



Bisoños Usuarios de GNU/Linux de Mallorca y Alrededores | Bergantells Usuaris de GNU/Linux de Mallorca i Afegitons

## Primeros pasos con Tomcat (259156 lectures)

Per Angel Luis Aparicio Gómez, [Seapot](http://seapot.webcindario.com) (<http://seapot.webcindario.com>)

Creado el 21/10/2003 16:51 modificado el 21/10/2003 16:51

Este artículo está dirigido a todos aquellos que quieran montar un servidor con soporte para Servlets de Java y páginas JSP. Se ha elegido la combinación Tomcat sobre un sistema Linux, (Mandrake en concreto)

**Corrección:** Cuando escribí el artículo, en mi ignorancia creí que era necesario tener instalado el Apache para usar Tomcat. Gracias a los comentarios recibidos me di cuenta que eso no es así, con lo que he eliminado toda referencia al Apache del artículo, para no confundir a la gente

## INTRODUCCIÓN

Hace poco tuve la necesidad de utilizar un servidor con soporte para Java. Tras decidirme por Apache Tomcat (debido a que ya había estado utilizando Apache de antemano) pude comprobar que a pesar de que en internet hay mucha información al respecto está todo un poco disperso, con lo que he decidido escribir este artículo para ponerlo todo más o menos juntito. Como siempre, si hay algún fallo, o se me pasa algo por alto, decídmelo y lo corregiré.

Los versiones usadas han sido los siguientes:

- Tomcat 5.0.12 Beta
- Linux Mandrake 9.1 (Aunque supongo que lo explicado valdrá en cualquier Linux, incluso en cualquier SO para el que esté disponible Tomcat)

El objetivo de este tutorial es explicar como montar el Tomcat, no explicar como se escribe un Servlet en Java. Si queréis información sobre esto último os recomiendo echar un vistazo a la lista de enlaces que he puesto al final.

## REQUISITOS PREVIOS

Voy a presuponer que la JSDK de java ya se encuentran instaladas en el sistema. Si no es el caso, se puede instalar de la siguiente forma (explicado muy por encima):

- **Java:** Bajamos la JSDK oficial en <http://java.sun.com/j2se/1.4.2/download.html><sup>(1)</sup>. Después añadís el directorio *bin* de la instalación de java al path del sistema.

Para ver si todo está correcto, comprobar que el shell reconoce el comando *java* .

## INSTALANDO EL TOMCAT

Lo primero que tenemos que hacer es bajarnos el Tomcat de la página oficial de Apache ( <http://jakarta.apache.org/site/binindex.cgi><sup>(2)</sup> ) en la cual deberemos buscar la última versión del Tomcat y bajarnoslo (Tened en cuenta que en dicha página vienen los descargables de todos los proyectos de Jakarta, buscad hasta encontrar la última versión de Tomcat). Si queréis compilar a partir del código fuente, la dirección es: (<http://jakarta.apache.org/site/sourceindex.cgi><sup>(3)</sup> )

Una vez descargado, copiamos el archivo al directorio donde vayamos a instalar Tomcat. Yo lo copié al directorio /var, pero podéis copiarlo donde queráis. Descomprimís el archivo y le cambiáis el nombre al directorio que se haya creado por otro más manejable, por ejemplo tomcat (no es necesario, es por comodidad). A partir de ahora llamaré a dicho



directorio \$CATALINA\_HOME, por seguir la documentación oficial de Tomcat ( CATALINA\_HOME = directorio de instalación de Jakarta-Tomcat)

Ahora tenéis que añadir la variable JAVA\_HOME al sistema, apuntando al directorio donde instalasteis el java (Para que el Tomcat sepa donde buscar el java). Después ir al directorio \$ CATALINA\_HOME/bin. En este directorio están los scripts para arrancar (*startup.sh*) y parar (*shutdown.sh*) el Tomcat. También encontrareis los ejecutables para Windows (los .exe y los .bat), que podéis borrar directamente (Si por alguna razón estáis usando Windows, pues entonces borra los .sh). Bueno, pues ejecutáis el script *startup.sh* y accedéis a <http://localhost:8080/><sup>(4)</sup> desde vuestro navegador favorito. Si todo ha ido bien, deberías ver la página principal de Tomcat (puede que el Tomcat tarde un poco en arrancar. Si inicialmente no se conecta al puerto 8080, darle a recargar en vuestro navegador y probad de nuevo)

## CREANDO UN SERVLET

Ahora vamos a crear un Servlet de ejemplo. Primero tenemos que añadir al CLASSPATH de java las bibliotecas de Servlets que nos proporciona el Tomcat. Para eso añadimos el directorio \$CATALINA\_HOME/common/lib/servlet-api.jar a la variable de entorno anteriormente mencionada (ojo, no escribáis \$CATALINA\_HOME, sustituirlo por el directorio donde tengáis instalado el Tomcat). Una vez preparado todo, pasamos a escribir nuestro ejemplo, que será el típico HolaMundo. El código concreto es:

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class HolaMundo extends HttpServlet
{
    public void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException
    {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println("
");out.println("
Prueba!");out.println("
");out.println("
¡Hola mundo!");
        out.println("
");
    }
}
```

Recomiendo que desarrolléis el ejemplo en un directorio ajeno a Tomcat, en vuestro home, o en donde tengais por costumbre hacer los desarrollos Java.

Bueno, grabáis el código en un fichero HolaMundo.java y lo compiláis y obtenéis el .class. Ahora tenéis que montar la estructura de directorios propia de un Servlet, acorde a la especificación oficial de SUN. Dicha estructura es la siguiente:

- **Nombre\_de\_la\_aplicación** (el que queráis, yo usaré prueba)
  - ◆ **META-INF** (nosotros no lo usaremos)
  - ◆ **WEB-INF** (Aquí incluimos el archivo web.xml, del que ya hablaremos)
    - ◇ **classes** (para guardar los archivos compilados)
    - ◇ **src** (opcional, para guardar las fuentes)
    - ◇ **lib** (contendrá los .jar externos. No lo usaremos)

Sería muy recomendable guardar toda esta estructura básica en un lugar aparte, para no tener que crearla cada vez que vayáis a desarrollar una aplicación nueva. Lo que tenemos que hacer a continuación es copiar nuestro archivo *HolaMundo.class* al directorio **classes** También podemos copiar *HolaMundo.java* a **src**, pero es opcional (como si queremos borrar el .java, en realidad el motor de Servlets no lo necesita para nada, aunque es recomendable guardar siempre los fuentes, como es natural). Posteriormente tenemos que crear el archivo web.xml, que dijimos antes. Este archivo es el que describe a nuestra aplicación. En nuestro caso debe ser el siguiente:

```
<?xml version="1.0" encoding="ISO-8859-15"?>
<!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.2//EN"
```



```
"http://java.sun.com/j2ee/dtds/web-app_2_2.dtd">
<web-app>
  <servlet>
    <servlet-name>prueba</servlet-name>
    <servlet-class>HolaMundo</servlet-class>
  </servlet>

  <servlet-mapping>
    <servlet-name>prueba</servlet-name>
    <url-pattern>hola</url-pattern>
  </servlet-mapping>
</web-app>
```

En *servlet-name* indicaremos el nombre de nuestra aplicación y lo haremos coincidir con el directorio donde la tenemos guardada (aquel cuyo contenido estamos creando ahora). En *servlet-class* ponemos la clase principal de nuestra aplicación y en *url-pattern* la dirección desde la cual accederemos al servlet. Para más información consultar la documentación oficial.

Bueno, copiamos nuestra aplicación al directorio `$CATALINA_HOME/webapps/` y reiniciamos el Tomcat (usando los scripts `shutdown.sh` y `startup.sh` comentados antes). Para comprobar que todo va bien accedemos a <http://localhost/prueba/hola><sup>(5)</sup> (Si habéis usado otro url-pattern u otro servlet-name la dirección general es `"http://localhost/$servlet-name/$url-pattern"`<sup>(6)</sup>). Si todo ha ido bien, deberíamos ver el consabido "¡Hola Mundo!".

Aún nos queda algo por hacer. Os habréis fijado que para cargar nuestra aplicación hemos tenido que reiniciar el servidor. Esto no es muy adecuado en la mayoría de los casos. Ahora solucionaremos dicho problema

## EL MANAGER DE TOMCAT

El manager es una aplicación para administración del motor Tomcat que usa una interfaz vía web (también se pueden usar mediante transmisiones http, pero no lo trataré en este tutorial). En principio y por razones de seguridad no podemos acceder al manager hasta que creemos un usuario de Tomcat con derecho de administrador. Para crearlo debemos modificar el archivo de configuración de usuarios de Tomcat, que se encuentra en `$CATALINA_HOME/conf/tomcat-users.xml`. A dicho archivo le tendremos que añadir las siguientes líneas:

```
<role rolename="manager"/>
<user username="root" password="xxxxxxx" roles="manager"/>
```

Debido a que es un documento xml, debemos respetar el orden de las etiquetas. Dicho de otro modo, poned la línea `<role>` debajo de las que ya están y lo mismo para la línea `<user>`. Con respecto a la línea `<user>` que hemos añadido, podemos ponerle el username que queramos, no hace falta que sea root (la condición de administrador se define en el atributo role, no en el nombre). Ponedle el password que queráis (si alguien se lo está preguntando, si, el password se guarda como texto plano, con lo que tal vez sea una buena idea de seguridad no llamar al administrador root, pues sería demasiado obvio)

Una vez añadido el usuario, reiniciamos el tomcat (recordad, `shutdown.sh`, `startup.sh`) y accedemos, desde nuestro navegador a la dirección <http://localhost:8080/manager/html><sup>(7)</sup>, introducimos los datos (username/password) del usuario recién creado y aparecerá ante nosotros la interfaz del manager. Dicha interfaz consta de 5 partes:

- ◆ **Message:** Aquí se nos mostrará el resultado de las ordenes que le vayamos dando al manager. Pueden ser OK o FAILED.
- ◆ **Manager:** Aquí tenemos 4 opciones. La primera recarga la lista de aplicaciones instaladas actualmente (dicha lista está justo debajo, ahora la veremos con más detenimiento), las dos siguientes nos permiten el acceso a la documentación del manager y la última nos permiten ver el estado del motor de Tomcat.
- ◆ **Applications:** Aquí tenemos la lista de aplicaciones web que está ejecutando Tomcat. Lo que a nosotros nos importa son los commands. Podemos parar (stop), iniciar (start), recargar (reload) o borrar (undeploy) la aplicación. Probad, por ejemplo a parar e iniciar posteriormente vuestra aplicación (acceded a ella con el navegador para ver los resultados). Probad también a modificar vuestra aplicación (haced que diga "Hola bulma!!", por ejemplo), recompilarla y recargarla.



- ◆ **Deploy:** Desde aquí podemos subir aplicaciones directamente al tomcat, que es justo lo que queríamos. Para ver como funciona, copiad la aplicación de ejemplo a un directorio seguro (al HOME; vamos) y borrarla del servidor (!undeploy!). Ahora en donde pone “WAR or Directory URL” (el tercer campo) escribir el path hacía la aplicación (ojo, hay que especificar el protocolo, es decir *file://lo\_que\_sea*, en mi caso *file://home/angel/prueba*“). Le damos a deploy y ya tenemos nuestra aplicación funcionando perfectamente. Comprobarlo como es habitual.
- ◆ **Server information:** Pues eso mismo. Tampoco hay mucho que contar.

## CONCLUSIÓN

Bueno pues eso es todo. Una vez montado el Tomcat, para añadir nuevos servlets solo tenéis que crear la estructura de directorios comentada antes y subirla al servidor (a mano, reiniciando o mediante el manager) Usar el manager puede parecer más inseguro que subir a mano, pero para desarrollar un servlet medianamente complicado se hace casi imprescindible.

Me he dejado varias cosas en el tintero. El uso de los archivos WAR, que no son mas que la anterior estructura de directorios, pero comprimida (son el equivalente en servlets a los archivos JAR normales, la W es de web application). Tampoco he comentado el sistema de hosts virtuales del Tomcat, o los sistemas de seguridad adicionales. Por no mencionar que aparte de Servlets podemos hacer también páginas JSP (Java Server Pages, parecido a PHP o ASP, pero en Java), pero este tutorial es una mera introducción al Tomcat. Para quién quiera profundizar, aquí dejo estos enlaces:

- ◆ [http://www.programacion.com/java/tutorial/servlets\\_basico/](http://www.programacion.com/java/tutorial/servlets_basico/)<sup>(8)</sup> : tutorial básico de programación de servlets.
- ◆ [http://programacion.com/java/tutorial/servlets\\_jsp/](http://programacion.com/java/tutorial/servlets_jsp/)<sup>(9)</sup> : Tutorial básico de Servlets y JSP.
- ◆ <http://java.sun.com/products/servlet/docs.html><sup>(10)</sup> : Documentación oficial. En inglés por supuesto.
- ◆ <http://localhost:8080/tomcat-docs><sup>(11)</sup> : La documentación que tendréis instalada actualmente en vuestro disco tras instalar Tomcat

---

### Lista de enlaces de este artículo:

1. <http://java.sun.com/j2se/1.4.2/download.html>
2. <http://jakarta.apache.org/site/binindex.cgi>
3. <http://jakarta.apache.org/site/sourceindex.cgi>
4. <http://localhost:8080/>
5. <http://localhost/prueba/hola>
6. [http://localhost/\\$servlet-name/\\$url-pattern](http://localhost/$servlet-name/$url-pattern)
7. <http://localhost:8080/manager/html>
8. [http://www.programacion.com/java/tutorial/servlets\\_basico/](http://www.programacion.com/java/tutorial/servlets_basico/)
9. [http://programacion.com/java/tutorial/servlets\\_jsp/](http://programacion.com/java/tutorial/servlets_jsp/)
10. <http://java.sun.com/products/servlet/docs.html>
11. <http://localhost:8080/tomcat-docs>

---

E-mail del autor: seapot\_ARROBA\_terra.es

Podrás encontrar este artículo e información adicional en: <http://bulma.net/body.phtml?nIdNoticia=1888>