

PRACTICA 1

Instalación y Configuración de Servidores de Internet: Apache Tomcat

Realizado por:

Alberto González Martínez 09072311F

1. Instalación de Apache-Tomcat

Una vez instalado Ubuntu 20.04 en nuestra máquina virtual, tendremos que instalar java y, posteriormente la versión 9.0.37 de Apache-Tomcat. Los pasos realizados se muestran a continuación.

1.1 Instalación de Java

Primero actualizamos y comprobamos si java está instalado en nuestra máquina virtual. Como vemos en la segunda imagen java no está instalado en la máquina virtual con lo que procedemos a su instalación.

```
alberto@ubuntu: ~  
alberto@ubuntu:~$ sudo apt update  
[sudo] password for alberto:  
Get:1 http://security.ubuntu.com/ubuntu focal-security InRelease [107 kB]  
Hit:2 http://us.archive.ubuntu.com/ubuntu focal InRelease  
Get:3 http://dl.google.com/linux/chrome/deb stable InRelease [1,811 B]  
Get:4 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease [111 kB]  
Get:5 http://security.ubuntu.com/ubuntu focal-security/main i386 Packages [140 kB]  
Get:6 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [548 kB]  
Get:7 http://security.ubuntu.com/ubuntu focal-security/main arm64 Packages [548 kB]  
Get:8 http://security.ubuntu.com/ubuntu focal-security/main armhf Packages [548 kB]  
Fetched 1,418 kB in 1s (1,418 kB/s)  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
16 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

```
alberto@ubuntu:~$ java -version  
Command 'java' not found, but can be installed with:  
  
sudo apt install openjdk-11-jre-headless # version 11.0.8+10-0ubuntu1~20.04, or  
sudo apt install default-jre # version 2:1.11-72  
sudo apt install openjdk-8-jre-headless # version 8u265-b01-0ubuntu2~20.04  
sudo apt install openjdk-13-jre-headless # version 13.0.3+3-1ubuntu2  
sudo apt install openjdk-14-jre-headless # version 14.0.1+7-1ubuntu1
```

A continuación, instalamos y comprobamos que la instalación se haya realizado correctamente.

```
alberto@ubuntu:~$ sudo apt install default-jre  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following additional packages will be installed:  
  ca-certificates-java default-jre-headless fonts-dejavu-extra java-common libatk-wrapper-java libatk-wrapper-java-jni openjdk-11-jre  
  openjdk-11-jre-headless  
Suggested packages:  
  fonts-ipafont-gothic fonts-ipafont-mincho fonts-wqy-microhei | fonts-wqy-zenhei  
The following NEW packages will be installed:  
  ca-certificates-java default-jre default-jre-headless fonts-dejavu-extra java-common libatk-wrapper-java libatk-wrapper-java-jni openjdk-11-jre
```

```
alberto@ubuntu:~$ java -version  
openjdk version "11.0.8" 2020-07-14  
OpenJDK Runtime Environment (build 11.0.8+10-post-Ubuntu-0ubuntu120.04)  
OpenJDK 64-Bit Server VM (build 11.0.8+10-post-Ubuntu-0ubuntu120.04, mixed mode, sharing)
```

```
alberto@ubuntu:~$ sudo apt install default-jdk  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done
```

```
alberto@ubuntu:~$ javac -version  
javac 11.0.8
```

1.2 Instalación de Apache y configuración del firewall

Instamos Apache, como se muestra a continuación.

```
alberto@ubuntu:~$ sudo apt install apache2
[sudo] password for alberto:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
```

Una vez instalado Apache, vemos que tenemos varios perfiles de Apache, siendo el más restrictivo el perfil Apache, el cual habilitaremos para permitir el tráfico.

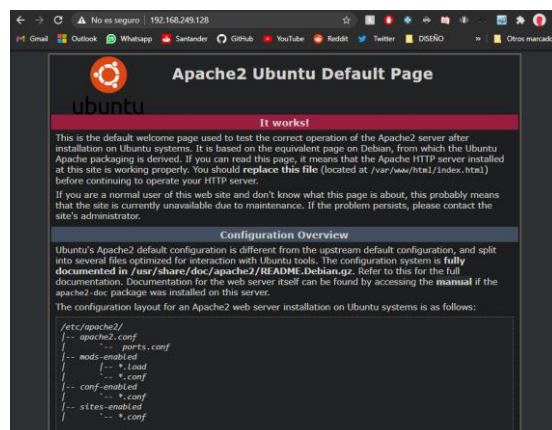
```
alberto@ubuntu:~$ sudo ufw app list
Available applications:
  Apache
  Apache Full
  Apache Secure
  CUPS
alberto@ubuntu:~$ sudo ufw allow 'Apache'
Rules updated
Rules updated (v6)
alberto@ubuntu:~$ sudo ufw status
Status: inactive
alberto@ubuntu:~$ sudo ufw enable
Firewall is active and enabled on system startup
```

Como vemos en la siguiente imagen están son la reglas creadas en nuestro firewall.

```
alberto@ubuntu:~$ sudo ufw status
Status: active

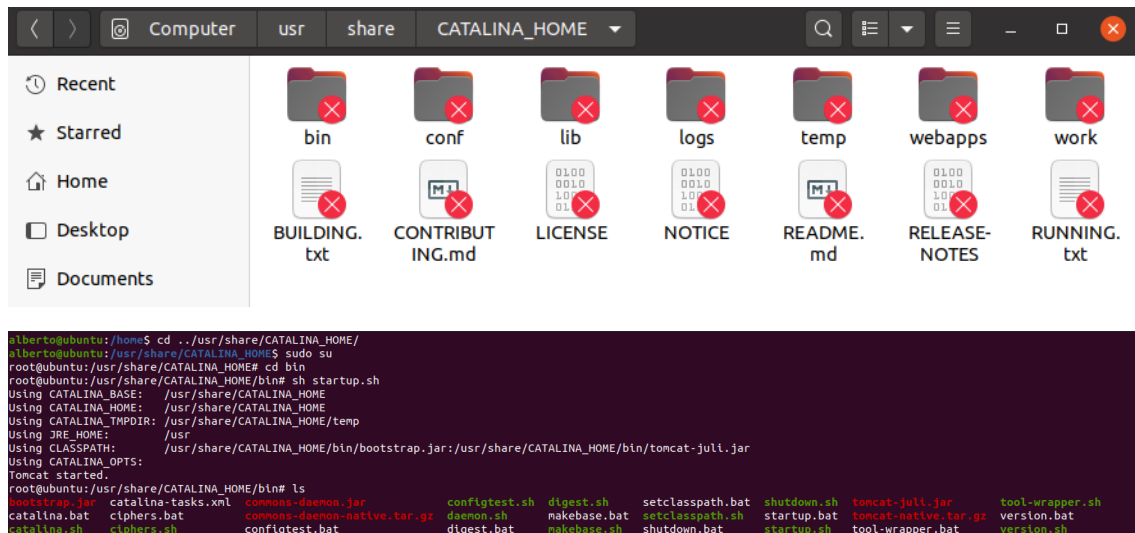
To Action From
--
Apache ALLOW Anywhere
80 ALLOW Anywhere
8080 ALLOW Anywhere
Apache (v6) ALLOW Anywhere (v6)
80 (v6) ALLOW Anywhere (v6)
8080 (v6) ALLOW Anywhere (v6)
```

Finalmente accedemos a nuestro servidor, mediante la ip de nuestra máquina o mediante localhost.

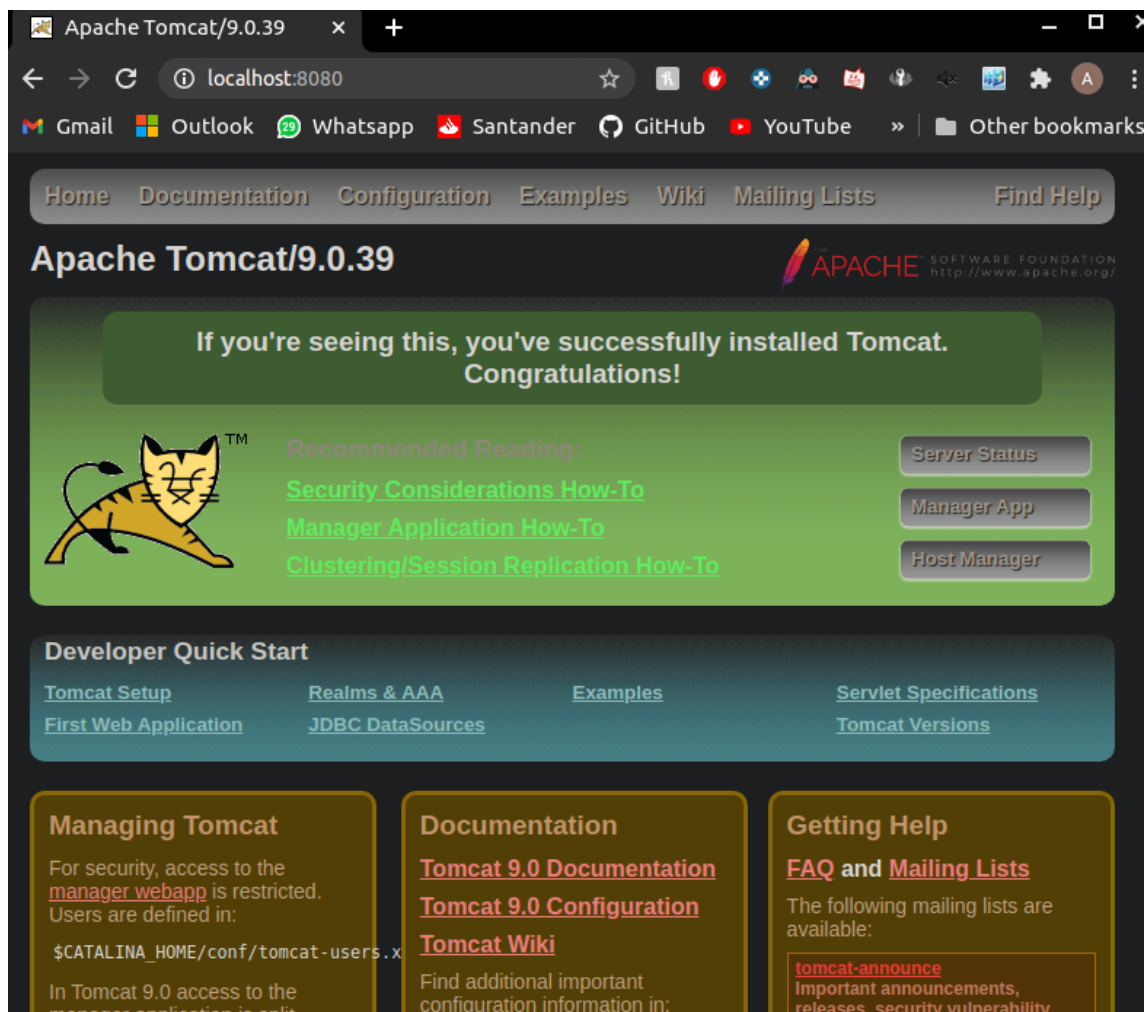


1.3 Instalación de Tomcat

Una vez tenemos instalado Apache, pasamos a instalar Tomcat. Para ello previamente tenemos que crear la carpeta CATALINA_HOME en el directorio /usr/share y descomprimir el archivo tar.gz en la carpeta CATALINA_HOME, una vez hecho esto, iniciamos el servidor.



Comprobamos que la instalación de Tomcat ha sido correcta.



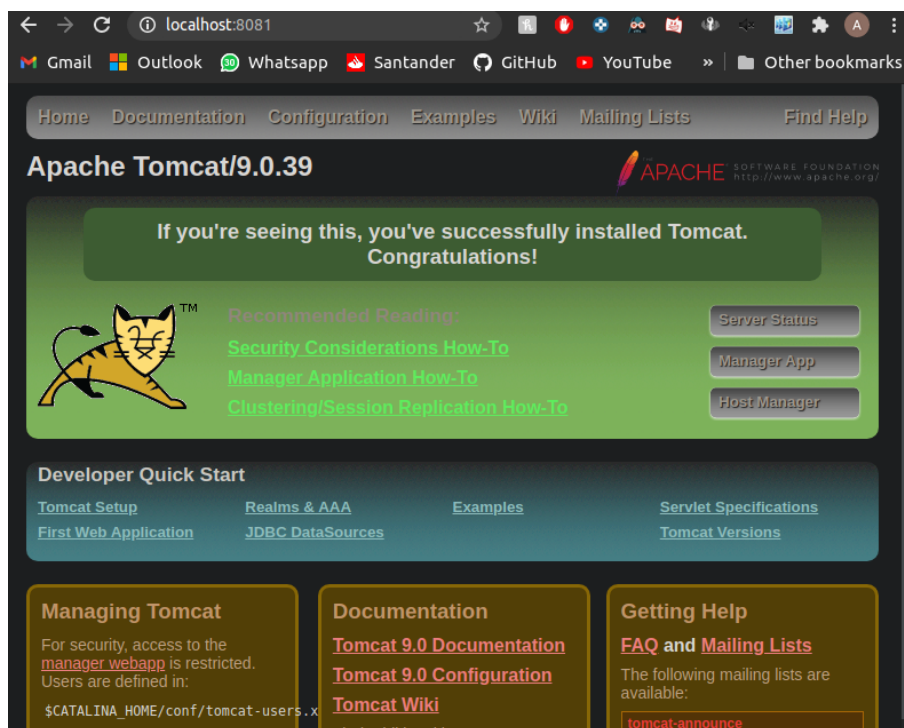
2. Configuración básica

Para modificar el puerto de comunicación de nuestro servidor del puerto 8080 al 8081, para ello tenemos que parar el servidor Apache-Tomcat y modificar el archivo server.xml. Una vez realizado esto guardamos los cambios, creamos una nueva regla de firewall y iniciamos de nuevo el servidor.

```
68 -->
69 <Connector port="8081" protocol="HTTP/1.1"
70         connectionTimeout="20000"
71         redirectPort="8443" />
72 <!-- A "Connector" using the shared thread pool-->
73 <!--
74 <Connector executor="tomcatThreadPool"
75             port="8080" protocol="HTTP/1.1"
76             connectionTimeout="20000"
77             redirectPort="8443" />
```

```
root@ubuntu:/usr/share/CATALINA_HOME/bin# sh shutdown.sh
Using CATALINA_BASE:   /usr/share/CATALINA_HOME
Using CATALINA_HOME:   /usr/share/CATALINA_HOME
Using CATALINA_TMPDIR: /usr/share/CATALINA_HOME/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /usr/share/CATALINA_HOME/bin/bootstrap.jar:/usr/share/CAT
ALINA_HOME/bin/tomcat-juli.jar
Using CATALINA_OPTS:
NOTE: Picked up JDK_JAVA_OPTIONS:  --add-opens=java.base/java.lang=ALL-UNNAMED -
-add-opens=java.base/java.io=ALL-UNNAMED --add-opens=java.rmi/sun.rmi.transport=
ALL-UNNAMED
root@ubuntu:/usr/share/CATALINA_HOME/bin# sh startup.sh
Using CATALINA_BASE:   /usr/share/CATALINA_HOME
Using CATALINA_HOME:   /usr/share/CATALINA_HOME
Using CATALINA_TMPDIR: /usr/share/CATALINA_HOME/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /usr/share/CATALINA_HOME/bin/bootstrap.jar:/usr/share/CAT
ALINA_HOME/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
```

Comprobamos que el servidor esta activo y en el puerto que hemos indicado.



3. Gestión de aplicaciones

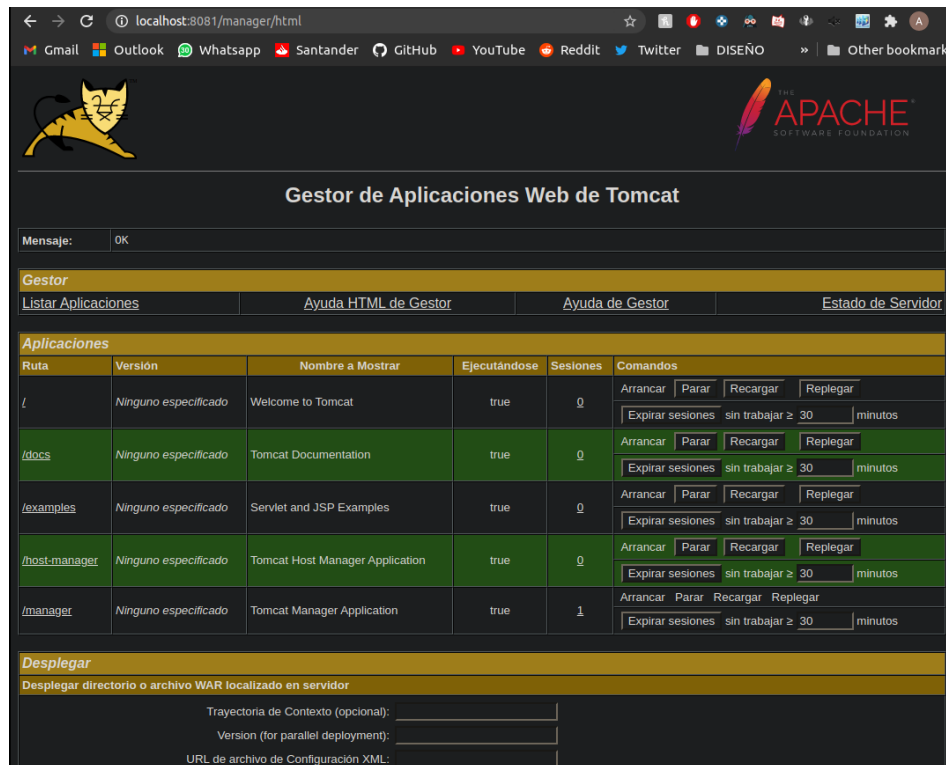
Añadimos un usuario para poder acceder a tomcat manager, para ello añadimos en el archivo tomcat-users.xml lo que se ve en la siguiente imagen.

```
18 <tomcat-users xmlns="http://tomcat.apache.org/xml"
19               xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
20               xsi:schemaLocation="http://tomcat.apache.org/xml tomcat-users.xsd"
21               version="1.0">
22 <user username="alberto" password="admin" roles="admin-gui,manager-gui"/>
```

Una vez realizados esos cambios, reiniciamos el servidor.

```
root@ubuntu:/usr/share/CATALINA_HOME/bin# sh shutdown.sh
Using CATALINA_BASE:   /usr/share/CATALINA_HOME
Using CATALINA_HOME:   /usr/share/CATALINA_HOME
Using CATALINA_TMPDIR: /usr/share/CATALINA_HOME/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /usr/share/CATALINA_HOME/bin/bootstrap.jar:/usr/share/CAT
ALINA_HOME/bin/tomcat-juli.jar
Using CATALINA_OPTS:
NOTE: Picked up JDK_JAVA_OPTIONS:  --add-opens=java.base/java.lang=ALL-UNNAMED -
--add-opens=java.base/java.io=ALL-UNNAMED --add-opens=java.rmi/sun.rmi.transport=
ALL-UNNAMED
root@ubuntu:/usr/share/CATALINA_HOME/bin# sh startup.sh
Using CATALINA_BASE:   /usr/share/CATALINA_HOME
Using CATALINA_HOME:   /usr/share/CATALINA_HOME
Using CATALINA_TMPDIR: /usr/share/CATALINA_HOME/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /usr/share/CATALINA_HOME/bin/bootstrap.jar:/usr/share/CAT
ALINA_HOME/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
```

Finalmente, ya podremos acceder al Gestor de Aplicaciones Web de Tomcat, habiendo introducido antes el usuario y contraseñas especificadas anteriormente.



localhost:8081/manager/html

Gestor de Aplicaciones Web de Tomcat

Mensaje: OK

Gestor

Listar Aplicaciones Ayuda HTML de Gestor Ayuda de Gestor Estado de Servidor

Ruta	Versión	Nombre a Mostrar	Ejecutándose	Sesiones	Comandos
/	Ninguno especificado	Welcome to Tomcat	true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar ≥ 30 minutos
/docs	Ninguno especificado	Tomcat Documentation	true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar ≥ 30 minutos
/examples	Ninguno especificado	Servlet and JSP Examples	true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar ≥ 30 minutos
/host-manager	Ninguno especificado	Tomcat Host Manager Application	true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar ≥ 30 minutos
/manager	Ninguno especificado	Tomcat Manager Application	true	1	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar ≥ 30 minutos

Desplegar

Desplegar directorio o archivo WAR localizado en servidor

Trayectoria de Contexto (opcional):

Version (for parallel deployment):

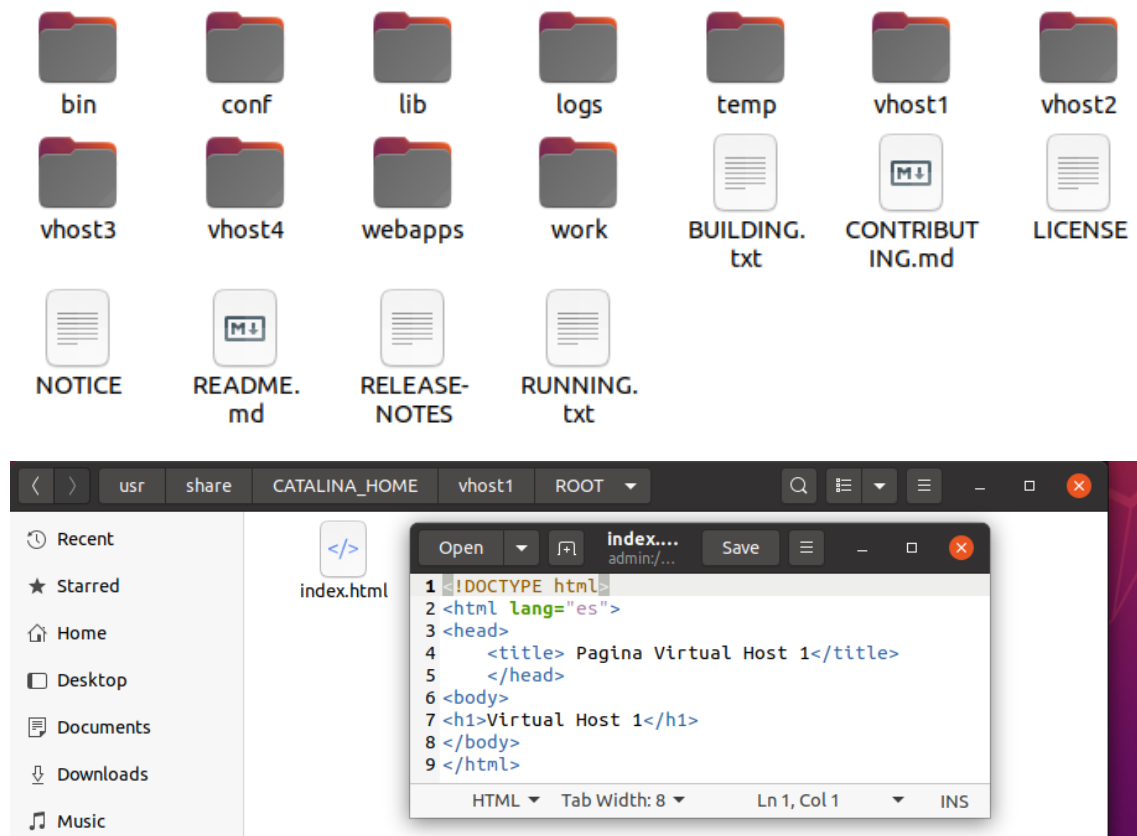
URL de archivo de Configuración XML:

4. Host Virtuales

Creamos dos nuevos servicios dentro del archivo server.xml, con los que crearemos 4 host virtuales, dos de ellos alojados en el puerto 81, ya que el puerto 80 estaba siendo usado por Apache y otro en el puerto 8085.

```
172 <Service name="Servicio1">
173   <Connector port="81" protocol="HTTP/1.1"/>
174   <Engine name="Servicio2" defaultHost="Localhost">
175     <Host name="localhost" appBase="vhost1"></Host>
176     <Host name="127.0.0.1" appBase="vhost2"></Host>
177   </Engine>
178 </Service>
179
180 <Service name="Servicio2">
181   <Connector port="8085" protocol="HTTP/1.1"/>
182   <Engine name="Servicio1" defaultHost="Localhost">
183     <Host name="localhost" appBase="vhost3"></Host>
184     <Host name="127.0.0.1" appBase="vhost4"></Host>
185   </Engine>
186 </Service>
```

Una vez realizado esto se crean cuatro carpetas dentro del directorio CATALINA_HOME/, sus nombres son vhost1, vhost2, vhost3, vhost4 y dentro de esas carpetas creamos la carpeta ROOT en donde creamos un nuevo archivo index.html, que tendrá el código html de la página del virtual host.



Finalmente comprobamos que todo funciona correctamente

