

Mónica Veiga Puente

Módulo Despliegue de Aplicaciones Web

Plexus Technologies

21 de enero de 2021

Elección de un hosting, configuración de Apache y Tomcat9, instalación de un servidor SSH, conexión remota con SSH y FTP.

### 1.- Elección de un hosting: cuáles hay, características y cuál elegir.

Un hosting es un servicio por el cual se obtiene conexión a un servidor que puede ser de correo electrónico, de aplicaciones web o de discos virtuales entre otros. Se puede contratar el dominio como parte del hosting o por separado. Dependiendo de para qué queramos usarlo necesitaremos un tipo de hosting u otro. Algunos tipos de hosting que podemos barajar son los siguientes:

Gratuito: suelen ofrecer un subdominio de otro sitio más que un dominio propio, la calidad es baja y ofrece recursos limitados. Está bien para aprender o para mantener algún proyecto personal sin coste, pero no para elaborar algo más profesional o grande.

Compartido: con este servicio se comparten los recursos de software y de hardware del servidor con otros clientes. Aunque cada uno tenga su propio espacio, los recursos que se puedan usar serán un porcentaje de los del servidor. Está bien para uso personal y es económico, puede llegar a aguantar bastante tráfico y tiene muy buena disponibilidad, pero hay que tener cuidado pues muchos proveedores pueden aprovecharse de que es un hosting más barato y ofrecer muy pocos recursos, acabando con un sitio muy lento e inestable. Algunas ventajas del hosting compartido son que es fácil de manejar, flexible porque hay diversos planes, suele ofrecer soporte técnico, suele ofrecer multialojamiento y el proveedor se encarga de mantenerlo. Ahora bien, algunos proveedores no se preocupan de la seguridad, el que los recursos sean limitados implica que pueda haber inestabilidad,

VPS: Se trata de un servidor privado en forma de máquina virtual y ofrece diferentes gamas de recursos a diferentes precios. Es muy flexible porque permite cambiar de gama en cualquier momento, será más rápido por no tener que compartirse con otros clientes, y se pueden servir diferentes sitios con un solo hosting. Entre las desventajas se encuentra el precio, que aumenta hasta comenzar en 20 o 30 €/mes, y que un hosting básico puede tener menos prestaciones que uno compartido de mayor calidad, ya que el compartido no garantiza que esos recursos siempre estén disponibles y el privado sí. Además, la administración del servidor será compleja, aunque también ofrece mayor control el mismo. En este sentido, tiene una orientación más profesional que los anteriores, ya que lo que importa con éste no es que aguante más tráfico sino que se tenga mayor control sobre él.

Especializado o gestionado: se trata de un hosting que ofrece un servicio completo: además de ofrecer el servidor también pone a disposición del cliente a su servicio técnico para que aseguren que los sitios funcionan. Entre sus ventajas se encuentran que liberan de trabajo al cliente, ya que no se necesita realizar un gran proceso de instalación, mantenimiento y seguridad. Al ser especialistas de la aplicación, garantizan que un mejor funcionamiento, mayor estabilidad y seguridad. El mayor problema de estos es el precio, que dependiendo del tráfico que se tenga puede ser de 22€ o de 75€.

Cloud Hosting: Como indica el nombre, se trata de un servicio en la nube que reparte sus recursos entre diferentes máquinas. Se trata de un servicio seguro y flexible, y el precio puede ser más asequible que en otros casos, aunque el precio de entrada va a ser mayor que el del hosting compartido.

A la hora de decantarse por uno u otro habrá que plantearse primeramente el tamaño del proyecto: si se trata de un negocio con el que se pretenden recibir muchas visitas y clientes será más efectivo usar uno que elimine la posibilidad de que se caiga o se ralentice el servidor: uno spv o uno gestionado, o incluso uno compartido de mayor gama que ofrezca buenas prestaciones. Si se trata de una tienda online y no se tiene mucho tiempo el gestionado será más indicado, pero si se tiene tiempo y conocimientos técnicos para dedicarle uno spv resultará muy apropiado, especialmente si se quieren alojar varios sitios diferentes.

En el caso de estar buscando un hosting para realizar un proyecto personal, si tiene una vertiente profesional (crear un portfolio o un blog personal, por ejemplo), deberemos elegir dependiendo del tráfico que tengamos. Si tenemos intención de crear varios y dedicarle mucho tiempo, o si tenemos conocimientos técnicos podremos usar un VPS, si queremos buenos recursos y no tenemos mucho tiempo nos irá mejor contratar uno gestionado. Sin embargo, en caso de que no esperemos tener a muchísimos usuarios conectados al mismo tiempo, un hosting compartido puede ser el ideal, ya que nos saldrá más barato, no tendremos que realizar nosotros el mantenimiento, y debería ser capaz de dar un buen servicio.

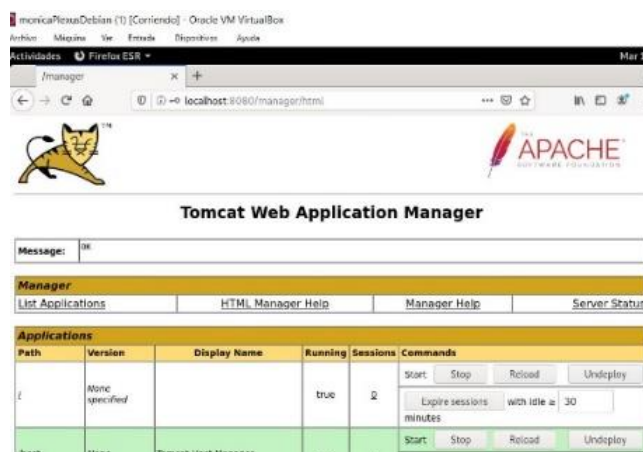
Si solo queremos realizar un proyecto con el que aprender o practicar, el hosting gratuito va a ser el idóneo.

2.- Desde 0: configuración de Tomcat9 para acceder a la web, despliegue de aplicación con Tomcat, configuración de proxies en Apache, instalación de un servidor SSH.

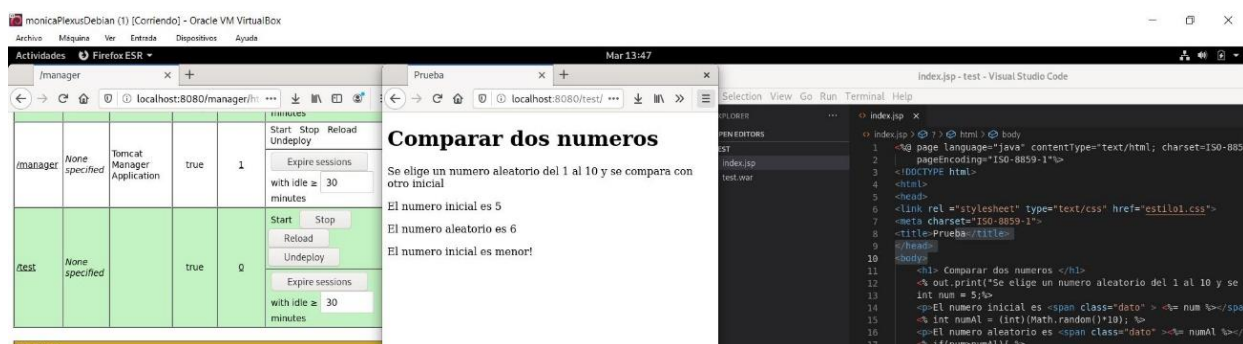
Primero instalo apache tomcat 9 con `apt-get install tomcat9` y el *manager* con `apt-get install tomcat9-admin`, configuro correctamente el *path* al archivo con los usuarios en `server.xml` y configuro el acceso web a tomcat9 declarando roles y usuarios en `tomcat-users.xml`:

```
monica@debian: /usr/lib/jvm/java-11-openjdk-amd64/bin
Ficheiro  Editar  Ver  Buscar  Terminal  Axuda
GNU nano 3.2      /etc/tomcat9/tomcat-users.xml

<user username="monica" password="abc123.." roles="admin-gui,manager-gui"/>
</tomcat-users>
```



Ahora voy a desplegar una aplicación en Tomcat. Para ello descargo VSCode en mi máquina Debian, creo un index.jsp, compruebo si tengo java instalado con `java -version`, descargo el jdk de Java con `apt install default-jdk` y configuro las variables de entorno con `export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64` y `export PATH=$PATH:/usr/lib/jvm/java-11-openjdk-amd64/bin`. Para desplegar el proyecto en Tomcat necesito generar el archivo war. Lo hago con la extensión de Tomcat for Java, voy al tomcat manager y selecciono el war para desplegarlo, accedo desde el navegador a la aplicación y vemos que ha funcionado:



Una vez he desplegado la aplicación en Tomcat, voy a configurar un Virtual Host para poder acceder a la aplicación a través de él. Primero habilito los proxies http, ajp y proxy con `a2enmod`:

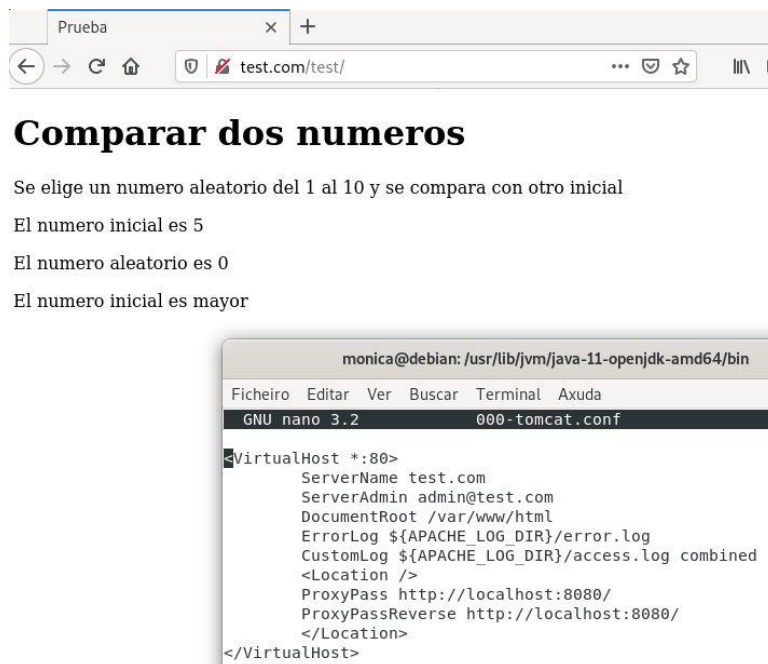
```

monica@debian: /usr/lib/jvm/java-11-openjdk-amd64/bin

Ficheiro Editar Ver Buscar Terminal Axuda
root@debian:/etc/apache2/sites-available# nano 000-tomcat.conf
root@debian:/etc/apache2/sites-available# a2enmod proxy
Enabling module proxy.
To activate the new configuration, you need to run:
    systemctl restart apache2
Enabling module proxy_ajp.
To activate the new configuration, you need to run:
    systemctl restart apache2
root@debian:/etc/apache2/sites-available# a2enmod proxy_http
Considering dependency proxy for proxy_http:
Module proxy already enabled
Enabling module proxy_http.
To activate the new configuration, you need to run:
    systemctl restart apache2
root@debian:/etc/apache2/sites-available# systemctl restart tomcat9
  
```

Además, he creado un Virtual Host llamado `000-tomcat.conf` donde he declaro la directiva `Location` y dentro el `ProxyPass` y el `ProxyPassReverse` para acceder

desde test.com al puerto 8080 en el que está siendo desplegado el war.<sup>1</sup> Vemos que ha funcionado:



Para deshabilitar el conector para el puerto 8080 de Tomcat he comentado las secciones del conector. Sin embargo, como lo siguiente que vamos a hacer es instalar un servidor SSH, luego lo descomentaré y cambiaré el puerto de escucha por el 443. Para instalar el servidor SSH, primero voy a crear la carpeta ssh y una vez dentro voy a generar un fichero y una clave con

```
keytool -genkey -alias nomedominio -keyalg RSA -keysize 2048 -keystore key.jks
```

y ahora genero el certificado autofirmado con

```
keytool -certreq -alias tomcat -keyalg RSA -file cert.csr -keystore key.jks
```

Para habilitar el SSL en Tomcat iremos al fichero server.xml y añadiremos lo siguiente al Connector Port, además de establecer el 443 como puerto de escucha:

---

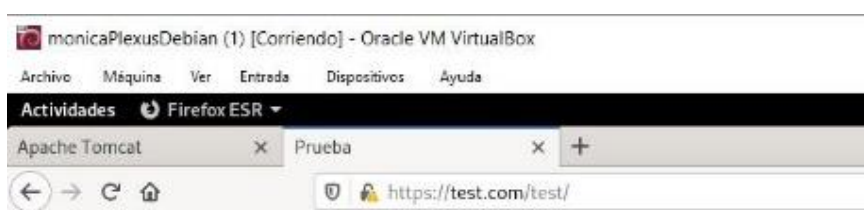
<sup>1</sup> También podríamos eliminar las etiquetas de Location y añadir una barra lateral / después de cada declaración del proxy.

```

monica@debian: /usr/lib/jvm/java-11-openjdk-amd64/bin
Ficheiro Editar Ver Buscar Terminal Axuda
GNU nano 3.2 /etc/tomcat9/server.xml
<Connector port="443" protocol="HTTP/1.1"
    connectionTimeout="20000"
    redirectPort="8443"
    SSLEnabled="true"
    scheme="https"
    keystoreFile="/etc/tomcat9/ssh/key.jks"
    keystorePass="abc123.."
    clientAuth="false"
    sslprotocol="TLS"

```

Probamos y funciona al introducir https:// para acceder. Si no se usa el protocolo https y se intenta usar http derivará en error, puesto que lo hemos configurada para que la conexión vaya por TLS. Vemos que funciona:



## Comparar dos numeros

Se elige un numero aleatorio del 1 al 10 y se compara con otro inicial

El numero inicial es 5

El numero aleatorio es 1

El numero inicial es mayor

3.- Desde 0: instalación de Apache y Tomcat, configuración de Tomcat9 para desplegar aplicaciones, creación de un host para el dominio monicaveigapiente.com, establecimiento de proxies, cambiar conector del puerto 8080, despliegue de aplicaciones subidas por SSH y FTP, cambios a la manager-app de tomcat y despliegue de aplicación a través de apache en monicaveigapiente.com y de tomcat utilizando servidor FTP.

Ahora, empezando desde el principio, repetimos los primeros pasos en una nueva máquina virtual: instalamos Apache y Tomcat9, y configuramos el acceso web de los usuarios:

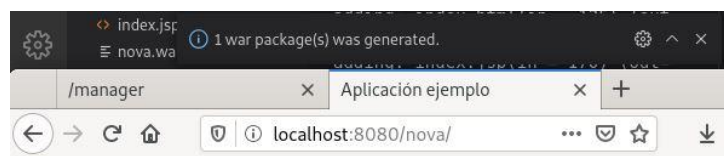


```

monica@debian: ~
Ficheiro Editar Ver Buscar Terminal Axuda
GNU nano 3.2 /etc/tomcat9/tomcat-users.xml
version="1.0">
<user username="monica" password="abc123.." roles="manager-gui,admin-gui"/>
</tomcat-users>

```

Preparo una aplicación, descargo Visual Studio Code y el default-jdk, asigno las variables JAVA\_HOME y PATH<sup>2</sup> y despliego el war generado con VSCode en tomcat:



## Título

Lorem ipsum...

Ahora creo un host virtual en Apache para el dominio monicaveigapunte.com:



```

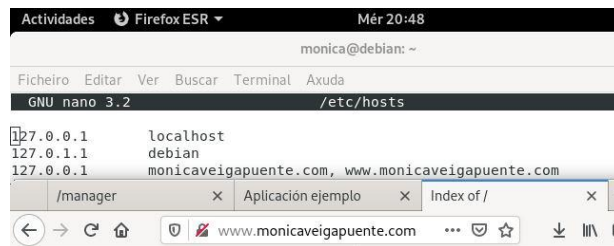
monica@debian: ~
Ficheiro Editar Ver Buscar Terminal Axuda
GNU nano 3.2 /etc/apache2/sites-available/000-mvp.conf
<VirtualHost *:80>
    ServerName monicaveigapunte.com
    ServerAlias www.monicaveigapunte.com
    ServerAdmin monica@monicaveiga.com
    DocumentRoot /var/www/html/mvp
    ErrorLog /var/www/html/mvp/error.log
    CustomLog /var/www/html/mvp/access.log combined
</VirtualHost>

```

Una vez hecho esto, habilito el sitio con a2ensite, lo declaro en hosts, creo la carpeta en la que voy a alojar el sitio, los ficheros de log que he declarado, y compruebo que funciona. Vemos que sí, ya que no le he creado un index.html aún:

<sup>2</sup> Con `export JAVA_HOME="/usr/lib/jvm/java-1.11.0-openjdk-amd64"` y `export PATH=$PATH:$JAVA_HOME/bin`



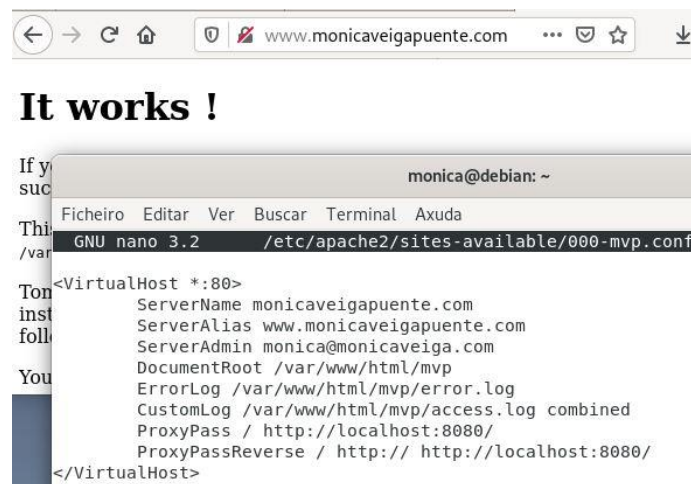


## Index of /

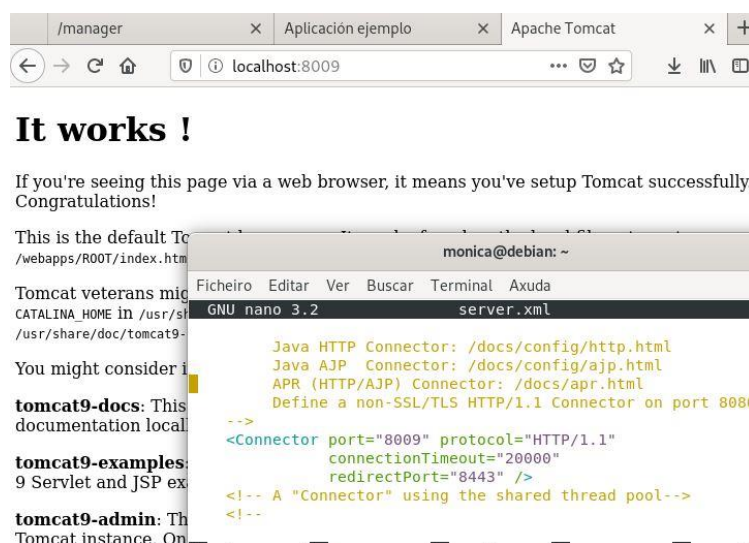
Name	Last modified	Size	Description
<a href="#">access.log</a>	2021-01-20 20:45	297	
<a href="#">error.log</a>	2021-01-20 20:45	1.9K	

Apache/2.4.38 (Debian) Server at www.monicaveigapiente.com Port 80

Ahora habilito los módulos de los proxies http y ajp con `a2enmod proxy_ajp` y `a2enmod proxy_http` y modifico la configuración para usarlos:



Seguidamente, cambio el conector para el puerto predeterminado de Tomcat a otro, como el 8009, lo pruebo:





El siguiente paso es subir remotamente por SSH al servidor la aplicación. Para esto, se puede hacer una copia segura remota desde la carpeta actual de la aplicación en `var/www/html` al directorio principal del servidor de tomcat en `/var/lib/tomcat9`. Para ello, primero genero el archivo war y después, utilizando la sintaxis del comando `scp` declaramos el archivo de origen y el de destino:



Para esto he usado solo el cliente de SSH, pero si quiero subir mi app remotamente con SSH desde otro equipo, hay que descargarse el `ssh-server`, lo que se puede hacer con `apt-get install openssh-server` ii

Para hacer la copia remota, antes de usar `scp` vamos a conectar los dos equipos a través de `ssh` introduciendo `ssh monica@127.0.0.1 -p 2222`, ya que previamente había configurado el 2222 como puerto de `ssh` en mi máquina virtual:



Ahora salgo de la sesión y ya puedo utilizar `scp` para enviar el archivo remotamente de un equipo a otro:

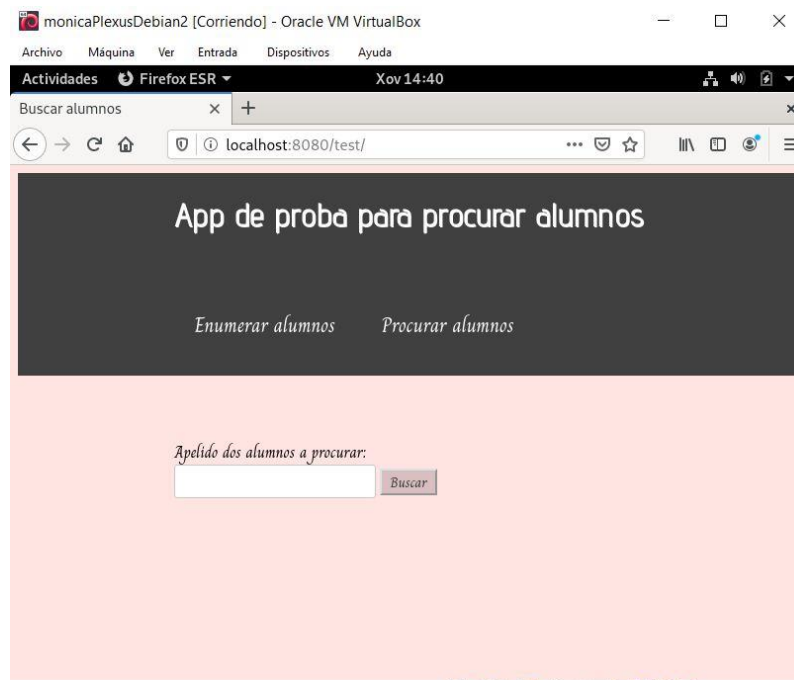
```
scp -P 2222
C:\Users\monic\Documents\1CServidor\3_3_gardarAlumnoImaxe.war
monica@127.0.0.1:/home/monica/Documentos/test.war
```

Ya he pasado desde mi anfitriona a la MV el archivo war. Como la he enviado al directorio Documentos para evitar problemas de permisos sin tener que suavizarlos,

desde la consola de debian uso scp para mover el archivo a la carpeta principal del servidor de webapps de tomcat:

```
scp /home/monica/Documentos/test.war /var/lib/tomcat9/webapps/test.war
```

Pruebo a conectarme y vemos que funciona:



El siguiente punto consiste en instalar un servidor FTP. Voy a instalar ProFTPD (también podríamos haber instalado otro, como Vsftpd). Para instalar cualquiera de ellos usaremos `apt-get install` seguido del nombre del programa. Si no tenemos instalado FTP también habrá que hacerlo de la misma forma. Luego voy a entrar en `/etc/proftpd/proftpd.conf` y voy a añadir las siguientes líneas para enjaular a los usuarios a su directorio de inicio:

```
monica@debian: ~
Ficheiro  Editar  Ver  Buscar  Terminal  Axuda
GNU nano 3.2 /etc/proftpd/proftpd.conf

#Virtual Users
DefaultRoot ~
RequireValidShell off
AuthUserFile /etc/proftpd/ftpd.passwd
```

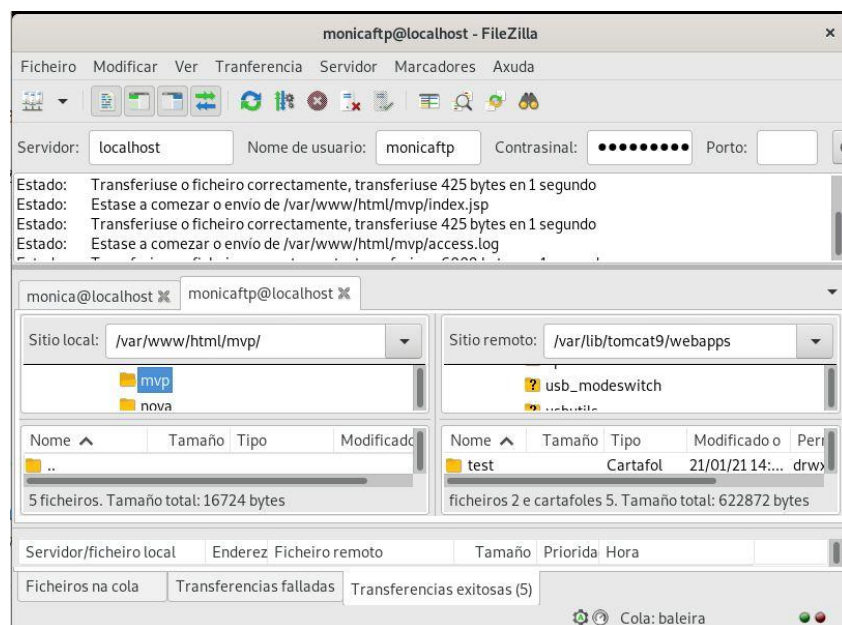
Seguidamente, voy a crear un usuario con `adduser monicaftp`, un usuario virtual con `ftpasswd` y una carpeta de inicio a la que le cambio el propietario y los permisos:

```

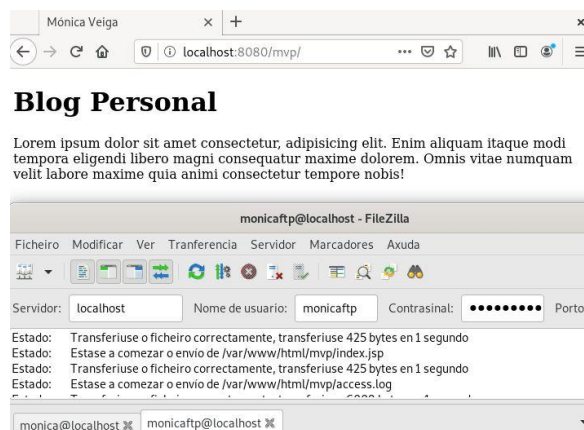
monica@debian: ~
Ficheiro Editar Ver Buscar Terminal Axuda
root@debian:/# ftpasswd -passwd --uid=1089 --name=monicaFtp --home=/srv/ftp/monicaFtp --
shell=/bin/false
ftpasswd: --passwd: missing --gid argument: default gid set to uid
ftpasswd: creating passwd entry for user monicaFtp
ftpasswd: /bin/false is not among the valid system shells. Use of
ftpasswd: "RequireValidShell off" may be required, and the PAM
ftpasswd: module configuration may need to be adjusted.
Password:
Re-type password:
ftpasswd: entry created
root@debian:/# mkdir /srv/ftp/monicaFtp
root@debian:/# cd /srv/ftp/monicaFtp
root@debian:/srv/ftp/monicaFtp# cd ..
root@debian:/srv/ftp# chown 1089 monicaFtp
root@debian:/srv/ftp# chmod 700 monicaFtp

```

Me descargo un cliente de FTP, en este caso Filezilla, me conecto y transfiero el directorio de la aplicación de mi interés (con el war ya generado en su interior) desde el sitio local al sitio remoto con éxito:



Entro en la dirección localhost:8080/mvp y compruebo que ha funcionado, vemos que sí:



Ahora hay que deshabilitar el acceso al tomcat manager, no obstante, éste ya está deshabilitado por defecto, y solo los usuarios establecidos en el fichero Tomcat-Users tienen acceso a la app. En este caso, solo hay una administradora principal definida:

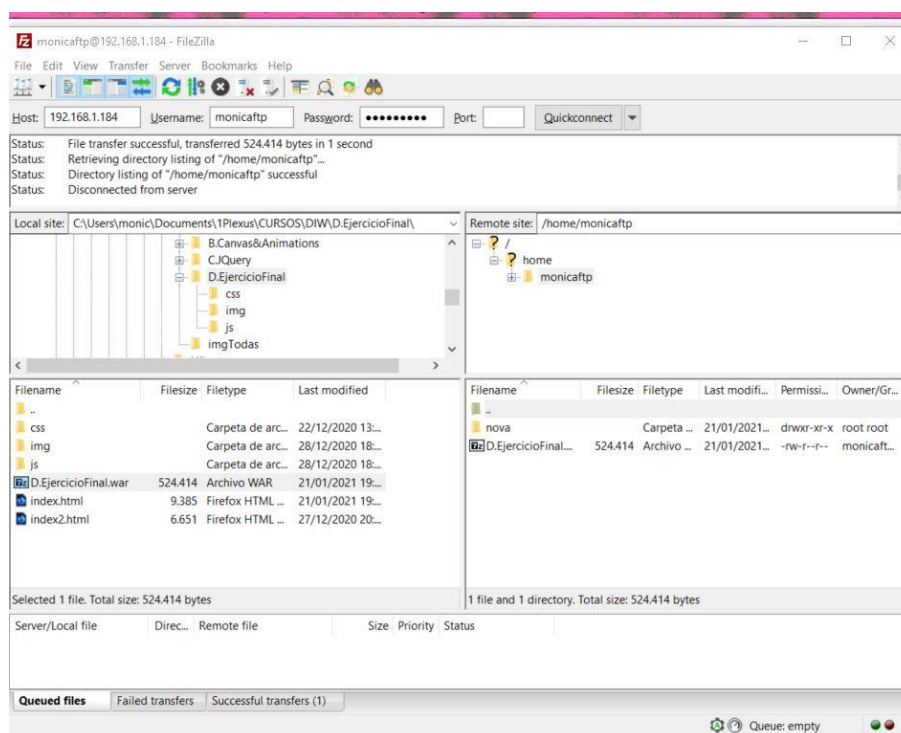
```
monica@debian: ~
Ficheiro Editar Ver Buscar Terminal Axuda
GNU nano 3.2 /etc/tomcat9/tomcat-users.xml Modific
<user username="monica" password="abc123.." roles="manager-gui,admin-gui"/>
</tomcat-users>
```

Otra posibilidad sería habilitar y restringir el acceso a unas determinadas ips. Para ello, en el archivo de context.xml puedo añadir en la etiqueta Context lo siguiente:

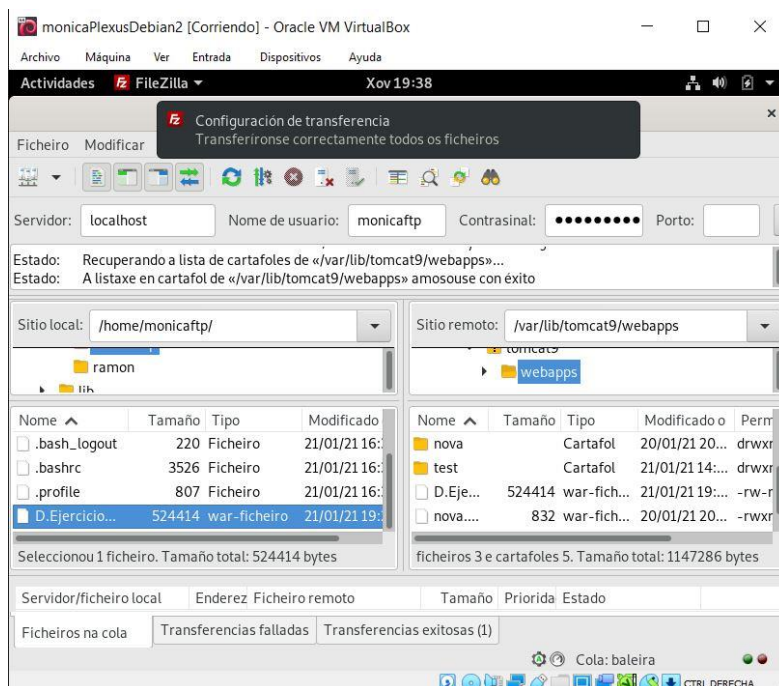
```
<Context antiResourceLocking="false" privileged="true">
  <Valve className="org.apache.catalina.valves.RemoteAddrValve"
    allow="127\.0\.0\.1"/>
</Context>
```

Ésta será otra forma para que sólo deje entrar a la IP que considere, y se puede permitir o denegar el acceso a una IP con allow y deny.

Ahora voy a desplegar otra aplicación web diferente, por una parte la enviaré al servidor de tomcat con ftp y por otra la desplegaré con apache como monicaveigapunte.com. De esta vez genero el war de un ejercicio anterior y, conectándome a la ip de mi máquina virtual desde la anfitriona a través de Filezilla con el protocolo ftp me envió el war:



Una vez está en mi máquina virtual, la vuelvo a pasar desde la carpeta de inicio a la del servidor por FTP:



Le cambio el nombre y los permisos y accedo desde el navegador con tomcat:



Finalmente, debemos crear un host virtual para desplegarla con apache en el dominio monicaveigapuerto.com. Pongo el proxy apuntando al directorio que interesa y habilito el sitio. Vemos que ha funcionado:

monicaPlexusDebian2 [Corriendo] - Oracle VM VirtualBox

Archivo Máquina Ver Entrada Dispositivos Ayuda

Actividades **Terminal** Xov 20:35

Agrotienda Sacha Green x Agrotienda Sacha Green x +

← → ↻ 🏠 [www.monicaveigapuate.com](http://www.monicaveigapuate.com) ... 🛡️ ☆




<b>Semillas de caléndula</b>	<b>Semillas de girasol</b>
2.5€	1.5€

```

monica@debian: ~
Ficheiro Editar Ver Buscar Terminal Axuda
GNU nano 3.2 /etc/apache2/sites-available/pg.conf

<VirtualHost *:80>
    ServerName monicaveigapuate.com
    ServerAlias www.monicaveigapuate.com
    ServerAdmin monica@monicaveiga.com
    DocumentRoot /var/www/html/pinkgreen
    ErrorLog /var/www/html/pinkgreen/error.log
    CustomLog /var/www/html/pinkgreen/access.log combined
    ProxyPass / http://localhost:8080/pinkgreen/
    ProxyPassReverse / http:// http://localhost:8080/pinkgreen/
</VirtualHost>

```