

## Cómo instalar Apache+Tomcat con soporte MySQL

En estas páginas vamos a explicar como configurar un servidor Apache, tanto en una plataforma Linux como en Windows, dotándole de soporte para Servlets y páginas JSP mediante el paquete Tomcat. Además instalaremos también el gestor de bases de datos MySQL para su uso desde nuestras páginas.

En la sección de enlaces encontrará las direcciones desde las que descargar todos los programas que serán necesarios a lo largo del tutorial.

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

## Cómo instalar Apache+Tomcat con soporte MySQL en Linux

1. [Instalando MySQL](#)
2. [Instalando el driver JDBC para MySQL](#)
3. [Instalando el servidor web Apache](#)
4. [Instalando Tomcat](#)
5. [Haciendo que cooperen Apache y Tomcat](#)
6. [Compilando nuestro propio mod\\_jk.so](#)

### Instalando MySQL

Para descargar MySQL iremos a <http://www.mysql.com>, en donde pulsaremos sobre la sección Downloads. En esta sección elegiremos la versión estable más moderna de MySQL (en estos momentos la 3.23). Descargaremos el código fuente de la última distribución (ahora la 3.23.33) en formato .tar.gz (descárguelo directamente si quiere desde <http://www.mysql.com/Downloads/MySQL-3.23/mysql-3.23.33.tar.gz>).

Descomprimiremos el código fuente (suponiendo que el archivo descargado sea el indicado anteriormente):

```
$tar xvzf mysql-3.23.33.tar.gz
$cd mysql-3.23.33
```

Ahora tenemos que crear un nuevo usuario (mysql) y un nuevo grupo (mysql) que serán los que use MySQL para ejecutarse:

```
$/usr/sbin/groupadd mysql
$/usr/sbin/useradd -g mysql mysql
```

Ya podemos compilar e instalar el gestor de bases de datos. En nuestro caso vamos a elegir **/usr/local/mysql** como directorio de instalación. Por lo tanto, si elige otro directorio para su instalación tendrá que modificar **/usr/local/mysql** siempre que lo nombremos posteriormente con el directorio donde haya instalado su versión de MySQL.

```
$/configure --prefix=/usr/local/mysql
$make
$make install
```

Ya tenemos instalada nuestro gestor de bases de datos MySQL. Ahora tendremos que instalar las tablas básicas que MySQL necesita para funcionar. Para ello ejecutaremos un script que se nos proporciona:

```
$scripts/mysql_install_db
```

Ahora tendremos que cambiar el usuario y el grupo propietarios del directorio de instalación para que se correspondan con aquellos con los que se ejecutara el demonio **mysqld**:

```
$chown -R mysql /usr/local/mysql  
$chgrp -R mysql /usr/local/mysql
```

Para ejecutar MySQL sólo tendremos que escribir:

```
$/usr/local/mysql/bin/safe_mysqld --user=mysql&
```

Si queremos que MySQL se ejecute cada vez que reiniciamos nuestro sistema Linux tendremos que añadir a uno de los ficheros de inicio (en mi caso, Red Hat Linux 7.0, la añadí al fichero **/etc/rc.d/rc.local**) la línea:

```
/bin/sh `cd /usr/local/mysql; ./bin/safe_mysqld --user=mysql&`
```

Tendremos que cambiar el password del administrador del gestor de bases de datos, ya que MySQL no pone ninguno por defecto. Para ello, una vez que tengamos MySQL ejecutándose, haremos lo siguiente:

```
$/usr/local/mysql/bin/mysqladmin -u root -p password 'mysql'
```

Una vez que ejecutemos lo anterior, nos pedirá el password, que como en este caso no estará puesto bastará con pulsar <Intro> y quedará fijado mysql como el nuevo password.

## Instalando el driver JDBC para MySQL

En la sección [Downloads](#) de la página de [MySQL](#), iremos a la sección del API para [JDBC](#). Elegimos que versión queremos descargar (nosotros usaremos la versión 1.2c, que puede descargar directamente desde <http://www.mysql.com/Downloads/Contrib/mm.mysql.jdbc-1.2c.tar.gz>. Si descarga otra versión tendrá que cambiar los números en los nombres de ficheros y directorios).

Descomprimos el fichero obtenido y nos cambiamos al directorio creado:

```
$tar xvzf mm.mysql.jdbc-1.2.tar.gz  
$cd mm.mysql.jdbc-1.2c
```

Veremos que en dicho directorio tenemos un fichero llamado **mysql\_comp.jar** que será el que usemos a no ser que nuestra JVM no soporte los ficheros jar comprimidos (tenemos el fichero **mysql\_uncomp.jar** en ese caso). Podemos dejar el fichero donde esta o copiarlo a otro directorio, ya que lo único que tenemos que hacer para poder utilizarlo es incluirlo en el CLASSPATH, y para cargarlo desde un programa Java utilizar la siguiente sentencia:

```
Class.forName("org.gjt.mm.mysql.Driver");
```

## 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 1.3.19 que se encuentra disponible para su descarga en [http://httpd.apache.org/dist/apache\\_1.3.19.tar.gz](http://httpd.apache.org/dist/apache_1.3.19.tar.gz)).

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

```
$tar xvzf apache_1.3.19.tar.gz
$cd apache_1.3.19
```

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**. Nuestro directorio de instalación será **/usr/local/apache**, por lo que recuerde cambiarlo cuando aparezca a continuación si decide instalarlo en otro sitio:

```
$. /configure --prefix=/usr/local/apache --enable-module=so
$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 (3.2.1) que es la que usaremos la podéis descargar directamente desde <http://jakarta.apache.org/builds/jakarta-tomcat/release/v3.2.1/bin/jakarta-tomcat-3.2.1.tar.gz>.

Ahora instalaremos Tomcat en **/usr/local/jakarta-tomcat-3.2.1**.

```
$cp jakarta-tomcat-3.2.1.tar.gz /usr/local
$tar xvzf /usr/local/jakarta-tomcat-3.2.1.tar.gz
$rm /usr/local/jakarta-tomcat-3.2.1.tar.gz
```

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

```
$TOMCAT_HOME=/usr/local/jakarta-tomcat-3.2.1; export TOMCAT_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-3.2.1/bin/startup.sh
```

## Haciendo que cooperen Apache y Tomcat

Para hacer que sea Apache el que sirva las páginas estáticas a Tomcat, necesitamos instalar el módulo "jk". Para ello, tenemos que descargar el fichero **mod\_jk.so** de la página de Tomcat. Una vez que tengamos este fichero lo debemos copiar en el directorio **/usr/local/apache/libexec**. Para Linux y plataforma i386 podemos descargar el fichero [http://jakarta.apache.org/builds/jakarta-tomcat/release/v3.2.1/bin/linux/i386/mod\\_jk.so](http://jakarta.apache.org/builds/jakarta-tomcat/release/v3.2.1/bin/linux/i386/mod_jk.so). Yo he probado este módulo con Red Hat Linux 7.0 y no me funcionó, por lo que si usted hace todo lo que digo en esta sección y Apache le da un error cuando intente arrancarlo vaya a la sección [Compilando nuestro propio mod\\_jk.so](#).

Este paso es opcional, pero recomendamos que configure Tomcat para que use AJPv13, ya que es un protocolo más rápido que AJPv12, y es capaz de identificar conexiones mediante HTTPS. Para añadir este protocolo tendrá que añadir el siguiente bloque a su fichero `TOMCAT_HOME/conf/server.xml`:

```
<Connector className="org.apache.tomcat.service.PoolTcpConnector">
  <Parameter name="handler"
    value="org.apache.tomcat.service.connector.Ajp13ConnectionHandler"/>
  <Parameter name="port" value="8009"/>
</Connector>
```

En el fichero `TOMCAT_HOME/conf/server.xml` verá que hay un bloque similar pero para el protocolo AJPv12. No lo borre ya que Tomcat lo usa para ser apagado. Añada el nuevo bloque a continuación.

Al ejecutar Tomcat se creará un fichero llamado **TOMCAT\_HOME/conf/mod\_jk.conf-auto** que copiaremos sobre el fichero **TOMCAT\_HOME/conf/mod\_jk.conf**:

```
$cp $TOMCAT_HOME/conf/mod_jk.conf-auto $TOMCAT_HOME/conf/mod_jk.conf
```

Editamos el fichero `mod_jk.conf` creado y cambiamos cada aparición de `Ajpv12` por `Ajpv13`.

Ahora tenemos que hacer que Apache utilice a Tomcat para servir las páginas `.jsp` y los Servlets. Para ello añadimos a nuestro fichero `/usr/local/apache/conf/httpd.conf` la siguiente línea:

```
Include TOMCAT_HOME/conf/mod_jk.conf
```

Ahora reiniciaremos Apache y Tomcat. Para ello:

```
$/usr/local/apache/bin/apachectl stop
$TOMCAT_HOME/bin/shutdown.sh
$TOMCAT_HOME/bin/startup.sh
$/usr/local/apache/bin/apachectl start
```

Ahora bastará con que intentemos acceder a una página `.jsp` o a un servlet de nuestro servidor web y Apache redirigirá la petición a Tomcat. Recuerde que sus páginas dinámicas deben estar en un subdirectorio de `TOMCAT_HOME/webapps`.

## Compilando nuestro propio `mod_jk.so`

En esta dirección se puede descargar un [mod\\_jk.so](#) que funciona con Red Hat Linux 7.0. Si no le funcionan ni este fichero ni el que se distribuye en la página web de [Tomcat](#), tendrá que compilar `mod_jk.so` para su sistema operativo Linux. Para ello necesita el código fuente de este módulo, que se puede descargar desde [http://www.javahispano.com/howto/apache-tomcat/mod\\_jk.tar.gz](http://www.javahispano.com/howto/apache-tomcat/mod_jk.tar.gz). Este fichero es un extracto del código fuente de Tomcat que contiene los fuentes del módulo para Apache 1.3. Si su versión es otra póngase en contacto con [nosotros](#).

Para compilar el módulo, descomprímalo y vaya al directorio `apache1.3`:

```
$tar xvzf mod_jk.tar.gz
$cd apache1.3
```

Compílelo con la utilidad **apxs** que incluye Apache (necesitará tener instalado el módulo de Perl5 en su sistema) de la siguiente manera:

```
$/usr/local/apache/bin/apxs -o mod_jk.so -I../jk -IJAVA_HOME/include
\ -IJAVA_HOME/include/linux -c *.c ../jk/*.c
```

Puede que la compilación falle por un problema de escritura de ficheros, por lo que para que se termine de construir el módulo tendrá que ejecutar:

```
$gcc -shared -o mod_jk.so *.o
```

Ahora ya tendrá su fichero `mod_jk.so` que sólo tendrá que copiar en el directorio `/usr/local/apache/libexec`.

## Cómo instalar Apache+Tomcat con soporte MySQL en Windows

1. [Instalando MySQL](#)

2. [Ejecutando MySQL en Windows 95/98](#)
3. [Ejecutando MySQL en Windows NT/2000](#)
4. [Instalando el driver JDBC para MySQL](#)
5. [Instalando el servidor web Apache](#)
6. [Instalando Tomcat](#)
7. [Haciendo que cooperen Apache y Tomcat](#)

## Instalando MySQL

Para descargar MySQL iremos a <http://www.mysql.com>, en donde pulsaremos sobre la sección Downloads. En esta sección elegiremos la versión estable más moderna de MySQL (en estos momentos la 3.23). Descargaremos los binarios para Windows de la última distribución (ahora la 3.23.33) en formato .zip (descárguelo directamente si quiere desde <http://www.mysql.com/Downloads/MySQL-3.23/mysql-3.23.33-win.zip>).

Descomprimos el fichero (p.e. con WinZip) y ejecutamos el fichero setup.exe. El programa de instalación nos guiará paso a paso.

Recuerde que si decide instalar MySQL en otro directorio distinto de c:\mysql tendrá que utilizar la opción --basedir=MYSQL\_HOME, considerando MYSQL\_HOME de ahora en adelante el directorio de instalación de MySQL.

## Ejecutando MySQL en Windows 95/98

Para ejecutar MySQL necesita tener instalado en su máquina el paquete Winsock que proporciona el protocolo TCP/IP. En Windows 98 lo podrá instalar desde su CD-ROM, pero con Windows 95 lo tendrá que descargar desde <http://www.microsoft.com>. De todas formas si usted ya tiene conexión a Internet en su ordenador, el paquete deberá estar instalado.

Para arrancar MySQL podemos ejecutar:

```
MYSQL_HOME\bin\mysqld
```

o

```
MYSQL_HOME\bin\mysqld-opt
```

El segundo es una versión optimizada del primero para procesadores Pentium.

Para detener la ejecución:

```
MYSQL_HOME\bin\mysqladmin -u root shutdown
```

## Ejecutando MySQL en Windows NT/2000

Para ejecutar MySQL en Windows NT, necesitará tener instalado como mínimo el Service Pack 3.

En Windows NT/2000 puede ejecutar MySQL como en windows 95/98, pero se también tiene la opción más recomendable de ejecutarlo como un servicio. Para instalar MySQL como un servicio ejecute:

```
MYSQL_HOME\bin\mysqld-nt --install
```

Ahora para arrancar MySQL use:

```
net start mysql
```

Y para parar MySQL:

```
net stop mysql
```

### Instalando el driver JDBC para MySQL

En la sección [Downloads](#) de la página de [MySQL](#), iremos a la sección del API para [JDBC](#). Elegimos que versión queremos descargar (nosotros usaremos la versión 1.2c, que puede descargar directamente desde <http://www.mysql.com/Downloads/Contrib/mm.mysql.jdbc-1.2c.tar.gz>. Si descarga otra versión tendrá que cambiar los números en los nombres de ficheros y directorios).

Descomprimos el fichero obtenido con Winzip y nos cambiamos al directorio creado.

Veremos que en dicho directorio tenemos un fichero llamado **mysql\_comp.jar** que será el que usemos a no ser que nuestra JVM no soporte los ficheros jar comprimidos (tenemos el fichero **mysql\_uncomp.jar** en ese caso). Podemos dejar el fichero donde está o copiarlo a otro directorio, ya que lo único que tenemos que hacer para poder utilizarlo es incluirlo en el CLASSPATH, y para cargarlo desde un programa Java utilizar la siguiente sentencia:

```
Class.forName("org.gjt.mm.mysql.Driver");
```

### Instalando el servidor web Apache

Para Windows podemos descargar directamente la última versión de Apache en formato .msi (En estos momentos está la versión 1.3.19 en [http://httpd.apache.org/dist/binaries/win32/apache\\_1.3.19-win32-no\\_src-r2.msi](http://httpd.apache.org/dist/binaries/win32/apache_1.3.19-win32-no_src-r2.msi), visite <http://httpd.apache.org> para ver cual 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.

### 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 (3.2.1) que es la que usaremos la podéis descargar directamente

desde <http://jakarta.apache.org/builds/jakarta-tomcat/release/v3.2.1/bin/jakarta-tomcat-3.2.1.tar.gz>.

Ahora instalaremos Tomcat, utilizando WinZip (o similar) para descomprimir el fichero descargado anteriormente. Vamos a suponer que lo descomprimos en C:\

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

```
SET TOMCAT_HOME=c:\jakarta-tomcat-3.2.1
```

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.

Ahora ya podemos lanzar nuestro servidor Tomcat con el script:

```
c:\jakarta-tomcat-3.2.1\bin\startup
```

## Haciendo que cooperen Apache y Tomcat

Para hacer que sea Apache el que sirva las páginas estáticas a Tomcat, necesitamos instalar el módulo "jk". Para ello, tenemos que descargar el fichero **mod\_jk.dll** de la página de Tomcat. Una vez que tengamos este fichero lo debemos copiar en el subdirectorio **libexec** de nuestra instalación de Apache. Para Windows podemos descargar el fichero [http://jakarta.apache.org/builds/jakarta-tomcat/release/v3.2.1/bin/win32/i386/mod\\_jk.dll](http://jakarta.apache.org/builds/jakarta-tomcat/release/v3.2.1/bin/win32/i386/mod_jk.dll).

Este paso es opcional, pero recomendamos que configure Tomcat para que use Ajpv13, ya que es un protocolo más rápido que Ajpv12, y es capaz de identificar conexiones mediante HTTPS. Para añadir este protocolo tendrá que añadir el siguiente bloque a su fichero TOMCAT\_HOME/conf/server.xml:

```
<Connector className="org.apache.tomcat.service.PoolTcpConnector">
  <Parameter name="handler"
    value="org.apache.tomcat.service.connector.Ajpv13ConnectionHandler"/>
  <Parameter name="port" value="8009"/>
</Connector>
```

En el fichero TOMCAT\_HOME\conf\server.xml verá que hay un bloque similar pero para el protocolo Ajpv12. No lo borre ya que Tomcat lo usa para ser apagado. Añada el nuevo bloque a continuación.

Al ejecutar Tomcat se creará un fichero llamado TOMCAT\_HOME\conf\mod\_jk.conf-auto que copiaremos sobre el fichero TOMCAT\_HOME\conf\mod\_jk.conf:

```
copy %TOMCAT_HOME%\conf\mod_jk.conf-auto %TOMCAT_HOME%\conf\mod_jk.conf
```



Editamos el fichero mod\_jk.conf creado y cambiamos cada aparición de Ajpv12 por Ajpv13.

Ahora tenemos que hacer que Apache utilice a Tomcat para servir las páginas .jsp y los Servlets. Para ello añadimos a nuestro fichero conf/httpd.conf de nuestra instalación de Apache la siguiente línea:

```
Include TOMCAT_HOME/conf/mod_jk.conf
```

Ahora reiniciaremos Apache y Tomcat. Para ello pararemos Apache desde el menú inicio, y después ejecutaremos:

```
TOMCAT_HOME\bin\shutdown  
TOMCAT_HOME\bin\startup
```

Y arrancaremos de nuevo Apache desde el menú inicio.

Ahora bastará con que intentemos acceder a una página .jsp o a un servlet de nuestro servidor web y Apache redirigirá la petición a Tomcat. Recuerde que sus páginas dinámicas deben estar en un subdirectorio de TOMCAT\_HOME\webapps.

## Enlaces

- La página de MySQL: <http://www.mysql.com>
- 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>

**Roberto Casas** compagina sus estudios de 5º de Ingeniería Informática en el Centro Politécnico Superior de Zaragoza con su trabajo en la empresa Master-D y javaHispano. Para cualquier duda o tirón de orejas, e-mail a: [roberto@javahispano.com](mailto:roberto@javahispano.com)