

# Curso online: **Instalación, Configuración y Administración de Apache + Tomcat**

## **Módulo 3. Apache Tomcat**

### **Capítulo 2. Historia**

Autores

Janine García Morera

Alexandra López de la Oliva Portugués

Julio Villena Román

Octubre de 2014

## Índice de contenidos

---

<b>Capítulo 2</b>	<b>Historia</b>	<b>2</b>
2.1.	Introducción	2
2.2.	Objetivos	2
2.3.	¿Qué es Apache Tomcat?	3
	2.3.1. Compatibilidad con J2EE	4
	2.3.2. Ventajas de Tomcat frente a otros servidores	4
	2.3.3. Implementación de las especificaciones	5
2.4.	Conclusiones	6

## CAPÍTULO 2 HISTORIA

### 2.1. Introducción

Como ya se ha mencionado, Apache Tomcat es un contenedor de servlets que desarrolla la Apache Software Foundation (ASF), en el ámbito del proyecto Jakarta y como proyecto de primer nivel, y que Sun Microsystems utilizó como implementación de referencia de sus estándares Servlets y JavaServer Pages (JSPs).

Proviene de un servidor de aplicaciones desarrollado inicialmente por Sun Microsystems para promover el desarrollo de aplicaciones Web basadas en Java y que fue cedido a la ASF a finales de los años 90.

Tomcat no se considera como un servidor de aplicaciones al no ser totalmente compatible con todas las APIs J2EE (Java 2 Enterprise Edition), ya que no tiene contenedor de EJBs (Enterprise Java Beans). Sí cumple el resto de las APIs, bien directamente o bien por medio de módulos externos.

Tomcat tiene algunas ventajas con respecto a otros contenedores, principalmente de índole comercial: es una aplicación gratuita, Open Source y además incluye antes que los demás las modificaciones en las especificaciones. Por otro lado, como cumple escrupulosamente las especificaciones (no tiene añadidos), cualquier aplicación desarrollada en Tomcat puede ser portada a otro servidor en teoría sin modificación ninguna.

La puesta en marcha de Tomcat se realiza por medio de un script que permite controlar el funcionamiento del ejecutable. En Windows además se puede arrancar y parar como un servicio. Algunas versiones incluyen, además, un módulo llamado Tomcat Monitor, que permite la gestión del servicio y de los ficheros de registro.

### 2.2. Objetivos

- Conocer los orígenes del servidor Apache Tomcat y las evoluciones que ha sufrido.
- Conocer la estructura de los directorios que contiene el software de Apache Tomcat y de las aplicaciones desplegadas.
- Saber cómo se puede arrancar y parar Apache Tomcat y las variables de entorno que necesita para su funcionamiento.

### 2.3. ¿Qué es Apache Tomcat?

Apache Tomcat es el contenedor de servlets y JSPs, desarrollado por la ASF dentro del proyecto Jakarta, que se utiliza oficialmente en la Reference Implementation para las tecnologías Java Servlet ([www.oracle.com/technetwork/java/index-jsp-135475.html](http://www.oracle.com/technetwork/java/index-jsp-135475.html)) y JSP ([www.oracle.com/technetwork/java/javaee/jsp/index.html](http://www.oracle.com/technetwork/java/javaee/jsp/index.html)).

Las especificaciones para Java Servlet y JSP las diseñó y desarrolló inicialmente Sun Microsystems y ahora Oracle Corporation (Sun Microsystems fue adquirida por Oracle en 2010) bajo el programa Java Community Process ([jcp.org/en/introduction/overview](http://jcp.org/en/introduction/overview)), organización abierta que tiene la responsabilidad del desarrollo de las tecnologías Java.

Apache Tomcat está desarrollado en un entorno abierto y colaborativo y se libera bajo la Apache Software License ([www.apache.org/licenses](http://www.apache.org/licenses)).

La historia de Tomcat se remonta a la implementación de referencia de la API Servlet de Sun. Antes de la aparición de la API Servlet, Java se utilizaba fundamentalmente para desarrollar applets para HTML. Las aplicaciones empresariales desarrolladas en Java tenían un gran tamaño, por lo que los tiempos de descarga necesarios al implementarlas como applets eran muy grandes. La API Servlet llevaba la parte lógica de las aplicaciones Web al lado del servidor en lugar del cliente, como es el caso de los applets, pero el servidor requiere un entorno para el código Java: **un servidor de aplicaciones Web**.

Sun disponía de una especificación API Servlet, pero se negaba a publicarla hasta tener una implementación de referencia (una implementación de referencia es un software en el que se basan las empresas para crear prototipos de aplicaciones con los que poner a prueba los conceptos y la utilización de una nueva tecnología).

En 1996 existían varios contenedores de servlets gratuitos, como es el caso del Jserv de Apache ([freecode.com/projects/apachejserv](http://freecode.com/projects/apachejserv)), que fue seguido por Jetty ([www.eclipse.org/jetty/](http://www.eclipse.org/jetty/)) y por el Locomotive Application Server ([www.boutell.com/lsm/lsmbyid.cgi/001715](http://www.boutell.com/lsm/lsmbyid.cgi/001715)), ya inactivo).

En 1997, Sun desarrolló la implementación inicial de referencia de la API Servlet y la presentó con el nombre de Java Servlet Development Kit (JSDK). Como el interés de Sun era expandir su tecnología, no lo comercializó y lo puso a libre disposición de los usuarios. En la primera mitad de 1998, Sun anunció su nueva especificación JSP. A partir de la release 2.1 de la JSDK, pasó a llamarse Java Servlet Web Development Kit (JSWDK), y se realizó una reescritura completa del core del antiguo servidor JSDK por parte de James Duncan Davidson, de Sun. Esta nueva implementación de referencia incluía un nuevo contenedor servlet llamado Tomcat, cuya versión inicial fue la 3.0 al sustituir la antigua JSDK 2.1.

En octubre de 1999 el desarrollador de la JSWDK convenció a Sun para que donara el código fuente del JSWDK a la ASF. Dos meses después aparecía la primera versión de Tomcat, la 3.0.

En ASF vieron las posibilidades que tenía ese código para desarrollar productos de código abierto para el servidor con calidad comercial, y crearon el Proyecto Jakarta (<jakarta.apache.org>) que ofrece actualmente gran cantidad de productos de código abierto, incluido Tomcat.

En 2001 se publicó la versión 4.0 de Tomcat, llamada Catalina, con un rediseño completo de la arquitectura y un nuevo código. En diciembre de 2003 se publicó la primera release estable de la versión 5.x. La versión 5.5.x surgió en agosto de 2004. La versión 6.x salió por primera vez en diciembre de 2006 y en junio de 2010 la primera versión beta de la 7.x. La primera versión estable de la 8.x salió en septiembre de 2014.

### 2.3.1. Compatibilidad con J2EE

Tomcat no es directamente compatible con todas las APIs de J2EE, pero todas, excepto EJB, están disponibles agregando archivos adicionales Jar descargables gratuitamente. Por ejemplo, si se necesita soporte de EJBs, se puede utilizar JBoss (<www.jboss.org>) que es un servidor EJB de código abierto integrable con Tomcat.

### 2.3.2. Ventajas de Tomcat frente a otros servidores

La primera ventaja (y fundamental) es que es gratuito, pero además es fácil de instalar, se ejecuta en máquinas con pocos requisitos hardware, y, al estar escrito íntegramente en Java a partir de la versión 4, es totalmente portable entre diferentes plataformas y arquitecturas. Además de que es compatible con las APIs más recientes de Java y ocupa muy poco espacio, es muy fiable, y es utilizado por gran cantidad de usuarios.

Una de las grandes ventajas de Tomcat es la sencillez a la hora de desplegar una aplicación web, por lo que es muy recomendable para realizar pruebas de aplicación, con la seguridad de que la aplicación será desplegable en otros servidores.

Entre los servidores J2EE comerciales más conocidos se encuentran **WebLogic** (desarrollado inicialmente por BEA Systems, posteriormente adquirida por Oracle Corporation) y **WebSphere Application Server** (de IBM). **EAServer** de Sybase Inc. es también conocido por ofrecer soporte a otros lenguajes diferentes a Java, como PowerBuilder. Entre los servidores de aplicaciones libres se encuentran **JOnAS** del consorcio ObjectWeb, **JBoss AS** de JBoss (división de Red Hat), **Geronimo** de Apache, **TomEE** de Apache, **Resin Java Application Server** de Caucho Technology, **Blazix** de Desiderata Software, **Enhydra Server** de Enhydra.org y **GlassFish** de Oracle.

Es importante resaltar que mucha gente confunde Tomcat como un servidor de aplicaciones; sin embargo, es solamente un contenedor de servlets. J2EE provee estándares que permiten a un servidor de aplicaciones servir como “contenedor” de los componentes que conforman dichas aplicaciones. Estos componentes, escritos en lenguaje Java, usualmente se conocen como Servlets, JSPs y EJBs y permiten implementar diferentes capas de la aplicación, como la interfaz de usuario, la lógica de negocio, la gestión de sesiones de usuario o el acceso a bases de datos remotas.

Esto no quiere decir que no se puedan desplegar grandes sistemas en producción: de hecho, Tomcat es utilizado como servidor en producción en muchos sitios web.

### 2.3.3. Implementación de las especificaciones

Tomcat es un contenedor de Servlets con un entorno JSP. Un contenedor de Servlets es un shell de ejecución que maneja e invoca servlets por cuenta del usuario.

Técnicamente podemos dividir los contenedores de Servlets en:

1. **Contenedores de Servlets stand-alone (independientes):** este es el modo por defecto usado por Tomcat. El propio contenedor de servlets lleva un servidor Web.
2. **Contenedores de Servlets dentro-de-proceso (inprocess):** el contenedor Servlet es una combinación de un plugin para el servidor web y una implementación de contenedor Java. El plugin del servidor web abre una máquina virtual Java dentro del espacio de direcciones del servidor web y permite que el contenedor Java se ejecute en él. Si una cierta petición debería ejecutar un servlet, el plugin toma el control sobre la petición y lo pasa al contenedor Java (usando JNI – Java Native Interface). Un contenedor de este tipo es adecuado para servidores multi-thread de un sólo proceso y proporciona un buen rendimiento pero está limitado en escalabilidad.
3. **Contenedores de Servlets fuera-de-proceso (outprocess):** el contenedor Servlet es una combinación de un plugin para el servidor web y una implementación de contenedor Java que se ejecuta en una máquina virtual Java fuera del servidor web. El plugin del servidor web y la máquina virtual Java del contenedor Java se comunican usando algún mecanismo IPC (normalmente sockets TCP/IP). Si una cierta petición debería ejecutar un servlet, el plugin toma el control sobre la petición y lo pasa al contenedor Java (usando IPCs – Interprocess Communications). El tiempo de respuesta en este tipo de contenedores no es tan bueno como el anterior, pero obtiene mejores rendimientos en otras cosas (escalabilidad, estabilidad, etc.).



Más información:

[www.java2learn.com/servlet-technology-model/introduction-to-servlets/](http://www.java2learn.com/servlet-technology-model/introduction-to-servlets/)

Tomcat puede utilizarse como un contenedor independiente (principalmente para desarrollo y depuración) o como plugin para un servidor web existente (actualmente se soportan los servidores Apache, IIS y Netscape). Esto significa que siempre que desplaguemos Tomcat tendremos que decidir cómo usarlo, y, si seleccionamos las opciones como plugin (como veremos más adelante), también necesitaremos instalar un adaptador de servidor web (el `mod_proxy_ajp`).

## 2.4. Conclusiones

- Tomcat es la implementación de referencia de Oracle Corporation/Sun Microsystems de sus estándares API Servlet y API JSP.
- Es una **aplicación 100% Java**, por lo que su despliegue es muy sencillo. Necesita para ejecutarse del soporte de una máquina virtual Java. Para Tomcat 7.x se necesita la versión 6 del JDK y para la versión 8.x se necesita la versión 7 del JDK.