

## Cómo instalar Apache + Tomcat 4.x

Puesto que con la versión 4 Tomcat ha cambiado su diseño por completo, en estas páginas podreis encontrar una guía de instalación de tomcat 4, su configuración para que trabaje con Apache. todo para Linux y para Windows (si alguien se anima para hacer el de Mac...).

Si lo que quereis hacer es añadir una nueva aplicación a Tomcat, en esta misma web teneis la guía para hacerlo. Aunque esta basada en la version 3.x de Tomcat, es totalmente aplicable, y solo hay que cambiar la sección de *Informando a Apache*, añadiendo las líneas de nuestra aplicación que se mencionam al final de esta guía (tanto en Windows como en Linux).

Para cualquier duda o sugerencia no dudes en ponerte en contacto con [nosotros](#).

## Cómo instalar Apache + Tomcat 4.x en Linux

Revisión 1.1

1. [Instalando el servidor web Apache](#)
2. [Instalando Tomcat](#)
3. [Haciendo que cooperen Apache y Tomcat](#)
4. [Arrancando y reinicando el servidor](#)

**Nota importante:** al escribir la primera versión de artículo se me olvido mencionar que el módulo de conexión que se usaba en la versión 3.x de Tomcat, *mod\_jk* sigue funcionando con Tomcat 4.x.

Debido a que lleva más tiempo funcionando, es posible que ese módulo de menos problemas que *mod\_webapp*, puesto que este último esta en sus primeras versiones. Puedes encontrar información sobre como configurar el sistema con *mod\_jk* en el siguiente [artículo](#)

### Instalando el servidor web Apache

En primer lugar descargaremos el código fuente de Apache desde su web, que está en <http://httpd.apache.org> (en nuestro caso usaremos la versión 1.3.22 que se encuentra disponible para su descarga en [apache\\_1.3.22.tar.gz](http://httpd.apache.org/dist/1.3.22/apache_1.3.22.tar.gz)).

Una vez tengamos el fichero, lo descomprimiremos e iremos al directorio donde se descomprima con:

```
$tar xvzf apache_1.3.22.tar.gz
$cd apache_1.3.22
```

Para que luego podamos usar el Tomcat como un módulo de Apache, configuraremos la instalación para que se incluya el soporte para DSO. Para ello tenemos que incluir la opción **--enable-module=so** cuando ejecutemos **configure**, además de otros valores normales como el directorio de instalación (**--prefix=...**). Nuestro directorio de instalación será **/usr/local/apache** (luego referido como **APACHE\_HOME**, por lo que cuando lo utilizemos

más adelante tendrás que sustituirlo si no has creado una variable de entorno al respecto), por lo que recuerde cambiarlo cuando aparezca a continuación si decide instalarlo en otro sitio:

```
$. ./configure --prefix=/usr/local/apache --enable-module=so --enable-rule=SHARED_CORE
$make
$make install
```

En estos momentos deberíamos tener un fichero llamado **/usr/local/apache/conf/httpd.conf**. Si no es así, copie o renombre de esta manera el fichero **/usr/local/apache/conf/httpd.conf-dist**. Ahora, edite el fichero y compruebe que contiene un par de líneas como las siguientes:

```
User nobody
Group nobody
```

Estos serán el grupo y el usuario con los que se ejecutará el servidor, por lo que recuerde que el usuario nobody y el grupo nobody tendrán que ser los propietarios de los directorios de nuestras páginas para que el servidor tenga permiso para acceder a ellas.

Si quiere lanzar el servidor ejecute:

```
/usr/local/apache/bin/apachectl start
```

Recuerde que si quiere que se ejecute el servidor al reiniciar el sistema tendrá que incluir la línea anterior en sus ficheros de inicio.

## Instalando Tomcat

Vamos a descargar antes que nada el código de Tomcat que se encuentra disponible en <http://jakarta.apache.org>. En este caso descargaremos los binarios, ya que al tratarse de un programa que está escrito en Java los podremos usar en cualquier plataforma con una JVM. La última versión estable (4.0.2) que es la que usaremos la podéis descargar directamente desde <http://jakarta.apache.org/builds/jakarta-tomcat-4.0/release/v4.0.1/bin/jakarta-tomcat-4.0.1.tar.gz>.

Ahora instalaremos Tomcat en **/usr/local/jakarta-tomcat-4.0.1**.

```
$cp jakarta-tomcat-4.0.1.tar.gz /usr/local
$tar xvzf /usr/local/jakarta-tomcat-4.0.1.tar.gz
$rm /usr/local/jakarta-tomcat-4.0.1.tar.gz
```

Una vez hecho esto, sólo tenemos que poner una variable llamada CATALINA\_HOME en nuestro entorno y que apunte al directorio anterior, lo que en bash (Bourne Again SHell) se hace así:

```
$CATALINA_HOME=/usr/local/jakarta-tomcat-4.0.1; export CATALINA_HOME
```

Recuerde que antes de lanzar el Tomcat debe de tener también una variable de entorno llamada JAVA\_HOME que apunte al directorio raíz de la instalación de su JVM, y que además el directorio JAVA\_HOME/bin debe estar incluido en la variable de entorno PATH.

Ahora ya podemos lanzar nuestro servidor Tomcat con el script:

```
/usr/local/jakarta-tomcat-4.0.1/bin/startup.sh
```

y pararlo con

```
/usr/local/jakarta-tomcat-4.0.1/bin/shutdown.sh
```

## Haciendo que cooperen Apache y Tomcat 4

Este es un de los aspectos que más han cambiado con la llegada de Tomcat 4, así que si usas otra versión, la 3.x no te servirá esta explicación, aunque tienes en esta misma web una explicación de como hacer lo mismo con versiones anteriores.

Se podría decir que ahora la configuración es más sencilla, solo hay que configurar una librería que contiene el programita que hace de puente entre ambos, el *famoso* mod\_webapp.

El primer paso, como siempre, es descargarse esta librería [webapp-module-1.0-tc40-linux-glibc2.2.tar.gz](http://webapp-module-1.0-tc40-linux-glibc2.2.tar.gz). Lo siguiente es descomprimirlo y copiarlo al directorio correspondiente de Apache:

```
$tar xvzf webapp-module-1.0-tc40-linux-glibc2.2.tar.gz
$cd webapp-module-1.0-tc40
$cp mod_webapp.so APACHE_HOME/libexec
```

Una vez copiados este archivo tenemos que retocar la configuración de Apache para que sepa que peticiones son para Tomcat. Para esto usamos nuestro editor favorito para cambiar el fichero

```
APACHE_HOME/conf/httpd.conf
```

buscamos la sección *LoadModule* y añadimos al final las siguientes líneas:

```
LoadModule webapp_module libexec/mod_webapp.so
AddModule mod_webapp.c
```

Sé que unas líneas más arriba hay una sección para los *AddModule*, pero si ponemos esa línea antes de cargar el módulo no nos arrancará Apache, así que mejor dejarlas en ese orden.

Y finalmente tenemos que *instalar* o informar a Apache de cuales son las peticiones a pasarle a Tomcat, en este caso añadiremos la aplicación con ejemplos que viene con Tomcat. Para ello añadimos al final del mismo fichero *httpd.conf*

```
WebAppConnection conexion warp localhost:8008
WebAppDeploy examples conexion /examples
```

La primera línea crea una conexión entre Apache y Tomcat a través de nuestro módulo WebApp. Los parámetros son fáciles, el primero (*conexion*) es el nombre que queremos dar a esa conexión. El segundo (*warp*) es el encargado de indicar el tipo de conexión, pero puesto que por ahora solo existe warp, habrá que ponerlo así. Finalmente tenemos que indicar el servidor donde esta Tomcat (*localhost* en mi caso) y el puerto por el que conectar (*8008*). No confundais este puerto con el puerto en el que escucha el servidor web que trae Tomcat (normalmente el 8080, aunque configurable). Este puerto es el que usará nuestro modulo de conexión, y su valor por defecto es el 8008, aunque podeis cambiarlo o consultarlo en el fichero CATALINA\_HOME/conf/server.xml, en la sección:

```
<Service name="Tomcat-Apache">
  <Connector className="org.apache.catalina.connector.warp.WarpConnector"
    port="8008" minProcessors="5" maxProcessors="75"
    enableLookups="true"
    acceptCount="10" debug="0"/>
  . . .
```

La segunda línea de las añadidas anteriormente es la que informa a Apache de cada una de nuestras aplicaciones en Tomcat, y habrá que añadir una por cada aplicación que tengamos, también cuando desarrollemos una nueva. El primer parametro (*examples*) será el nombre que de la URL de la aplicación (*http://localhost/examples/*) y que deberá ser igual al nombre de la aplicación en Tomcat, el segundo (*conexion*) es el nombre de la conexión creada en la línea anterior (si queremos usar más tendremos que jugar con la configuración de CATALINA\_HOME/conf/server.xml), y por último introducimos el nombre de la aplicación en Tomcat (*/examples*), que es el nombre del directorio dentro de CATALINA\_HOME/webapps/ donde hemos colocado la aplicación. Si queremos trabajar con servidores virtuales, tendremos que añadir esta línea a cada servidor virtual que deba servir la aplicación.

Por último, y de forma opcional, si queremos podemos añadir una URL para una aplicación que nos mostrará la configuración de el modulo de comunicación.

```
WebAppInfo /webapp-info
```

Dirigiendonos a la URL <http://host/webapp-info> podremos ver dicha información.

## Arrancando y reiniciando el servidor

Lo último que queda por decir para completar esta guía de instalación, es que debido a la naturaleza del modulo de conexión entre Apache y Tomcat, la secuencia de arranque y reinicio de ambos servidores tiene que ser en el mismo orden.

Deberemos arrancar primero Tomcat y después Apache.

Para reiniciar, primero detendremos Apache, después Tomcat, y arrancaremos como antes, primero Tomcat y después Apache.

Debido a esto, para desarrollo, os recomiendo que useis directamente Tomcat como servidor web, en el puerto 8080, en el 80 o en que queráis, y solo useis la combinación de ambos cuando lleveis vuestra aplicación a producción.

## Cómo instalar Apache + Tomcat 4.x en Windows

Revisión 1.1

1. [Instalando el servidor web Apache](#)
2. [Instalando Tomcat](#)
3. [Haciendo que cooperen Apache y Tomcat](#)
4. [Arrancando y reinicando el servidor](#)

**Nota importante:** al escribir la primera versión de artículo se me olvidó mencionar que el módulo de conexión que se usaba en la versión 3.x de Tomcat, *mod\_jk* sigue funcionando con Tomcat 4.x.

Debido a que lleva más tiempo funcionando, es posible que ese módulo de menos problemas que *mod\_webapp*, puesto que este último está en sus primeras versiones. Puedes encontrar información sobre cómo configurar el sistema con *mod\_jk* en el siguiente [artículo](#)

### Instalando el servidor web Apache

Para Windows podemos descargar directamente la última versión de Apache en formato .msi (En estos momentos está la versión 1.3.22 en [apache\\_1.3.22-win32-x86.msi](#), véisite <http://httpd.apache.org> para ver cuál es la versión más reciente). Para que Windows reconozca estos ficheros deberá tener instalado el Windows Installer. El programa de instalación le guiará a continuación.

También podemos descargar un fichero exe autoejecutable, pero son 3 MB más de descarga. Podéis bajarlo aquí [apache\\_1.3.22-win32-x86.exe](#)

Elegido el formato que deseéis, como siempre en Windows, doble click y que él haga lo que quiera ;-). Lo único que debéis hacer es decir dónde queréis instalarlo (ese directorio lo tenéis que sustituir cuando escribamos APACHE\_HOME), el nombre que queráis dar a vuestro servidor, y poco más.

### Instalando Tomcat

Vamos a descargar antes que nada el código de Tomcat 4 que se encuentra disponible en <http://jakarta.apache.org>. En este caso descargaremos los binarios, ya que al tratarse de un programa que está escrito en Java los podremos usar en cualquier plataforma con una JVM. La última versión estable (4.0.1) que es la que usaremos la podéis descargar directamente desde [jakarta-tomcat-4.0.1.tar.gz](#).

También podeis descargar Tomcat en otros formatos, como fichero ejecutable .exe o con formato winzip .zip, pero puesto que los ficheros .tar.gz son los que menos ocupan y pueden ser descomprimidos por WinZip y casi cualquier otra herramienta de compresión, yo son los que os recomiendo. El fichero .exe hace algunas cosas más, como enlaces directos y configuración, e incluso instalarlo como servicio si tenemos Windows NT/2000 , pero bueno, si nos lo hacen todo no sabremos donde buscar cuando halla un problema.

Ahora instalaremos Tomcat desde un fichero comprimido, para aprender las cosas manuales, utilizando WinZip (o similar) para descomprimir el fichero descargado anteriormente. Vamos a suponer que lo descomprimimos en C:\

Ahora tenemos que poner una variable llamada CATALINA\_HOME en nuestro entorno y que apunte al directorio anterior. En Windows 95/98 añadiremos la línea siguiente al fichero **autoexec.bat**:

```
SET CATALINA_HOME=c:\jakarta-tomcat-4.0.1
```

En Windows NT/2000 tendrá que hacerlo desde el diálogo Propiedades del Sistema (para abrirlo botón derecho del ratón sobre MiPC y Propiedades), en la sección Avanzado.

Recuerde que antes de lanzar el Tomcat debe de tener también una variable de entorno llamada JAVA\_HOME que apunte al directorio raíz de la instalación de su JVM, y que además el directorio JAVA\_HOME/bin debe estar incluido en la variable de entorno PATH. Si no lo tenemos así configurado no funcionará el arranque manual.

Ahora ya podemos lanzar nuestro servidor Tomcat con el script:

```
c:\jakarta-tomcat-4.0.1\bin\startup.bat
```

y pararlo con:

```
c:\jakarta-tomcat-4.0.1\bin\shutdown.bat
```

Si hemos sido vagos y hemos instalado Tomcat desde el fichero ejecutable, pues podemos ejecutarlo desde el propio menu de inicio con los accesos directos adecuados, aunque si algún día necesitamos arrancarlo manualmente tendremos que configurar todo como se ha dicho anteriormente. De la misma forma, en Windows NT/2000 podremos ejecutarlo y detenerlo como cualquier otro servicio.

**Importante:** por alguna razón, al menos en mi caso, si arranco Tomcat por medio del menú de inicio, Apache y Tomcat no se acaban de entender, medio colgando el equipo, así que os aconsejo arrancar Tomcat manualmente, y si alguien conoce la razón de este problema que me lo haga saber.

## Haciendo que cooperen Apache y Tomcat 4

Este es un de los aspectos que más han cambiado con la llegada de Tomcat 4, así que si usas otra versión, la 3.x no te valdrá, aunque tienes en esta misma web una explicación de como hacer lo mismo con versiones anteriores.

Se podría decir que ahora la configuración es más sencilla, solo hay que configurar una librería que contiene el programita que hace de puente entre ambos, el *famoso* `mod_webapp`.

El primer paso, como siempre, es descargarse esta librería [webapp-module-1.0-rc40-windows.zip](http://webapp-module-1.0-rc40-windows.zip). Al descomprimir este fichero vemos que contiene, a parte de los ficheros habituales de licencia y demás, dos ficheros:

```
libapr.dll
mod_webapp.so
```

Lo que tenemos que hacer es copiar estos dos ficheros al directorio:

```
APACHE_HOME\modules
```

Una vez copiados estos archivos tenemos que retocar la configuración de Apache para que sepa que las peticiones son para Tomcat. Para esto editamos el fichero

```
APACHE_HOME\conf\httpd.conf
```

buscamos la sección *LoadModule* y añadimos al final las siguientes líneas:

```
LoadModule webapp_module modules/mod_webapp.so
AddModule mod_webapp.c
```

Sé que unas líneas más arriba hay una sección para los *AddModule*, pero si ponemos esa línea antes de cargar el módulo no nos arrancará Apache, así que mejor dejarlas en ese orden.

Y finalmente tenemos que *instalar* o informar a Apache de cuáles son las peticiones a pasarle a Tomcat, en este caso añadiremos la aplicación con ejemplos que viene con Tomcat. Para ello añadimos al final del mismo fichero *httpd.conf*

```
WebAppConnection conexion warp localhost:8008
WebAppDeploy examples conexion /examples
```

La primera línea crea una conexión entre Apache y Tomcat a través de nuestro módulo `WebApp`. Los parámetros son fáciles, el primero (*conexion*) es el nombre que queremos dar a esa conexión. El segundo (*warp*) es el encargado de indicar el tipo de conexión, pero puesto que por ahora solo existe `warp`, habrá que ponerlo así. Finalmente tenemos que indicar el servidor donde está Tomcat (*localhost* en mi caso) y el puerto por el que conectar (*8008*). No confundáis este puerto con el puerto en el que escucha el servidor web que trae Tomcat (normalmente el 8080, aunque configurable). Este puerto es el que usará nuestro módulo de conexión, y su valor por defecto es el 8008, aunque podéis cambiarlo o consultarlo en el fichero `CATALINA_HOME\conf\server.xml`, en la sección:

```
<Service name="Tomcat-Apache">
  <Connector className="org.apache.catalina.connector.warp.WarpConnector"
    port="8008" minProcessors="5" maxProcessors="75"
    enableLookups="true"
    acceptCount="10" debug="0"/>
```

. . .

La segunda línea de las añadidas anteriormente es la que informa a Apache de cada una de nuestras aplicaciones en Tomcat, y habrá que añadir una por cada aplicación que tengamos, también cuando desarrollemos una nueva. El primer parametro (*examples*) será el nombre que de la URL de la aplicación (*http://localhost/examples/*) y que deberá ser igual al nombre de la aplicación en Tomcat, el segundo (*conexion*) es el nombre de la conexión creada en la línea anterior (si queremos usar más tendremos que jugar con la configuración de `CATALINA_HOME\conf\server.xml`), y por último introducimos el nombre de la aplicación en Tomcat (*/examples*), que es el nombre del directorio dentro de `CATALINA_HOME\webapps\` donde hemos colocado la aplicación. Si queremos trabajar con servidores virtuales, tendremos que añadir esta línea a cada servidor virtual que deba servir la aplicación.

Por último, y de forma opcional, si queremos podemos añadir una URL para una aplicación que nos mostrará la configuración de el modulo de comunicación.

```
WebAppInfo /webapp-info
```

Dirigiendonos a la URL <http://host/webapp-info> podremos ver dicha información.

## Arrancando y reiniciando el servidor

Lo último que queda por decir para completar esta guía de instalación, es que debido a la naturaleza del modulo de conexión entre Apache y Tomcat, la secuencia de arranque y reinicio de ambos servidores tiene que ser en el mismo orden.

Deberemos arrancar primero Tomcat y después Apache.

Para reiniciar, primero detendremos Apache, después Tomcat, y arrancaremos como antes, primero Tomcat y después Apache.

Debido a esto, para desarrollo, os recomiendo que useis directamente Tomcat como servidor web, en el puerto 8080, en el 80 o en que queráis, y solo useis la combianción de ambos cuando lleveis vuestra aplicación a producción.

## Enlaces

- La página del servidor web Apache: <http://httpd.apache.org>
- La página de Tomcat: <http://jakarta.apache.org>
- Distribución Java para Linux: <http://www.blackdown.org>

**Alberto Molpeceres** es ahora mismo desarrollador de aplicaciones en ámbito cliente/servidor para la empresa **T-Systems - debis Systemhaus** en Munich (Alemania). Cuando no está trabajando o "metiendo caña" al resto de los integrantes de javaHispano intenta pasear con su novia, buscar la desaparecida lógica del idioma alemán o intenta olvidar la *pesadilla* que es buscar piso en Munich.  
Para cualquier duda o tirón de orejas, e-mail a: [al@javahispano.com](mailto:al@javahispano.com)