

## 1. Host virtuales en Tomcat

La necesidad de incluir más de una aplicación desplegada en el servidor Tomcat, y que cada una responda a un nombre diferente, nos hace retomar el concepto de virtual host.

En los siguientes apartados veremos como podemos crear host virtuales en Tomcat.

### 1.1 Modificar el archivo Hosts en windows

La mayoría de los sistemas operativos, incluyendo todas las versiones de Microsoft Windows, utilizan el archivo Hosts para traducir nombres de dominio a sus respectivas direcciones IP. Si no se dispone de un servidor DNS, esta es una opción para emularlo.

En los sistemas operativos de Microsoft, el archivo de Hosts se encuentra en la siguiente ruta:

C:\Windows\System32\drivers\etc

El archivo Hosts es un archivo de texto que no tiene extensión. Contiene líneas de texto en las que encontramos direcciones IP, seguido de uno o más nombres de dominio.

Debes editar este archivo y modificarlo para que traduzca tu nombre de dominio a tu dirección IP.

### 1.2 Incluir Virtual Host en Tomcat

En el archivo de configuración por defecto (\conf\server.xml) vemos que se establece un único servicio y una única instancia del servidor. El elemento <Server>.

```
<Server port="8005" shutdown="SHUTDOWN">
```

El puerto de Tomcat es 8080 y sin embargo está configurado en el puerto 8005. Este puerto es en el que se inicia una instancia del servidor (JVM) y en el que escucha por si llegan señales de apagado (shutdown), que podrían generarse en una máquina remota, o desde el propio servidor.

El elemento <Server> puede contener otros tres:

- <Service>: Un grupo de conectores asociados con un motor. Es necesario al menos uno.
- <Listener>: Clases que tienen que escuchan y manejan eventos que tienen que ver con el ciclo de vida del servidor, por ejemplo después de arrancar, etc.
- <GlobalNamingResources>: Recursos globales que pueden ser usados en esta instancia del servidor por los componentes que los necesiten, por ejemplo una base de datos.

## <Service>

El propósito de un servicio es agrupar un motor que procese las peticiones con uno o más conectores que gestionen los protocolos de comunicación. El servicio por defecto es el motor Catalina.

```
<Service name="Catalina">
```

Un <Service> contiene al menos un <Connector> y solo un <Engine> que es obligatorio.

<Connector>

Este elemento tiene mucho que ver con los dos modos de funcionamiento de Tomcat:

- Como servidor único: En el que Tomcat realiza las funciones de servidor de aplicaciones y de servidor web.
- Como servidor de aplicaciones: En el que Tomcat colabora con un servidor web que hace de frontend. El servidor web dirige todas peticiones de JSPs y Servlets a Tomcat.

Los dos conectores más comunes son HTTP y AJP. El segundo es usado para conectar con los servidores (Apache u otros). Ambos pueden funcionar con SSL para mejorar la seguridad.

El puerto por defecto para HTTP es 8080. Se puede cambiar, y si por ejemplo Tomcat va a estar en producción como un servidor en solitario podríamos modificarlo al puerto estándar de HTTP, el 80.

<Engine>

El motor es quien procesa las peticiones realmente. El nombre es el que le demos a la instancia (al service) y defaultHost indica a qué host virtual se le pasará una petición en caso de que no se especifique ninguno ya que el mismo motor puede procesar peticiones dirigidas a múltiples hosts virtuales de los especificados en este archivo.

```
<Engine name="Catalina" defaultHost="localhost">
```

Un motor contiene uno o más <Host>. Para incluir incluir un Virtual Host en Tomcat, tenemos que añadir una entrada <Host> con los datos del host virtual que se quiera añadir.

```
<Host name="nombreDeDominio" appBase="webapps" unpackWARs="true" autoDeploy="true">
  <Alias>AliasDeDominio</Alias>
  <Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs" prefix="NombreAPP_access_log" suffix=".txt" pattern="%h %l %u %t &quot;%r&quot; %s %b" />
  <Context path="" docBase="RutaDesdeLaRaizDeLaAplicacion" debug="0" reloadable="true"/>
</Host>
```

## 1.3 Actividades

Completa el documento a entregar de tu aplicación calculadora con los siguientes apartados.

- Añade un host virtual para tu aplicación calculadora. De esta forma, tu aplicación responderá a una url (como por ejemplo `www.micalculator.org`)
- Haz que funcione con dos nombres de dominio. Para ello tendrás que definir un Alias, y previamente incluir ese alias en el archivo de host para la misma ip.
- Haz de tu servidor un servidor seguro redireccionando todo el tráfico al puerto 443.

Cambiar tu conector 8080 para que escuche en el puerto 80. Ahora debes hacer que las peticiones a puerto 80 (http) sean redirigidas a 443. El conector SSL que ya tienes configurado con el KeyStore correspondiente, cambia el puerto 8443 por 443.

```
<Connector port="8080" protocol="HTTP/1.1"
    connectionTimeout="20000"
    redirectPort="443" />
```

```
<Connector port="80" protocol="HTTP/1.1"
    connectionTimeout="20000"
    redirectPort="443" />
```

```
<Connector protocol="org.apache.coyote.http11.Http11NioProtocol"
    port="443" maxThreads="200"
    scheme="https" secure="true" SSLEnabled="true"
    keystoreFile="C:\fichero.jks" keystorePass="abc123."
    clientAuth="false" sslProtocol="TLS"/>
```

**Nota:** si usamos el puerto 443, evitamos que en nuestra url en la barra del navegador, nos aparezca el puerto detras de la url.

- Comprueba que puedes acceder a tu web Calculadora desde otro ordenador diferente. Recuerda que debes modificar el archivo Hosts en dicho ordenador, o hacer uso de un servidor DNS para que resuelva tu dominio calculadora.