

2.7.- TOMCAT.

2.7.1. Introducción.

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. Tomcat es el servidor web y de aplicaciones del proyecto Jakarta, se dice que es servidor web ya que gestiona solicitudes y respuestas Http (incluye el servidor Apache) gracias a sus conectores Http; además es servidor de aplicaciones o contenedor de Servlets/JSP (Catalina).

Se pueden observar varios tipos de contenedores de Servlets:

- A. Contenedores de Servlets Stand-alone (Independientes): son una parte integral del servidor web. Este es el caso de un servidor web basado en Java. Es el modo por defecto usado por Tomcat. Sin embargo, la mayoría de los servidores, no están basados en Java, lo que lleva a otros dos tipos diferentes de contenedores.
- B. Contenedores de Servlets dentro-de-proceso: 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 JVM (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 debe ejecutar un servlet, el plugin toma el control sobre la petición y lo pasa al contenedor Java. Un contenedor de este tipo es adecuado para servidores multi-hilos de un sólo proceso y proporciona un buen rendimiento. El inconveniente es que está limitado en escalabilidad.
- C. Contenedores de Servlets fuera-de-proceso: el contenedor Servlet es una combinación de un plugin para el servidor web y una implementación de contenedor Java que se ejecuta en una JVM fuera del servidor web. El plugin del servidor web y el JVM 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). El tiempo de respuesta en este tipo de contenedores no es tan bueno como el anterior, pero posee una mayor escalabilidad, estabilidad, etc.

Tomcat puede utilizarse como un contenedor solitario (principalmente para desarrollo y depuración) o como plugin para un servidor web existente. Al desplegar Tomcat se debe decidir cuál será su uso, y si es alguna de las dos últimas opciones será necesario instalar un adaptador de servidor web.

Luego, Tomcat puede funcionar como servidor web por sí mismo. En sus inicios existió la percepción de que el uso de Tomcat de forma autónoma era sólo recomendable para entornos de desarrollo y entornos con requisitos mínimos de

velocidad y gestión de transacciones. Hoy en día ya no existe esa percepción y Tomcat es utilizado como servidor web autónomo en entornos con alto nivel de tráfico y alta disponibilidad.

2.7.2. Estructura de directorios.

Una vez realizada la instalación de Tomcat, en la cual no se ha entrado en detalles ya que posteriormente se verán todos los pasos a seguir, se obtiene una estructura de directorios Tomcat entre los que destacan^[20]:

Directorio	Descripción
/bin	Núcleo de Tomcat. Contiene, entre otros, los scripts de arrancar/parar.
/classes	Se puede crear este directorio para añadir clases adicionales al classpath. Cualquier clase que añadamos a este directorio encontrará un lugar en el classpath de Tomcat.
/common	Clases comunes que pueden utilizar Catalina y las aplicaciones web.
/conf	Contiene varios ficheros de configuración incluyendo server.xml (fichero de configuración principal de Tomcat) y web.xml que configura los valores por defecto para las distintas aplicaciones desplegadas en Tomcat.
/logs	Directorio donde Tomcat sitúa los ficheros de diario.
/lib	Contiene varios ficheros <i>jar</i> que son librerías utilizadas por Tomcat.
/server	Clases utilizadas solamente por Catalina.
/shared	Clases compartidas por todas las aplicaciones web.
/src	Los ficheros fuentes del API Servlet.
/webapps	Contiene aplicaciones Web.
/works	Directorio donde Tomcat sitúa los ficheros intermedios durante su trabajo.

Tabla 46.- Directorios Tomcat.

2.7.3. Ficheros de configuración.

Los aspectos de configuración más importantes se encuentran recogidos en dos ficheros situados bajo "%CATALINA_HOME%\conf":

- *server.xml*: fichero de configuración global de Tomcat. Recoge los valores por defecto a utilizar por todas las aplicaciones web cargadas en la instancia de Tomcat, como pudiera ser por ejemplo la página a cargar por defecto.
- *web.xml*: configura los distintos contextos en Tomcat. Contiene la definición estructural del servidor: nombre del host, servicios, conectores, etc.

2.7.3.1. *server.xml*.

El fichero de configuración principal de Tomcat es *server.xml*. Es el descriptor de la configuración del servidor. Las características más importantes que describe son puertos de invocación, de parada y conectores, así como describir los contextos de aplicación.

Tiene dos objetivos:

- Proporcionar una configuración inicial para los componentes de Tomcat.
- Especificar la estructura de Tomcat, es decir, permitir que Tomcat arranque y se construya a sí mismo.

Este fichero de configuración está escrito en XML y sus elementos más importantes se describen en la siguiente tabla:

Etiqueta	Descripción
<server> </server>	(Único y engloba toda la configuración) Define el elemento de configuración básico del fichero server.xml. Es único y contiene uno o más servicios ("Service"). El atributo "port" indica el puerto destinado a la escucha del comando de cierre, indicado por "shutdown" o cierre.
<Listener/>	(Único) Permite definir las clases JMX que permitirá escuchar Tomcat.
<GlobalNamingResources> </GlobalNamingResources>	(Único) Permite definir elementos JNDI para ser utilizados globalmente.
<Service> </Service>	Estas etiquetas permiten agrupar uno o más conectores de forma que compartan un único contenedor de aplicaciones. Poseen un único atributo, "name", que fija los identificadores individuales. Si "name" se fija como "Catalina" o "Tomcat-Standalone", se habilitará a Tomcat como servidor web independiente.
<Connector />	(En el interior de "Service") Connector representa una conexión al usuario a través de un servidor Web o directamente al navegador del usuario (en una configuración independiente). Connector es el responsable del control de los hilos en Tomcat y de leer/escribir las peticiones/respuestas desde los sockets conectados a los distintos clientes. Entre sus parámetros de configuración están el puerto de escucha, "port", la clase encargada de su definición, "className" y el número máximo de conexiones simultáneas permitidas, "acceptCount".
<Engine> </Engine>	(En el interior de "Service") Punto donde se procesan las peticiones que llegan a los "Connector" que posean en la cabecera el valor de "defaultHost" como destino.
<Logger/>	(En el interior de "Service" o de "Host") Permite establecer el nombre del fichero de logs. Como parámetros tiene la clase encargada de su definición, "className", el formato nombre del archivo, como la unión de un prefijo, "prefix", y un sufijo, "suffix".
<Host></Host>	Definen uno o más elementos Host virtuales para atender las peticiones.
<Context> </Context>	(En el interior de "Host") Se utiliza para indicar la ruta ("docBase") a partir de la cual se encuentran las aplicaciones a ser ejecutadas en Tomcat y el path url ("path") a partir del cual acceder a los servicios.

Tabla 47.- Etiquetas a destacar de server.xml.

2.7.3.2. *web.xml*.

Tomcat permite al usuario definir los valores por defecto de *web.xml* para todos los contextos poniendo un fichero *web.xml* por defecto en el directorio “/conf”. Al construir un nuevo contexto, Tomcat usa el fichero *web.xml* por defecto, como la configuración base, y el fichero *web.xml* específico de la aplicación (el localizado en el “WEB-INF/web.xml” de la aplicación).

En la estructura de *web.xml* destacan los siguientes elementos:

Etiqueta	Descripción
<servlet> </servlet>	Declara la configuración de los servlets.
<servlet-name> </servlet-name>	Indica el nombre de la clase.
<servlet-class> </servlet-class>	Indica el paquete donde se encuentra la clase.
<servlet-mapping> </servlet-mapping>	Relaciona los servlets declarados con las “url” que van a escuchar.

Tabla 48.- Etiquetas a destacar de web.xml.

2.7.4. Scripts de Tomcat.

Tomcat es un programa Java, y por lo tanto es posible ejecutarlo desde la línea de comandos, después de configurar algunas variables de entorno. Sin embargo, configurar cada variable de entorno y seguir los parámetros de la línea de comandos usados por Tomcat es tedioso y propenso a errores. En su lugar, Tomcat proporciona unos scripts para configurarlo, arrancarlo y pararlo fácilmente.

La siguiente tabla presenta los scripts más importantes para el usuario:

Script	Descripción
tomcat	El script principal. Configura el entorno apropiado, incluyendo <i>CLASSPATH</i> , <i>TOMCAT_HOME</i> y <i>JAVA_HOME</i> , y arranca Tomcat con los parámetros de la línea de comando apropiados.
startup	Arranca Tomcat en segundo plano. Acceso directo para Tomcat start.
shutdown	Para Tomcat (lo apaga). Acceso directo para Tomcat stop.

Tabla 49.- Script Tomcat.

El script más importante para los usuarios es *tomcat* (*tomcat.sh/tomcat.bat*). Los otros scripts relacionados con *tomcat* sirven como un punto de entrada simplificado a una sola tarea (configuran diferentes parámetros de la línea de comandos, etc.).

2.7.5. Aplicación Web.

Con toda la información sobre JSP, la creación del servlet y el contenedor de servlet ya se posee todo lo necesario para crear una aplicación Web.

Una vez creado el servlet, se necesita desplegarlo. La especificación Java Servlet 2.2 presenta dos características importantes: una aplicación Web y un archivo de aplicación Web (*WAR*).

Una aplicación Web es una colección de servlets, páginas HTML, clases, y otros recursos que se pueden empaquetar y ejecutar en varios contenedores.

El archivo *WARs* representa la aplicación web java empaquetada o comprimida lista para ser desplegada en el servidor.

2.7.6. Estructura de directorios de una Aplicación Web.

La especificación Servlet 2.2 define la estructura de directorios para los ficheros de una aplicación Web. En primer lugar, se debe especificar la ruta donde se sitúan las aplicaciones Web en el contenedor Tomcat:

`%CATALINA_HOME%/webapps`

* Nota: `%CATALINA_HOME%` es la ruta donde se ha instalado Tomcat.

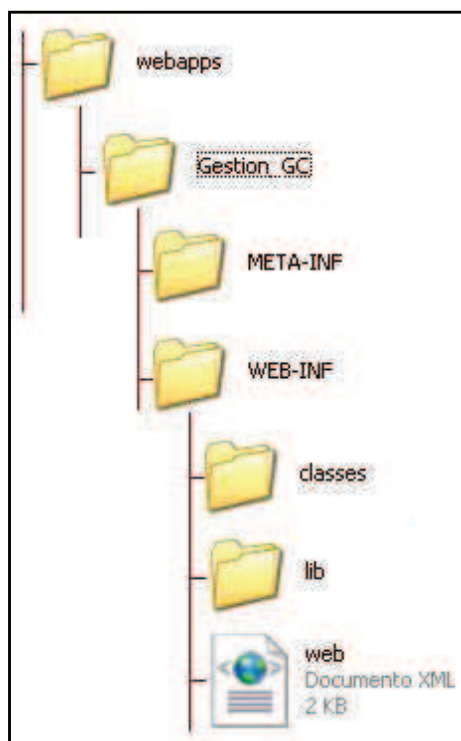


Ilustración 26.- Estructura de directorios de una aplicación Web.

Cada aplicación Web se encuentra en un directorio debajo de *webapps* (cada directorio debajo de *webapps* se denomina contexto de Tomcat). Por tanto, cada directorio corresponde a un contexto y por defecto existe un contexto *ROOT*. En el interior de cada contexto la estructura de directorios es siempre la misma:

- Contexto, directorio superior o raíz: posee el nombre de la aplicación y define la raíz de documentos para la aplicación Web (en el caso de la ilustración corresponde con “*Gestion_GC*”).
- META-INF: contiene información útil de herramientas Java.
- WEB-INF: contiene el fichero de configuración *web.xml* y los siguientes directorios:
 - *classes*: contiene las clases Java adicionales a las del JDK que serán empleadas en los JSP’s y servlets.
 - *lib*: contiene las bibliotecas (comprimidas con *jar*) que utiliza la aplicación.