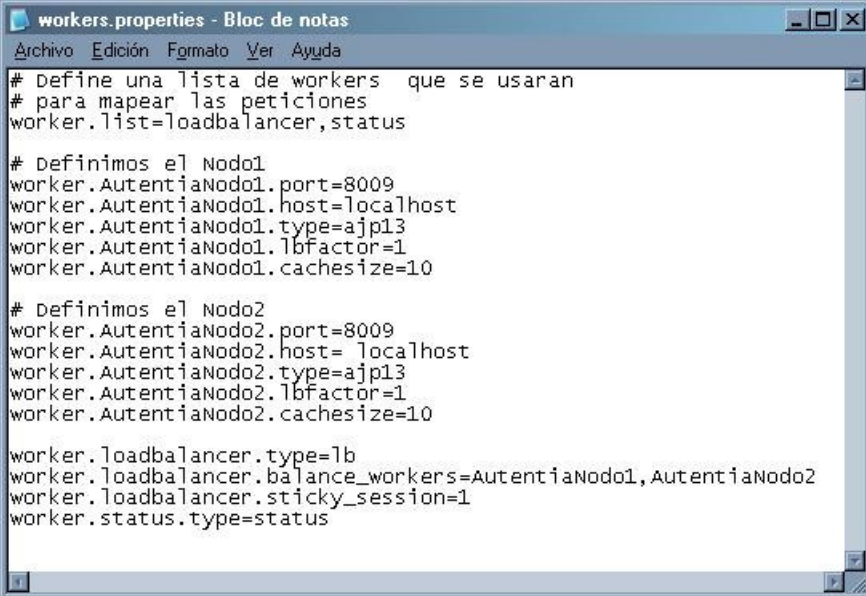
# Archivo de URI de aplicaciones
JkMountFile conf/uriworkermap.properties

# Para shared memory.
JkShmFile logs/jk.shm

# jkstatus
<Location /jkstatus/>
    JkMount status
    order deny,allow
    deny from all
    Allow from 127.0.0.1
</Location>

```

4. Creamos el archivo "**C:\Archivos de programa\Apache Group\Apache2\conf\workers.properties**" con el siguiente contenido, en el cual definimos los worker que se comunicarán con Jboss:



```

workers.properties - Bloc de notas
Archivo Edición Formato Ver Ayuda

# Define una lista de workers que se usaran
# para mapear las peticiones
worker.list=loadbalancer,status

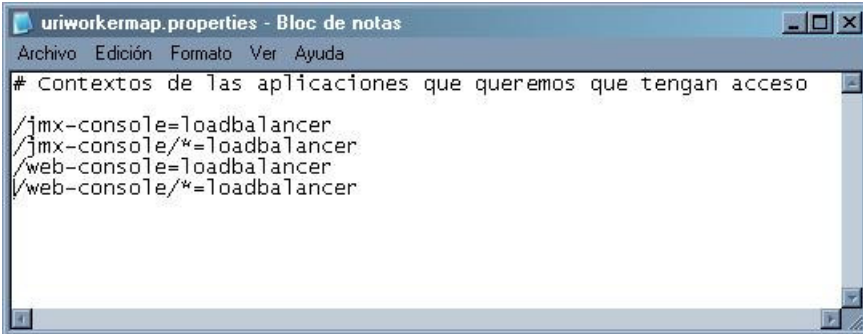
# Definimos el Nodo1
worker.AutentiaNodo1.port=8009
worker.AutentiaNodo1.host=localhost
worker.AutentiaNodo1.type=ajp13
worker.AutentiaNodo1.lbfactor=1
worker.AutentiaNodo1.cachesize=10

# Definimos el Nodo2
worker.AutentiaNodo2.port=8009
worker.AutentiaNodo2.host=localhost
worker.AutentiaNodo2.type=ajp13
worker.AutentiaNodo2.lbfactor=1
worker.AutentiaNodo2.cachesize=10

worker.loadbalancer.type=lb
worker.loadbalancer.balance_workers=AutentiaNodo1,AutentiaNodo2
worker.loadbalancer.sticky_session=1
worker.status.type=status

```

5. Creamos el archivo "**C:\Archivos de programa\Apache Group\Apache2\conf\uriworkmap.properties**" con el siguiente contenido, mediante el cual asociamos los worker a cada uno de los context-root de nuestras aplicaciones y sus recursos:



```

uriworkmap.properties - Bloc de notas
Archivo Edición Formato Ver Ayuda

# Contextos de las aplicaciones que queremos que tengan acceso
/jmx-console=loadbalancer
/jmx-console/*=loadbalancer
/web-console=loadbalancer
/web-console/*=loadbalancer

```

Una vez modificados los archivos, reiniciamos el servidor Web.

3. Instalación del servidor de aplicaciones JBoss:

Una vez descargado y descomprimido en el directorio raíz (**C:\jboss-4.0.3SP1**):

1. Ahora editamos el archivo "**C:\jboss-4.0.3SP1\bin\run.conf**" y descomentamos la línea que contiene la variable **JAVA_HOME** y la establecemos en nuestro directorio de instalación de java. En nuestro caso será:
JAVA_HOME="C:\j2sdk1.4.2_10"
2. Editamos el fichero "**C:\jboss-4.0.3SP1\server\default\deploy\jbossweb-tomcat55.sar\server.xml**" y añadimos el nombre de nuestro worker como se muestra en la imagen:

```

emptySessionPath="true" enableLookups="false" redirectPort="8443"
protocol="AJP/1.3"/>

<!-- SSL/TLS Connector configuration using the admin dev1 guide keystore
<Connector port="8443" address="${jboss.bind.address}"
    maxThreads="100" strategy="ms" maxHttpHeaderSize="8192"
    emptySessionPath="true"
    scheme="https" secure="true" clientAuth="false"
    keystoreFile="${jboss.server.home.dir}/conf/chap8.keystore"
    keystorePass="rmi+ssl" sslProtocol = "TLS" />
-->

<Engine name="jboss.web" defaultHost="localhost" jvmRoute="AutentiaNodo1">

    <!-- The JAAS based authentication and authorization realm implementati
on
    that is compatible with the jboss 3.2.x realm implementation.
    - certificatePrincipal : the class name of the
    org.jboss.security.auth.certs.CertificatePrincipal impl
    used for mapping X509[] cert chains to a Principal.
    -->
    <Realm className="org.jboss.web.tomcat.security.JBossSecurityMgrRealm"
        certificatePrincipal="org.jboss.security.auth.certs.SubjectDNMapping
..
-- SELECCIONAR --
15 37,80 22%

```

3. Editamos el archivo "C:\jboss-4.0.3SP1\server\default\deploy\jbossweb-tomcat55.sar\META-INF\jboss-service.xml" y activamos el atributo UseJK como se muestra en la imagen:

```

<!--
Whether to use MOD_JK(2) for load balancing with sticky session
combined with JvmRoute. If set to true, it will insert a JvmRouteFilter
to intercept every request and replace the JvmRoute if it detects a
failover. In addition, you will need to set the JvmRoute inside
Tomcat, e.g.,
    Engine name="jboss.web" jvmRoute="Node1" defaultHost="localhost"
in server.xml.

For clustering purpose only.
-->
<attribute name="UseJK">true</attribute>

<attribute name="Domain">jboss.web</attribute>

<!-- A mapping to the server security manager service which must be
operation compatible with type
org.jboss.security.plugins.JaasSecurityManagerServiceMBean. This is only
needed if web applications are allowed to flush the security manager
authentication cache when the web sessions invalidate.
-->
<depends optional-attribute-name="SecurityManagerService"
    proxy-type="attribute">jboss.security:service=JaasSecurityManager
</depends>
-- SELECCIONAR --
40 112,47 86%

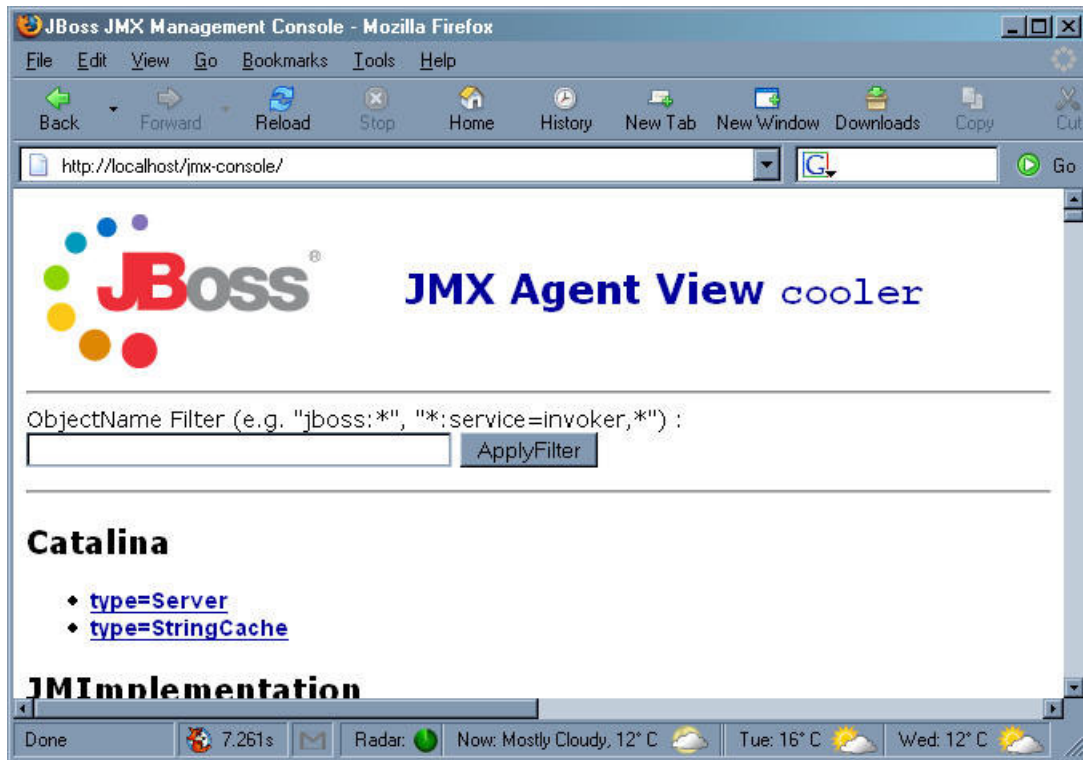
```

4. Una vez finalizada la instalación del servidor de aplicaciones realizamos la prueba de ejecución iniciando "C:\jboss-4.0.3SP1\bin\run.bat".

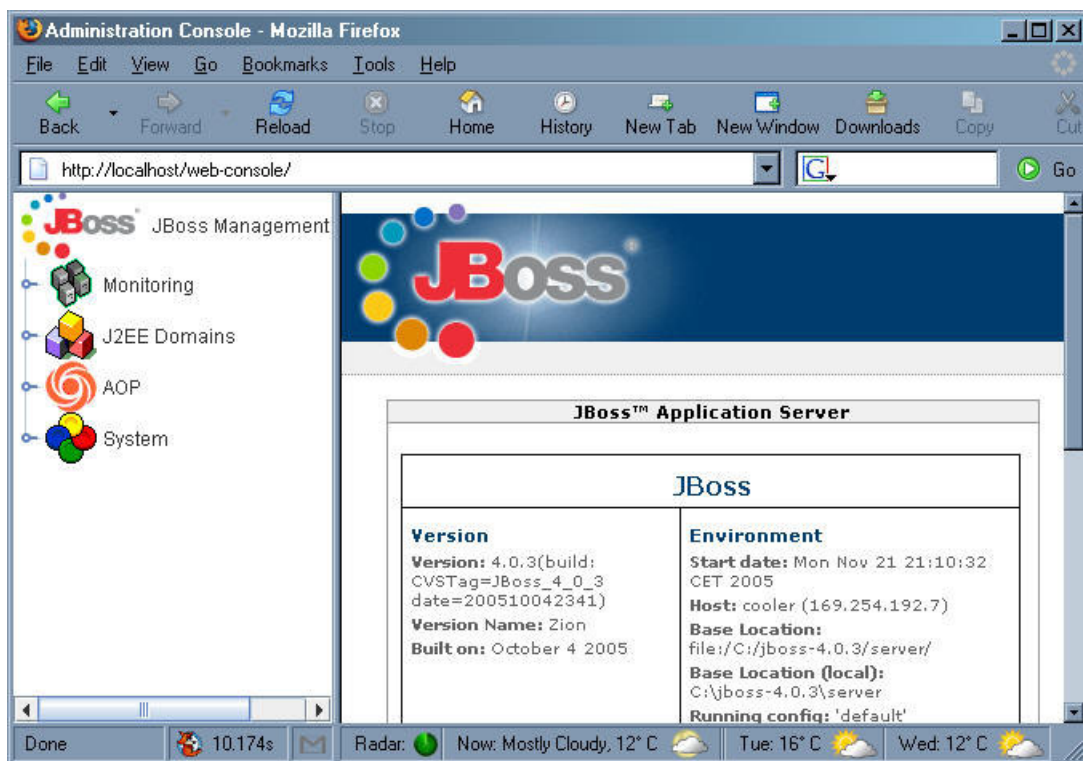
NOTA: para parar el servidor nos basta con pulsar **CTRL+C**

4. Prueba de la instalación:

<http://localhost/jmx-console/>



<http://localhost/web-console/>



Conclusión:

Mediante esta técnica, podemos añadir aplicaciones web con tan solo insertar una entrada en el archivo "**uriworkermmap.properties**" con el **context-root** de nuestras aplicaciones y los recursos a solicitar.

Recordad que este ejemplo es muy básico y que para configuraciones más especializadas podéis contactar con nosotros en [Autentia](#).

Enlaces:

[mod_jk-1.2.15](#)

[JBoss 4.0.3 RC1](#)

[j2se-1.4.2](#)[Apache 2.0.55](#)

Si desea contratar formación, consultoría 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="text"/>
	<input type="button" value="Enviar"/>

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

Nombre Corto

[Introducción al Java Web Start](#)

[Pruebas Web con JWebUnit](#)

[Generador automático de Webservices](#)

[Creación de ejemplo en PHP](#)

[Rendimiento de aplicaciones Web](#)

[Creación de Webs con Power WebSite Builder](#)

[Activación de la seguridad en Apache](#)

[Instalar un Web en tu máquina Windows](#)

[Instalar JBoss](#)

[Construir un Servidor Web en Java](#)

Descripción

Os mostramos como podéis distribuir aplicaciones de consola de un modo sencillo a través de Java Web Start. Es el mismo principio de los applets aplicado a aplicaciones.

Os mostramos como automatizar las pruebas de caja negra (desde el punto de vista de usuario final) de vuestro Web con el Framework gratuito JWebUnit. Esta técnica es perfecta para crear test de regresión de aplicaciones Web complejas.

Os mostramos como crear un servicio Web a partir de una clases, gracias a generadores automáticos de código y NetBeans

Os enseñamos a construir en PHP el ejemplo de como mostrar la frase del día

En este tutorial veremos una introducción al funcionamiento de la Suite e-Test de Empirix.

Creación de páginas web sin necesidad de conocimientos HTML, usando la herramienta Power Website Builder

Alejandro Pérez nos enseña como securizar Apache a través de autenticación básica y certificados de seguridad SSL.

Si trabajais habitualmente en plataforma Windows, en este tutorial podreis encontrar todos los pasos necesarios para instalar un servidor Web en vuestra propia máquina. Esto puede ser el primer paso para crear una Intranet.

Os mostramos como instalar en servidor gratuito de aplicaciones JBOSS así como a automatizar su arranque y parada.

En este tutorial os enseñamos los principios de las aplicaciones multi-hilo a través de la creación de un servidor web básico en Java. Podremos ver en un ejemplo real el uso de sockets, threads, excepciones, etc.

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](#)

