Configurando un Servidor Web

Francisco Javier Otero Herrero

Grupo ATU

7-4-2025

Configurando un Servidor Web

Francisco Javier Otero Herrero | GRUPO ATU

Configurando un Servidor Web

Continuando con el caso anterior, vamos a seguir realizando la configuración del servidor de la actividad anterior y tomando como punto de partida las respuestas del ejercicio anterior. Indica y explica los pasos que has seguido en una grabación resumen de vídeo del proceso de instalación y configuración que pide este ejercicio en tu ordenador.

En una práctica anterior dejamos constancia de la instalación de Apache en Windows Server 2022 al cual le añadimos el módulo de PHP, junto MySQL y PhpMyAdmin mediante explicaciones cortas y capturas de pantalla. En esta práctica se va realizar la instalación de Tomcat en el mismo servidor y configurar un conector llamado AJP para conseguir comunicar Apache con Tomcat.

Dejamos todo documentado en último apartado llamado ***Anexo.***

PREGUNTAS/ACTIVIDADES A REALIZAR

1. ***Para instalar módulos, ¿cuál es el primero que debe ser instalado/activado? ¿Qué módulo y cómo se instalaría para poder ejecutar Java en Apache? ¿Cómo verificarías que el módulo de Java está correctamente instalado?***

   Para poder ejecutar aplicaciones ***Java (como servlets o JSP) en Apache HTTP Server,*** no se hace directamente desde Apache como se hace con ***PHP***, sino que se utiliza un conector entre Apache y un contenedor ***de servlets Java, como Apache Tomcat.*** La integración más común se realiza mediante el módulo ***mod_jk o mod_proxy_ajp.*** Vamos paso por paso:

   Para ejecutar Java en Apache, necesitas primero un contenedor de servlets, como Tomcat. Luego, debes instalar y configurar un módulo conector, siendo el más común:

   - *mod_jk (más tradicional)*
   - *mod_proxy_ajp (más moderno, viene incluido en Apache)*

   ¿Qué módulo y cómo se instalaría?

   - ***Opción 1: Usar mod_proxy_ajp (más sencillo, viene con Apache)***
   - Habilitar el módulo en Apache (en Windows Server o Linux)
   - ***Opción 2: Usar mod_jk (requiere instalación manual)***
   - **Descargar mod_jk desde el sitio de Apache:**
   - *https://tomcat.apache.org/connectors-doc/*
   - *Copiar el archivo mod_jk.so al directorio modules de Apache.*
   - *Editar httpd.conf para cargar el módulo*

Configurando un Servidor Web

¿Cómo verificar que el módulo de Java está correctamente instalado?

Con mod_proxy_ajp o mod_jk, puedes verificar:

- *Reinicia Apache y asegúrate de que no da errores.*
- *Accede a tu aplicación Java mediante Apache:*
- *Por ejemplo: http://localhost/app*
- *Verifica los logs de Apache:*
- *logs/error.log o logs/mod_jk.log (si usas mod_jk).*
- *Debe mostrar que las peticiones están siendo enviadas correctamente a Tomcat.*
- *Verifica que Tomcat esté corriendo y que responde en el puerto 8009 (AJP).*

2. *Define y e x p l i c a el proceso de configuración de permisos de acceso y ejecución.*

La configuración de permisos de acceso y ejecución es el proceso mediante el cual se determina quién puede ver, acceder o ejecutar archivos y directorios dentro del servidor. Es una medida de seguridad esencial para proteger archivos sensibles y controlar el comportamiento de las aplicaciones web. Porque es tan importante:

- *Evita accesos no autorizados*
- *Restringe la ejecución de scripts peligrosos*
- *Protege datos confidenciales*
- *Cumple con buenas prácticas de seguridad en servidores web*

Como se configura:

En Apache (nivel de servidor web):

- Se usa el archivo de configuración httpd.conf o archivos .htaccess.
- Ejemplo básico:

```
<Directory "C:/xampp/htdocs/privado">
    Options -Indexes
    Require all denied
</Directory>
```

- ***Options -Indexes:*** evita que se pueda listar el contenido si no hay un index.html
- ***Require all denied:*** nadie puede acceder a esa carpeta

Configurando un Servidor Web

En Windows (nivel de sistema operativo):

- Haz clic derecho sobre el archivo o carpeta (por ejemplo, htdocs)
- Selecciona "Propiedades" > "Seguridad"
- Elige el usuario o grupo (por ejemplo, IUSR o Everyone)
- Asigna los permisos deseados:
  - Leer → permite ver archivos
  - Escribir → permite modificar archivos
  - Ejecutar → permite ejecutar scripts o programas
  - Control total → acceso total (recomendado solo para administradores)

Comprobar si están bien configurados:

- Intenta acceder desde un navegador a las rutas protegidas o habilitadas.
- Revisa los logs de Apache (error.log y access.log)
- Usa herramientas como telnet o curl para probar accesos desde diferentes IPs
- Comprueba si al ejecutar un archivo PHP o JSP, responde correctamente o da error 403/500

3. ***¿Cómo configurarías los parámetros de Apache para ofrecer el mejor rendimiento posible?***

I. *Elegir el MPM adecuado (Multi-Processing Module)*

Apache puede funcionar con distintos módulos de procesamiento (MPM). Los más comunes:

prefork → estable, pero usa más memoria (procesos por cada petición).

worker → usa hilos, más eficiente.

event → mejor para sitios con muchas conexiones simultáneas (ideal para rendimiento).

II. *Ajustar el número de procesos/hilos*
Ejemplo para event o worker:

```
<IfModule mpm_event_module>
    StartServers             2
    MinSpareThreads         25
    MaxSpareThreads         75
    ThreadsPerChild         25
    MaxRequestWorkers      150
    MaxConnectionsPerChild 1000
</IfModule>
```

Configurando un Servidor Web

- ***ThreadsPerChild:*** cuántas conexiones simultáneas maneja cada proceso
- ***MaxRequestWorkers:*** máximo de clientes a la vez (antes era MaxClients)
- ***MaxConnectionsPerChild:*** evita que procesos se queden colgados mucho tiempo.

III. *Habilitar y configurar caché*
IV. *Desactivar módulos innecesarios*
V. *Optimizar compresión y transferencia*
VI. *Habilitar KeepAlive*
VII. *Pruebas y monitorización:*

- Una vez configurado, puedes:
  - Medir el rendimiento con herramientas como ApacheBench (ab) o JMeter
  - Monitorizar consumo con el Administrador de tareas en Windows
  - Revisar los logs para cuellos de botella

4. *Desarrolla una documentación de operación con los errores más comunes que podemos encontrarnos en Apache y las posibles soluciones.*

Error 1: "AH00558: Could not reliably determine the server's fully qualified domain name"

- ***Causa:*** Apache no tiene configurado un nombre de servidor.
- ***Solución:*** Edita el archivo httpd.conf y añade una línea con el nombre del servidor: “ServerName localhost”. Si se está en un entorno real, reemplazar localhost por el dominio o la IP del servidor.

Error 2: 403 Forbidden

- ***Causas:***
  - El servidor no tiene permisos para acceder al archivo o carpeta.
  - Falta un archivo index.html o similar.
  - Permisos incorrectos en Windows o en la configuración Apache.
- ***Soluciones:***
  - Verifica los permisos del archivo/carpeta (clic derecho > Seguridad en Windows).
  - Asegúrate de que haya un archivo de inicio válido (index.html, index.php, etc.)
  - Revisa si Require all denied está bloqueando el acceso:

```
<Directory "C:/xampp/htdocs">
Require all granted
</Directory>
```

Configurando un Servidor Web

Error 3: 404 Not Found

- ***Causa:*** El archivo solicitado no existe en la ruta indicada o Apache no está apuntando al directorio correcto.

- ***Soluciones:***
    - Verifica que el archivo esté realmente en la carpeta correspondiente (htdocs o la configurada).
    - Comprueba la ruta en la URL.
    - Asegúrate de que no hay errores de mayúsculas/minúsculas.

Error 4: 500 Internal Server Error

- ***Causa:*** *Error genérico, suele deberse a fallos en scripts PHP o configuraciones incorrectas (.htaccess mal escrito, por ejemplo).*

- ***Soluciones:***
    - Activa el display_errors en PHP para ver más detalles.
    - Revisa el archivo .htaccess, puede tener sintaxis incorrecta.
    - Consulta los archivos de log de Apache (logs/error.log).

Error 5: "Port 80 is already in use"

- ***Causa:*** Otro programa (como Skype o IIS) está usando el puerto 80.
- ***Soluciones:*** Cambia el puerto de Apache en httpd.conf: ***Listen 8080***

Error 6: "AH00072: make_sock: could not bind to address"

- ***Causa:*** Apache no puede abrir el puerto especificado (conflicto o falta de permisos).
- ***Soluciones:***
    - Ejecuta Apache como administrador.
    - Cambia el puerto en httpd.conf, como en el error anterior.

Configurando un Servidor Web

Error 7: "Access denied for user 'root'@'localhost'" (en phpMyAdmin/MySQL)

- ***Causa:*** Contraseña incorrecta o usuario mal configurado.
- ***Soluciones:***
    - Verifica el archivo config.inc.php de phpMyAdmin:

```
$cfg['Servers'][$i]['auth_type'] = 'cookie';
$cfg['Servers'][$i]['user'] = 'root';
$cfg['Servers'][$i]['password']='tu_contraseña';
```

Asegúrate de que el usuario root tenga permisos desde localhost.

Error 8: "Client denied by server configuration"

- ***Causa:*** El acceso ha sido bloqueado por las directivas de Apache (Require, Order, etc.)
- ***Soluciones:*** En Apache 2.4, asegúrate de usar:

```
Require all granted
```

- En versiones más antiguas (2.2), puede usarse:

```
Order allow,deny
Allow from all
```

Recomendaciones adicionales

- Revisa siempre los archivos de log:
    - logs/error.log
    - logs/access.log
- Asegúrate de reiniciar Apache después de cualquier cambio:
    - `httpd -k restart`
- Usa navegadores en modo desarrollador (F12) para ver más detalles de errores HTTP.

Configurando un Servidor Web

5. Anexo: Instalar Tomcat para poder ejecutar Java en Apache

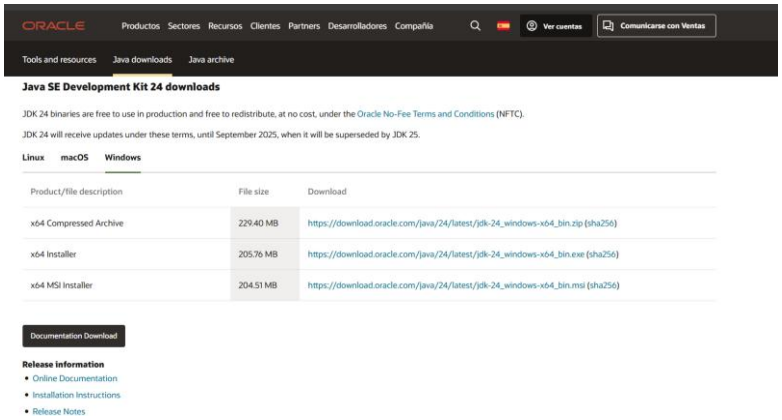
Vamos a proceder con la instalación de ***Apache Tomcat*** en nuestro ***Windows Server 2022*** que tenemos instalado en máquina virtual. Con esto conseguiremos que nuestro server sea capaz de ejecutar aplicaciones ***Java.***

Antes de instalar Tomcat, debemos asegurarnos de que tenemos lo siguiente:

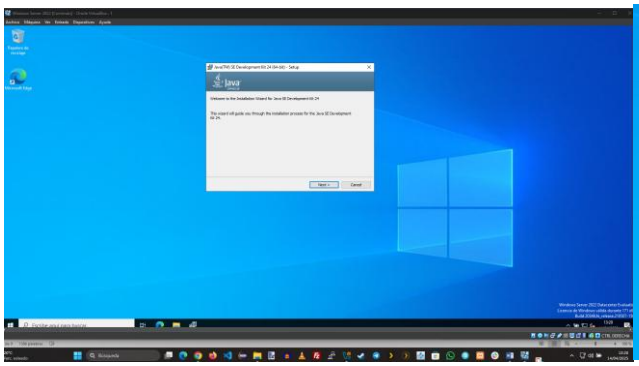
- ***Java Development Kit (JDK)***
- ***Permisos de administrador***
- ***Espacio en disco***

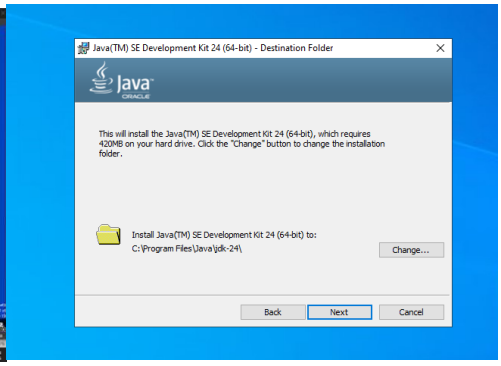
En nuestro caso cumplimos dos de los tres requisitos, por ello vamos a comenzar con la instalación del ***JDK.*** Descargaremos desde el sitio oficial:

https://www.oracle.com/es/java/technologies/downloads/#jdk24-windows

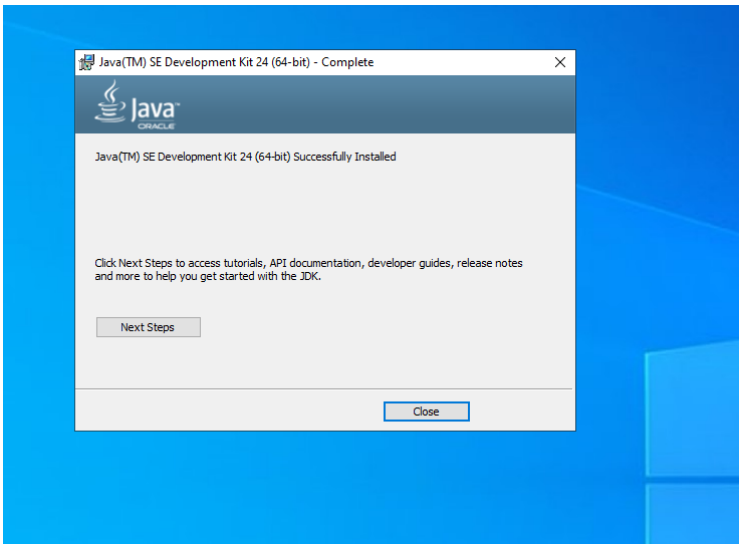


Una vez descargado, comenzamos instalación:

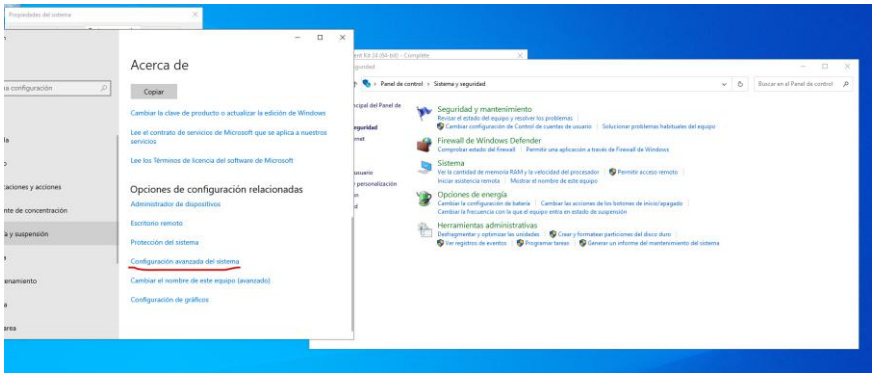




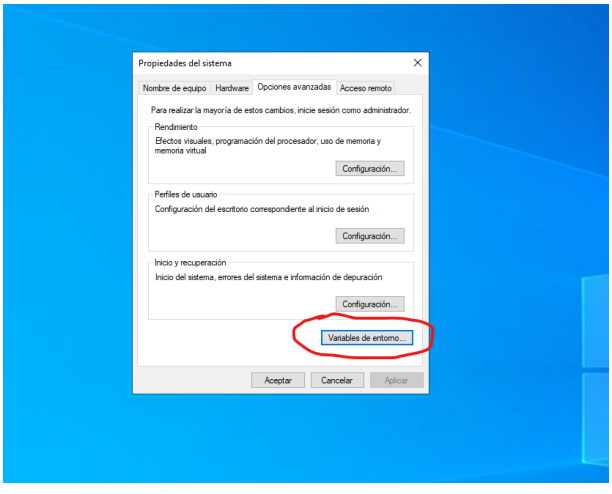
Configurando un Servidor Web



El siguiente paso será configurar la variable de entorno que llamaremos ***JAVA_HOME.*** Para ello buscamos el panel de control, vamos a Sistema, y después en configuración avanzada del sistema.

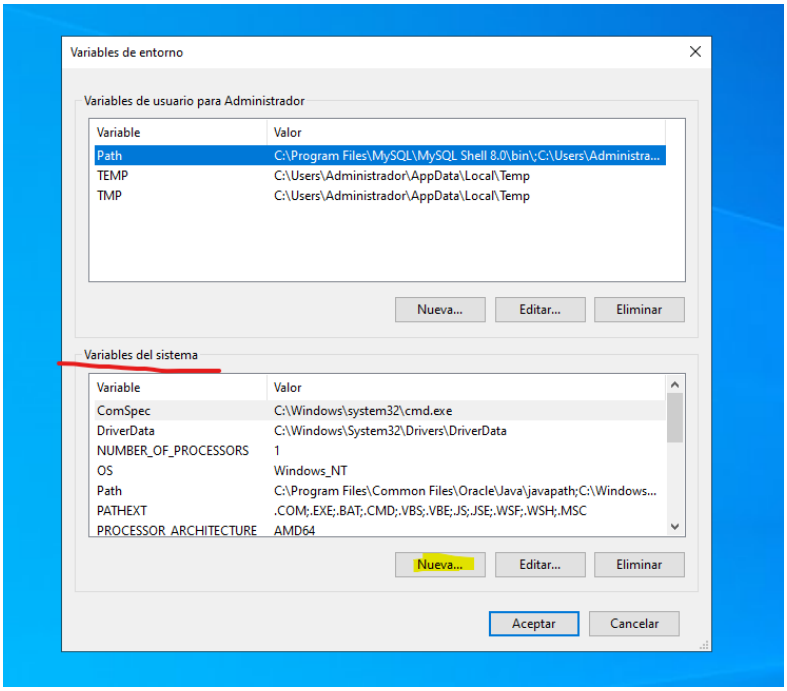


En la ventana que nos aparece accederemos a la parte de variables de entorno.

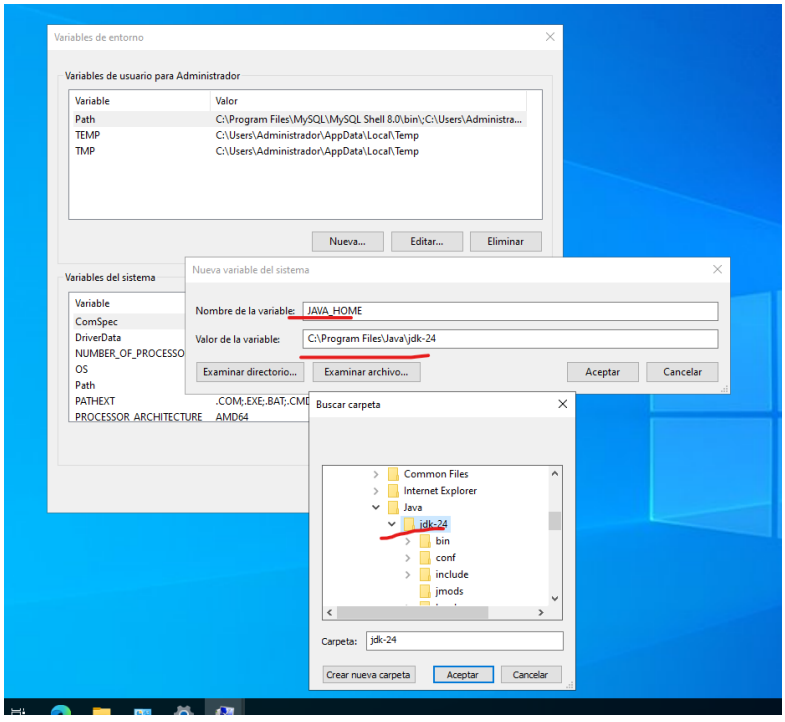


Configurando un Servidor Web

Ahora creamos una nueva variable de entorno.

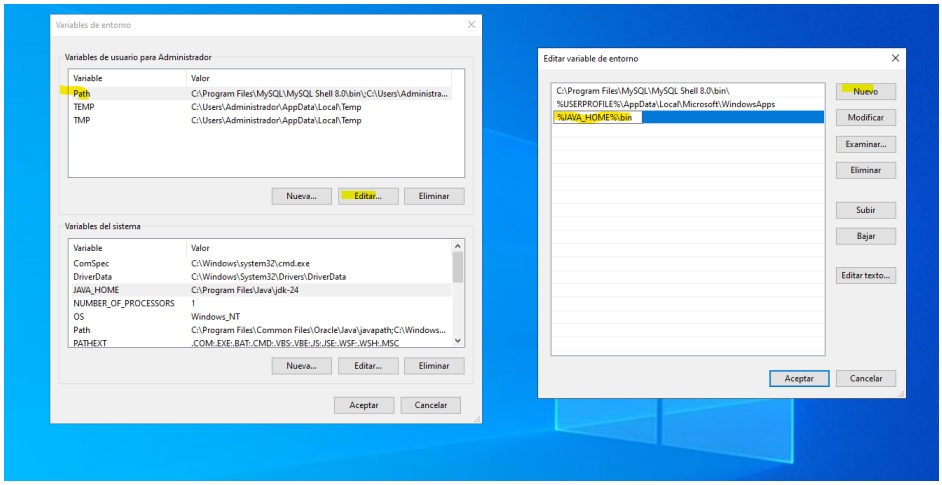


Al pulsar sobre nueva se abrirá una nueva ventana, debemos añadir el nombre y el valor de la variable como se puede ver en la siguiente imagen.

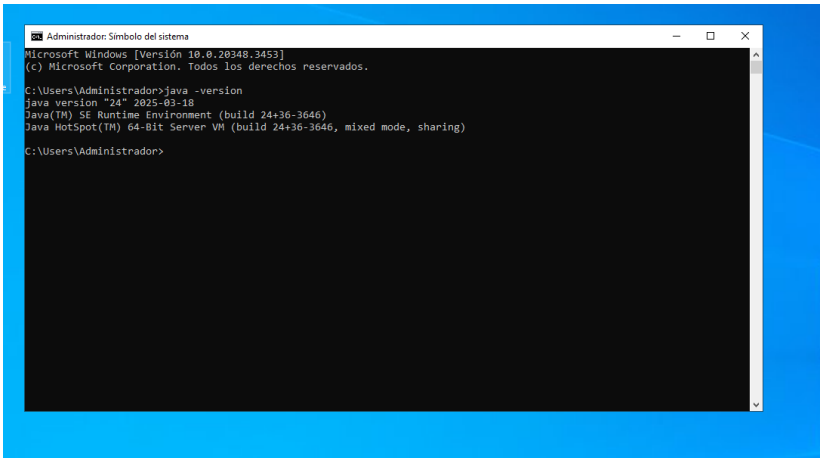


Configurando un Servidor Web

En este punto tenemos que editar la variable ***path,*** y crear nueva entrada en dicha variable.



Llegados a este punto, debemos probar la instalación para verificar que todo fue correctamente, para ello, abrimos un terminal (CMD) y ejecutamos el siguiente comando: `java -version`



Podemos comprobar que todo fue correcto, por lo que comenzamos a descargar e instalar *Apache Tomcat.* Como siempre vamos al sitio oficial para descargarlo.

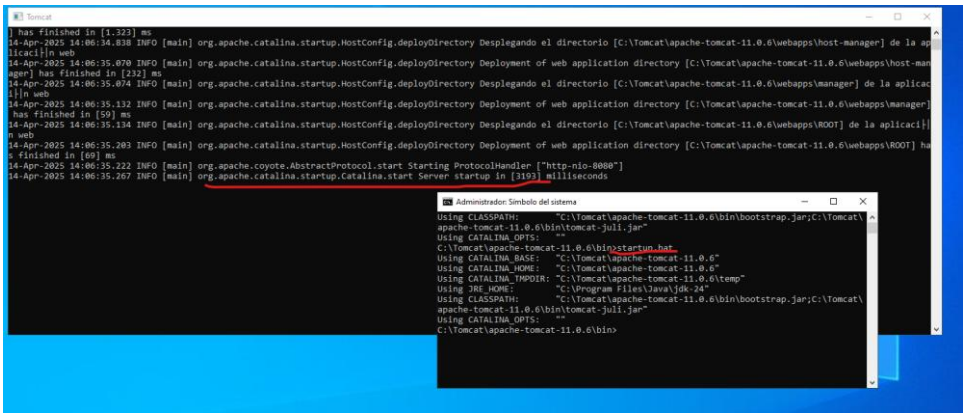
https://tomcat.apache.org/download-11.cgi

Configurando un Servidor Web

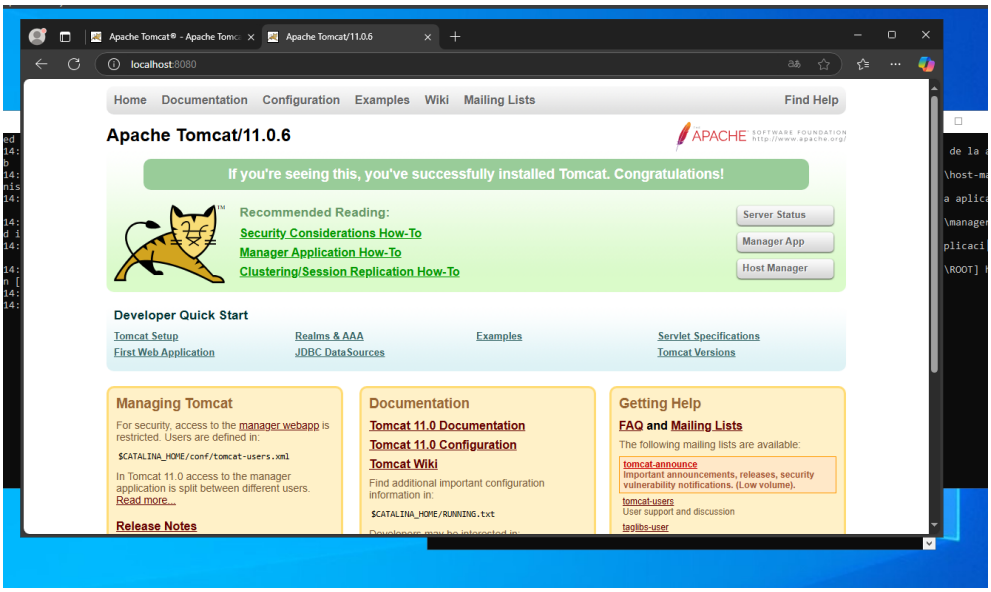
Descargamos el archivo Zip, en mi caso lo he descomprimido en una carpeta llamada ***Tomcat,*** ruta del archivo: *C:/Tomcat/apache-tomcat-11.0.6.*

Ahora debemos realizar el mismo paso que hicimos cuando instalamos el JDK, por lo que debemos añadir nueva entrada a la variable del sistema *path.*

En este punto, vamos a probar si hemos instalado y configurado todo correctamente, para ello abrimos una terminal, navegamos hasta el directorio del servidor Apache Tomcat, en nuestro caso: *C:\Tomcat\apache-tomcat-11.0.6\bin* y ejecutamos el comando ***startup.bat.***



Podemos ver que Tomcat funciona correctamente, ahora si vamos al navegador y ponemos ***http://localhost:8080*** veremos lo siguiente:



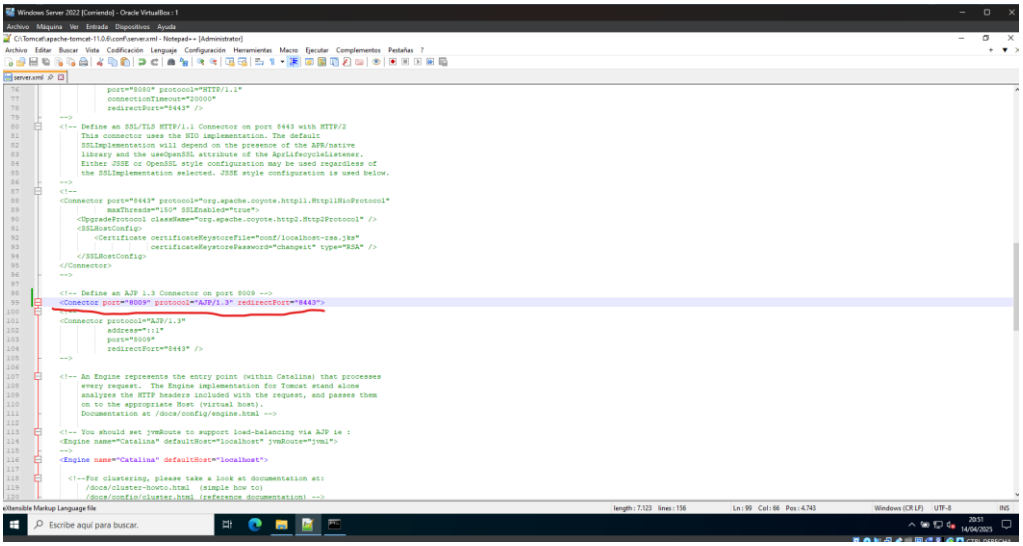
Confirmamos que funciona correctamente.

Configurando un Servidor Web

Ahora, para conectar ***Apache HTTP Server con Apache Tomcat***, necesitamos configurar un ***proxy inverso*** o usar un ***conector AJP (Apache JServ Protocol).***
Esto permite que ***Apache HTTP Server*** maneje las solicitudes ***HTTP*** y reenvíe las solicitudes específicas ***(como aplicaciones Java) a Tomcat***.

En nuestro caso hemos decidido usar un ***conector AJP,*** ya que le hemos preguntado a ***Qwen(IA)*** y dice que el protocolo AJP es más eficiente que HTTP para la comunicación entre Apache HTTP Server y Tomcat. Por ello el primer paso será configurar el conector AJP en Tomcat, para ello abriremos el archivo ***server.xml*** y buscamos la siguiente línea de código:

```
<Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />
```



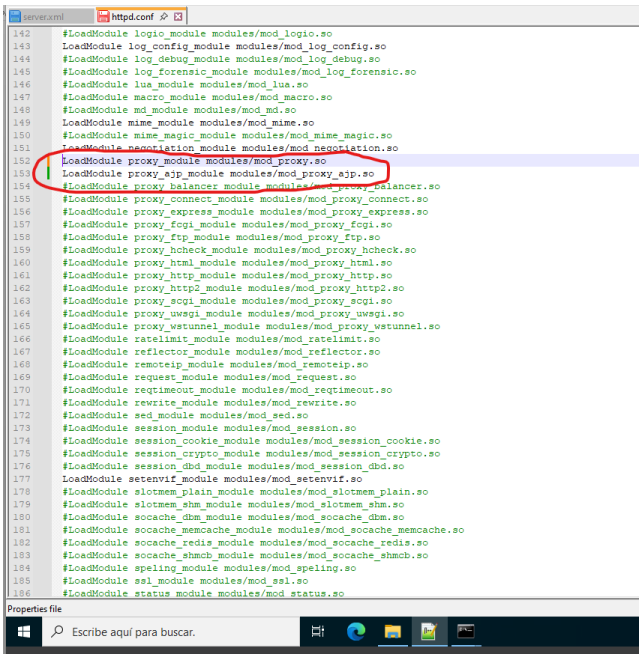
En mi caso particular tuve que añadir esa línea de código del fichero ya que no venía definida. Una vez realizado este paso debemos reiniciar el servicio de Tomcat.

Configurando un Servidor Web

El siguiente paso será habilitar el módulo AJP en Apache, para ello buscaremos el archivo de configuración principal de Apache, *httpd.conf* y descomentar las siguientes líneas de código:

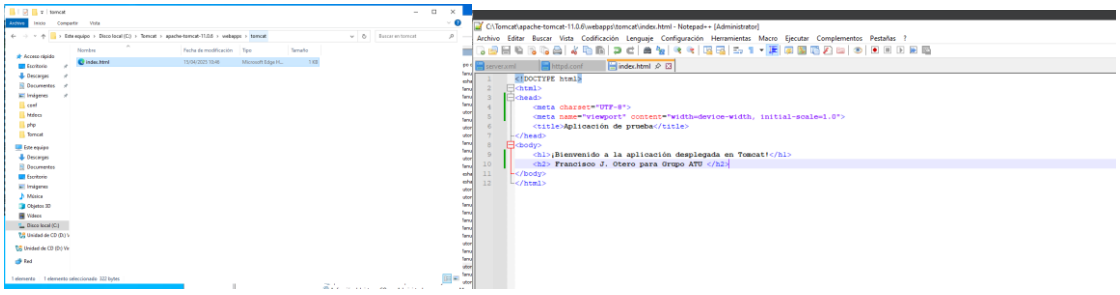
```
LoadModule proxy_module modules/mod_proxy.so

LoadModule proxy_ajp_module modules/mod_proxy_ajp.so
```



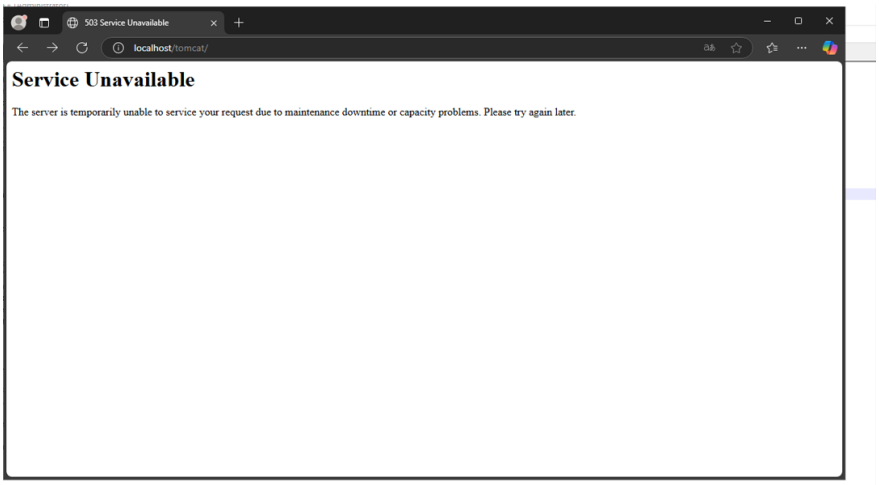
En este caso si venían dichas líneas de código, solo tuvimos que quitar el símbolo # para que dejara de ser un comentario, guardamos el fichero y como cada cambio realizado, debemos reiniciar el servicio de Apache.

Ahora es momento de probar si la configuración fue correcta, para ello abriremos un navegador y escribimos ***http://localhost/tomcat*** para probar si funciona. Antes de probar he creado una carpeta llamada ***tomcat*** dentro de la carpeta ***webapps,*** dentro de ella he añadido un fichero ***HTML*** sencillo para probar la comunicación entre Apache y Tomcat.

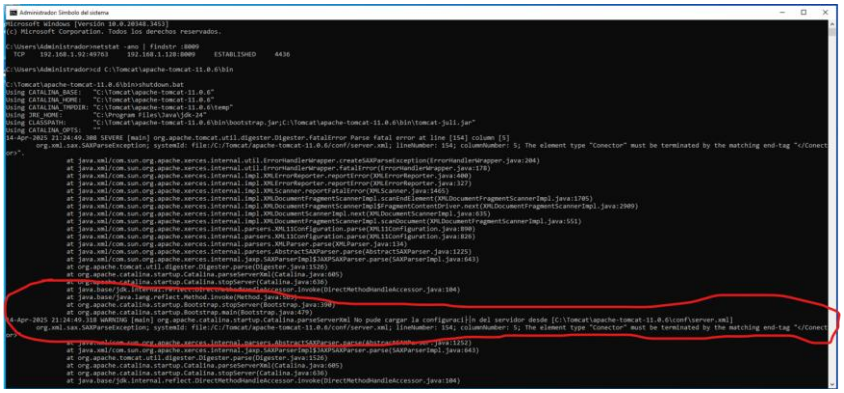


Configurando un Servidor Web

Es momento de probar la configuración.



Como se puede ver en la imagen anterior, la comunicación entre Apache y Tomcat no funciona. Tuvimos varios errores más aparte de este. Al intentar apagar Tomcat nos dio el siguiente error.



Este error muestra un error de sintaxis en la configuración del fichero de Tomcat ***server.xml,*** indica que hemos escrito mal la etiqueta ***"Connector".*** Hemos corregido más errores de sintaxis tanto en el fichero de Tomcat mencionado como en el fichero principal de configuración de Apache. Aun así, al probar de nuevo la comunicación entre ambos seguía fallando.

Tras revisar una y otra vez los ficheros de configuración encontré el error que impedía la comunicación, el error se encontraba en el fichero *server.xml* de Tomcat. En concreto en la etiqueta Connector ya mencionada:

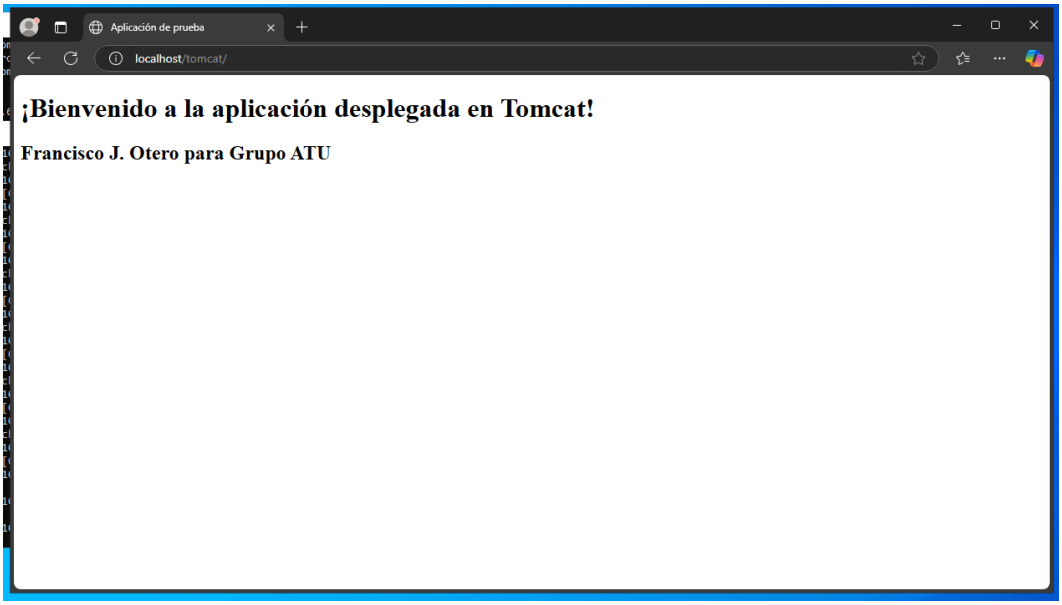
```
<Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />
```

Configurando un Servidor Web

Tras preguntar a ChatGPT que podría fallar en mi configuración me ofreció varios posibles fallos y uno de ellos fue el causante, en la etiqueta anterior debemos añadir un atributo más, que a partir de la ***version de Tomcat 9.0.31*** el conector AJP requiere el atributo ***secretRequired*** por seguridad. En nuestro caso lo dejamos así:

```
<Connector port="8009" protocol="AJP/1.3" redirectPort="8443"
secretRequired="false" />
```

Si le dejamos activado (true), debemos añadir una clave secreta y pasársela desde Apache. Lo más sencillo para pruebas locales es dejar sin activar dicho atributo (false). Una vez realizado este cambio, reiniciamos tanto Apache como Tomcat, probamos de nuevo, y en esta ocasión la comunicación funciona correctamente.



Tras varios intentos conseguimos que funcionara todo de la forma esperada, los errores más comunes bajo mi punto de vista son los errores en sintaxis.

Configurando un Servidor Web

6. Enlaces de interés.

https://www.osmosislatina.com/apache/modulos.htm

https://www.osmosislatina.com/apache/java.htm

https://www.rehurtado.com/ver_documento.php?id=29