

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

**Módulo 4. Cooperación entre Apache y Tomcat**  
**Capítulo 2. Conectores**

Autores

Janine García Morera

Alexandra López de la Oliva Portugués

Julio Villena Román

Octubre de 2014

## Índice de contenidos

---

<b>Capítulo 2</b>	<b>Conectores</b>	<b>2</b>
2.1.	Introducción	2
2.2.	Conectores Web	2
2.3.	Protocolos de Conector	3
2.3.1.	Protocolo AJP	4
2.3.2.	Protocolo WARP	5
2.4.	¿Cómo conectar Apache con Tomcat?	6

## CAPÍTULO 2 CONECTORES

### 2.1. Introducción

La cooperación entre Apache y Tomcat es útil por muchas razones ya analizadas. Realmente Tomcat puede cooperar con cualquier servidor web (especialmente IIS y Sun One), pero la cooperación con Apache es mucho más estrecha.

Los elementos que permiten la cooperación con Apache y Tomcat son los conectores. Los que ofrecen mejor rendimiento son los que utilizan el protocolo AJP (también es posible realizar la cooperación mediante HTTP con el módulo `mod_proxy`).

De todos los módulos, el que ofrece mejor rendimiento es `mod_jk`, aunque probablemente en algún momento ASF abandone definitivamente su desarrollo en beneficio del nuevo módulo `mod_proxy_ajp` (en cualquier caso, la última versión estable hasta la fecha, `jk-1.2.40`, fue liberada en abril de 2014).

Así pues, los objetivos de este apartado son conocer el concepto de cooperación entre Apache y Tomcat y las razones para implementar dicha cooperación, conocer los protocolos que permiten realizar dicha cooperación, y saber qué módulos implementan cada uno de los protocolos y cuáles pueden ser usados en un entorno de trabajo.

### 2.2. Conectores Web

Realmente Tomcat puede cooperar con cualquier servidor Web, pero está especialmente preparado para hacerlo con Apache web server, IIS (Internet Information Server) de Microsoft o Sun One/Oracle iPlanet (antes Sun Java System Web Server, antes Netscape Enterprise Server), por medio de los conectores web de Tomcat. En este capítulo se analizarán las posibilidades de conexión con Apache web server, al tratarse del único producto Open Source de los tres citados.

Todos los conectores web de Tomcat se basan en un módulo de Apache, que ha variado con la versión (actualmente existe `mod_jk`, pero anteriormente estuvieron `mod_jk2`, `mod_jserv`, `mod_webapp`, etc.), y que Apache carga como cualquier otro módulo, y una serie de directivas específicas de ese módulo. En el lado de Tomcat, cada instancia de contenedor tiene un componente de módulo de contenedor en forma de clase Java que implementa la interfaz `org.apache.catalina.Conector`, y que varía en función del tipo de conector.

Cualquiera que sea el tipo de conector que se utiliza, el funcionamiento del sistema es siempre el mismo: el servidor web recibe las peticiones del cliente, contesta aquellas que pueda procesar por sí mismo (normalmente las peticiones de contenido estático o las de contenido dinámico basadas en CGI) y pasa a Tomcat todas las peticiones de ejecución de servlets o JSPs, que las procesa y devuelve la respuesta al servidor web que, a su vez, la pasa al cliente.

Si el sistema está correctamente configurado, sólo pasará a Tomcat peticiones de este tipo, de tal manera que cualquier petición que no sea de ejecución de servlet o JSP no será pasada a Tomcat aunque el servidor web no sepa procesarla (es decir, será tratada como un error por el servidor web).

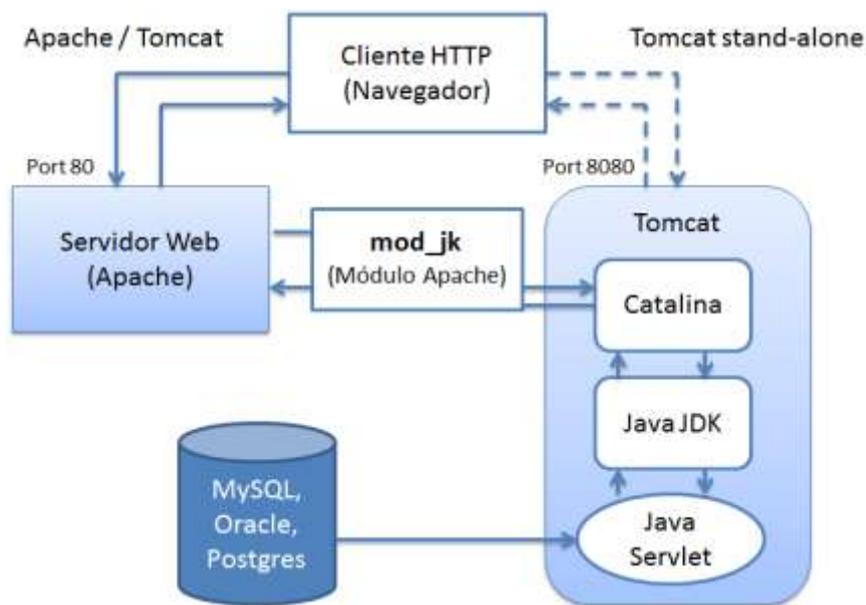


Figura 4.2.1: Conexión entre Apache y Tomcat

### 2.3. Protocolos de Conector

La forma más genérica de realizar la conexión entre Apache y Tomcat es mediante el uso del módulo `mod_proxy`, módulo que implementa la funcionalidad de proxy en Apache y se utiliza con propósito general para este fin.

En relación con Tomcat se implementaría configurando Apache para que remitiera cualquier petición a unos directorios específicos (`/servlet` y `/jsp`, por ejemplo) a Tomcat, asumiendo que todos los servlets y jsp se almacenan en los mismos directorios y que todo el contenido de estos directorios es del mismo tipo:

```
ProxyPass /servlet/ http://localhost:8080/servlet/
ProxyPassReverse /servlet/ http://localhost:8080/servlet/
ProxyPass /jsp/ http://localhost:8080/jsp/
ProxyPassReverse /jsp/ http://localhost:8080/jsp/
```

Este tipo de conexión, que es rápida y sencilla, tiene sin embargo el inconveniente de que no permite el mantenimiento de sesión y no permite el balanceo de carga. Para resolver estos problemas se diseñaron los Protocolos de Conector, esto es, implementaciones específicas (en forma de módulos) que permiten la interrelación entre el servidor web y Tomcat.

El protocolo AJP es un protocolo binario que define el formato de los datos transmitidos y los comandos de control.

### 2.3.1. Protocolo AJP

El protocolo AJP (Apache Jserv Protocol) es una versión optimizada del protocolo HTTP que permite que un servidor web (como Apache) dialogue con Tomcat. Es el protocolo en el que se basaba el módulo `mod_jserv` que permitía la comunicación entre Apache (versiones 1.2 y 1.3) y Jserv (el antecesor de Tomcat) que no permitía servir HTTP. La versión actual (AJP13) utiliza un formato binario para la transmisión de datos entre el servidor web y Tomcat (las versiones anteriores, AJP10 y AJP11 utilizaban un formato textual) y es un protocolo orientado a paquetes.

El web server se comunica con Tomcat sobre conexiones TCP. Para minimizar el impacto en el rendimiento del proceso de creación del socket, el servidor web intenta mantener conexiones TCP persistentes con el contenedor de servlets, y reutilizar cada conexión para varios ciclos de solicitud/respuesta. Pero una vez que una conexión ha sido asignada a una petición en particular, no será utilizada por ninguna otra hasta que el ciclo solicitud/respuesta ha finalizado, lo que hace el funcionamiento mucho más simple, pero como contrapartida el número de conexiones abiertas puede ser muy alto.

Una vez que el servidor web ha abierto una conexión con el contenedor de servlets, ésta puede estar en uno de los dos estados siguientes:

- En estado de espera (idle), si la conexión no está gestionando ninguna petición.
- Asignada, si la conexión está procesando una petición.

Una vez que la conexión está asignada para procesar una determinada petición, la información básica de la petición (como por ejemplo, las cabeceras HTTP) se envía en una codificación binaria, cuyo formato no será tratado en este curso.

Los módulos Apache que implementan, o han implementado este protocolo, son los siguientes:

- **mod\_jserv:** fue el primer módulo de conexión entre Apache y Tomcat (entonces Jserv). Era un módulo de uso obligatorio en entornos en producción, porque Jserv no disponía de pila HTTP. Este módulo ya no está soportado a partir de la versión 5 de Tomcat.
- **mod\_jk:** versión mejorada de `mod_jserv`, ha sido el módulo de interconexión usado a partir de la versión 4 de Tomcat. Tiene compatibilidad con todas las versiones del protocolo AJP existentes hasta ahora (excepto la 1.0). Supone una gran mejora con respecto a `mod_jserv`, sobre todo al ofrecer una mejor compatibilidad con SSL y permitir el uso con más servidores web que `mod_jserv`. Pese a que se produjo una reescritura del código que dio lugar a una nueva versión (`mod_jk2`), Apache recomienda utilizar `mod_jk` para entornos en producción. Es la única opción posible actualmente para servidores web no Apache.
- **mod\_jk2:** es una reescritura de `mod_jk` y usa el Apache Portable Runtime (APR). Se define como la siguiente generación de `mod_jk`. Admite ser usado por los mismos web servers que `mod_jk`, y las mismas versiones de Tomcat pero, pese a que supone una mejora sobre `mod_jk` al haberse diseñado expresamente para ser usado con las versiones 2.x de Apache y ofrecer mejor rendimiento, su desarrollo fue abandonado por la Apache Software Foundation a partir del 15 de noviembre de 2004 (según la web de Apache, por la falta de interés de los desarrolladores) y ya no está soportado. Apache ofrece como alternativa `mod_jk`, cuyas últimas versiones han recibido gran parte del código de `mod_jk2`, o el nuevo módulo `mod_proxy_ajp`.
- **mod\_proxy\_ajp:** el módulo `mod_proxy` ha sido reescrito en las últimas versiones de Apache (a partir de la 2.2.1) para soportar **balanceo de carga** y usar un nuevo transporte llamado `mod_proxy_ajp`, que provee soporte para el protocolo AJP13 y que requiere el servicio `mod_proxy`. Este módulo se proporciona con Apache (no con Tomcat) a partir de la versión 2.2.

### 2.3.2. Protocolo WARP

A título de curiosidad, hace años se definió un protocolo llamado WARP. El protocolo WARP se basaba en la definición de una estructura de paquetes para realizar la transferencia de datos entre el servidor web y el contenedor de servlets. Parece ser que recibe su nombre de la unidad de medida de la velocidad usada en la serie Star Trek, porque se diseñó específicamente pensando en el rendimiento (un WARP es una unidad de medida superior a la velocidad de la luz, en la serie).

El protocolo WARP se plasmó en implementación de un solo módulo, el `mod_webapp`. Este módulo sólo fue utilizable en la versión 4 de Tomcat y con gran cantidad de limitaciones (no admite balanceo de carga y además tiene muchos errores). El módulo `mod_webapp` no funcionaba en Windows, su desarrollo fue abandonado por Apache y además no está soportado desde hace mucho tiempo.

## 2.4. ¿Cómo conectar Apache con Tomcat?

El caso más habitual de cooperación (y el que veremos en este curso) es el del servidor web Apache y el contenedor de Servlets Tomcat. La forma de conseguir la cooperación varía en función de los conectores utilizados y de las versiones. En nuestro caso veremos las posibilidades de cooperación entre Apache 2.4 y Tomcat 7.0.

Las dos posibilidades de cooperación entre los dos servidores se reducen al uso del protocolo HTTP o al protocolo AJP13.

- **HTTP.** El conector HTTP de Tomcat está activo por defecto. Tomcat soporta `mod_proxy` como balanceador de carga (incluido desde la versión 2.2 de Apache como parte de la distribución estándar).
- **AJP13.** El conector AJP tiene un mejor rendimiento que un proxy HTTP, por lo que se recomienda su utilización para entornos de producción (clusters de servidores). Con estas versiones, hay dos posibilidades:
  - ➔ **Conector JK2.** Como se ha indicado antes, este módulo ya no se desarrolla, por lo que aunque es factible su uso, no es recomendable.
  - ➔ **`mod_proxy_ajp`.** Nuevo en la versión 2.2 de Apache, está en fase de perfeccionamiento y en el futuro se prevé que sustituya al anterior.

Cualquiera de las dos opciones es válida para montar un entorno de cooperación Apache+Tomcat. Sin embargo, el módulo `mod_proxy_ajp` es un módulo de reciente incorporación a Apache, por lo que aún no está totalmente maduro. Es recomendable por lo tanto, en un entorno en producción, el uso del módulo `mod_jk`, más complejo y trabajoso de configurar, pero que ofrece más confianza.

En los siguientes capítulos vamos a analizar las dos posibilidades recomendadas: `mod_jk` y `mod_proxy_ajp`.