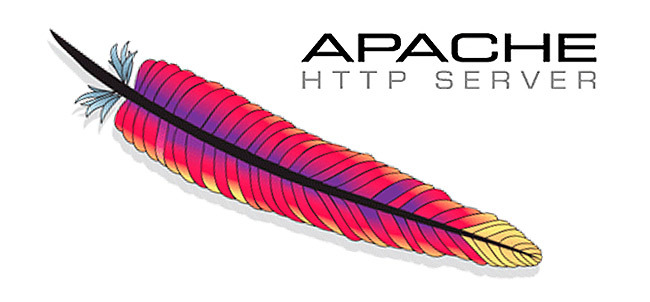
# **Tema 1: Apache**



Felipe Cendrero Diaz

Tabla de contenido

[**Tema 1: Apache** 1](#_Toc41290929)

[1.1 Instilación y configuración Apache 3](#_Toc41290930)

[1.2Virtual hosts apache2 5](#_Toc41290931)

[1.2.1 VirutalHosts basados en nombres 5](#_Toc41290932)

[1.2.2 VirtualHosts basados IP´s y puertos 8](#_Toc41290933)

[1.4 Módulo ssl para Apache 12](#_Toc41290934)

[1.5 Autenticación y control de acceso 17](#_Toc41290935)

[1.6 Logs 21](#_Toc41290936)

# 1.1 Instalación de Apache

Vamos a proceder a instalar apache2, que será el servidor web que montaremos.

Para ello, debemos introducir, una vez iniciado debian, el siguiente comando:

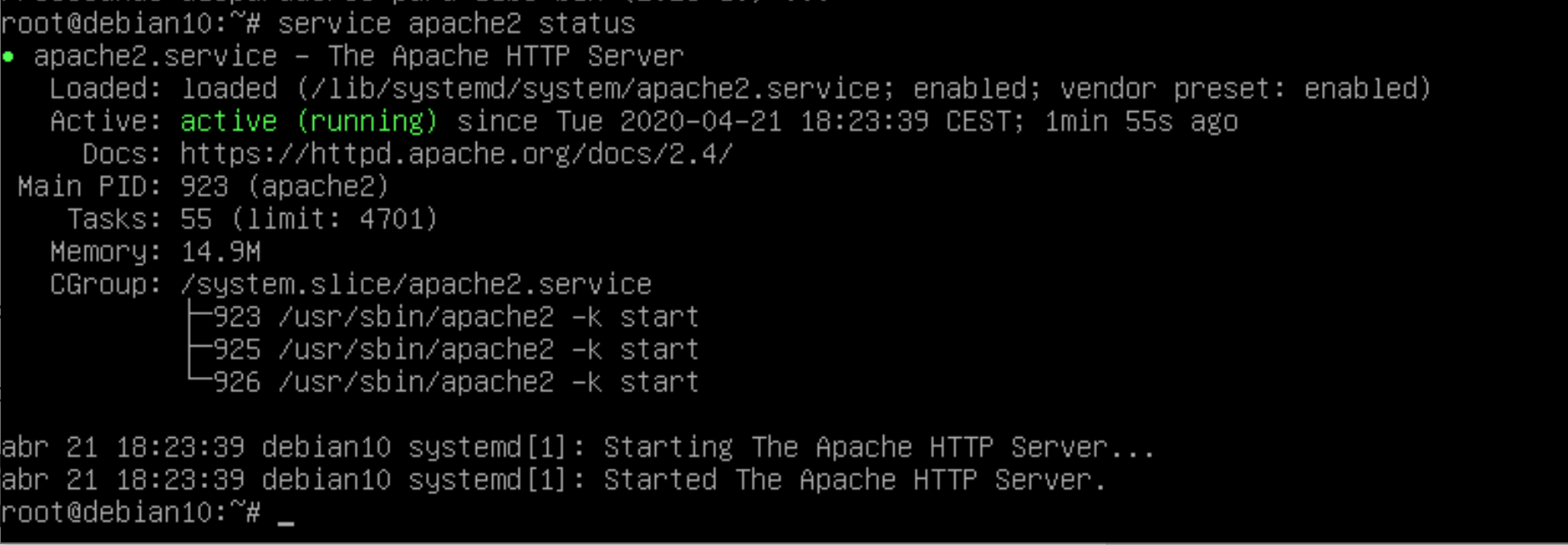
apt-get install apache2



Pulsamos “ S “ cuando nos pregunte si queremos instalarlo, y con esto, ya tendremos instalado apache2.

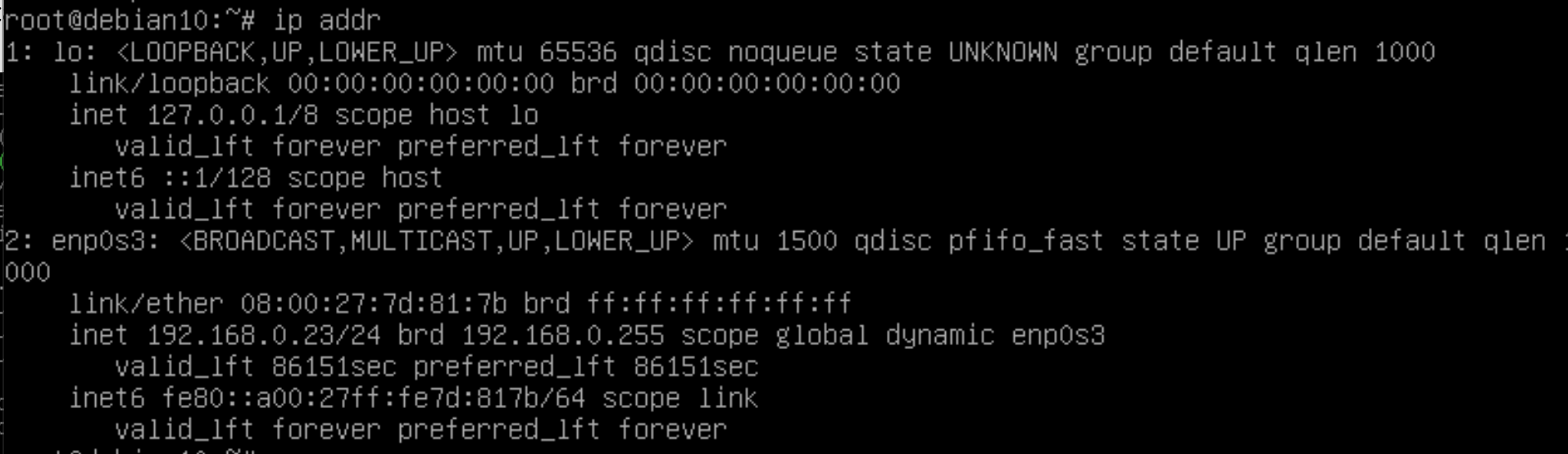
Para comprobar si se ha instalado sin problemas vamos a utilizar el siguiente comando:

service apache2 status

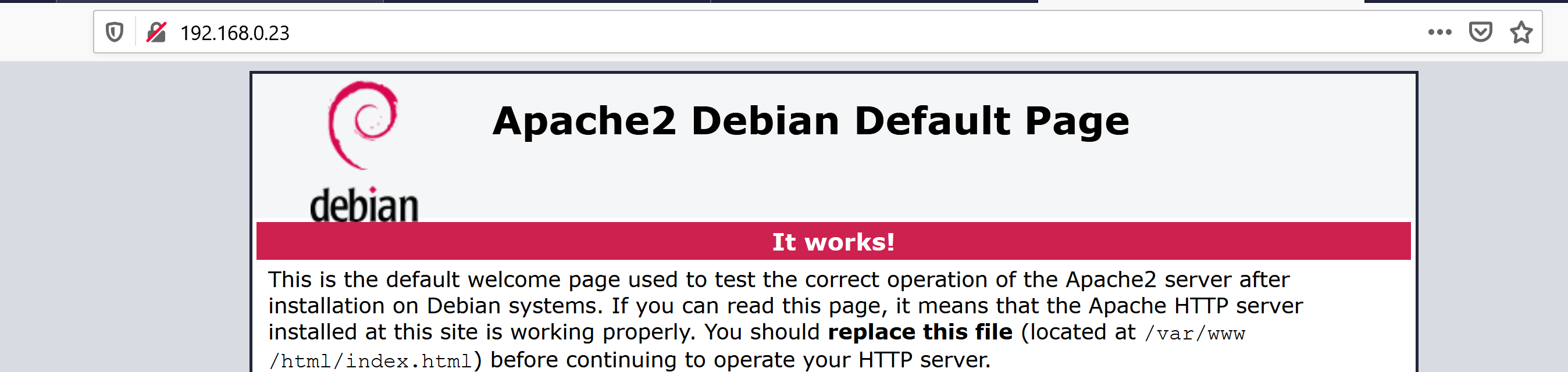


Otro modo de comprobarlo es poner el comando:

Ip addr



Y nos da la dirección IP del servidor. Una vez la tengamos, vamos al navegador y ponemos dicha dirección ip de la máquina virtual en el navegador web.



# 

# 1.2 Virtual hosts apache2

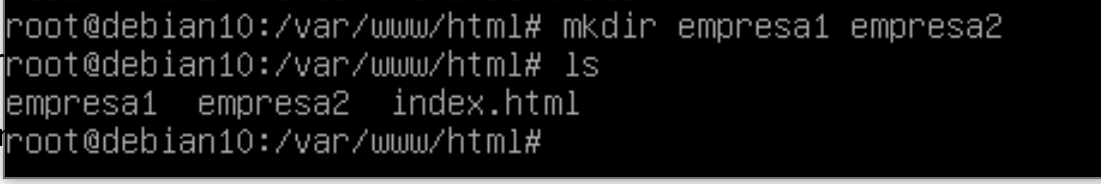
## 1.2.1 VirutalHosts basados en nombres

Ahora vamos a configurar nuestro apache2 para que pueda servir varias páginas web en el mismo servidor.

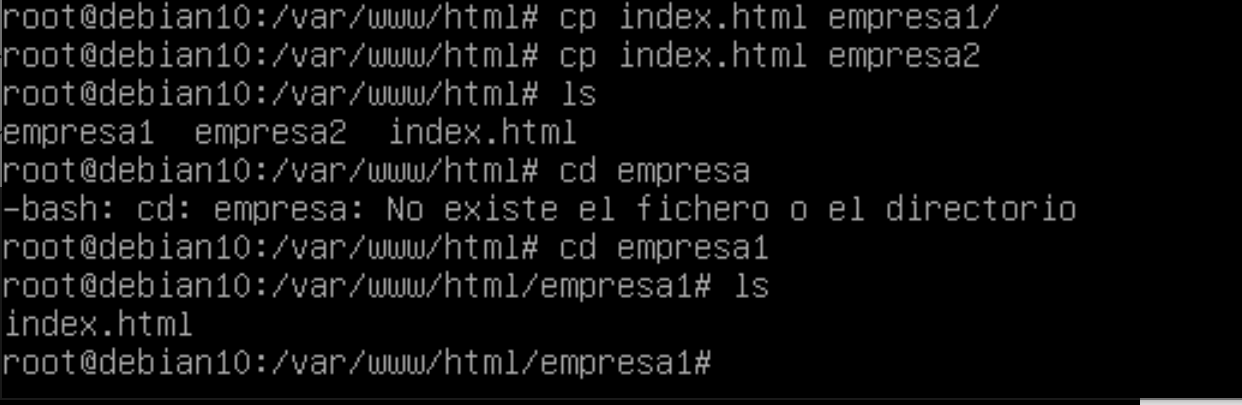
Lo primero que deberemos hacer es ir a la ruta.

/var/www/html/

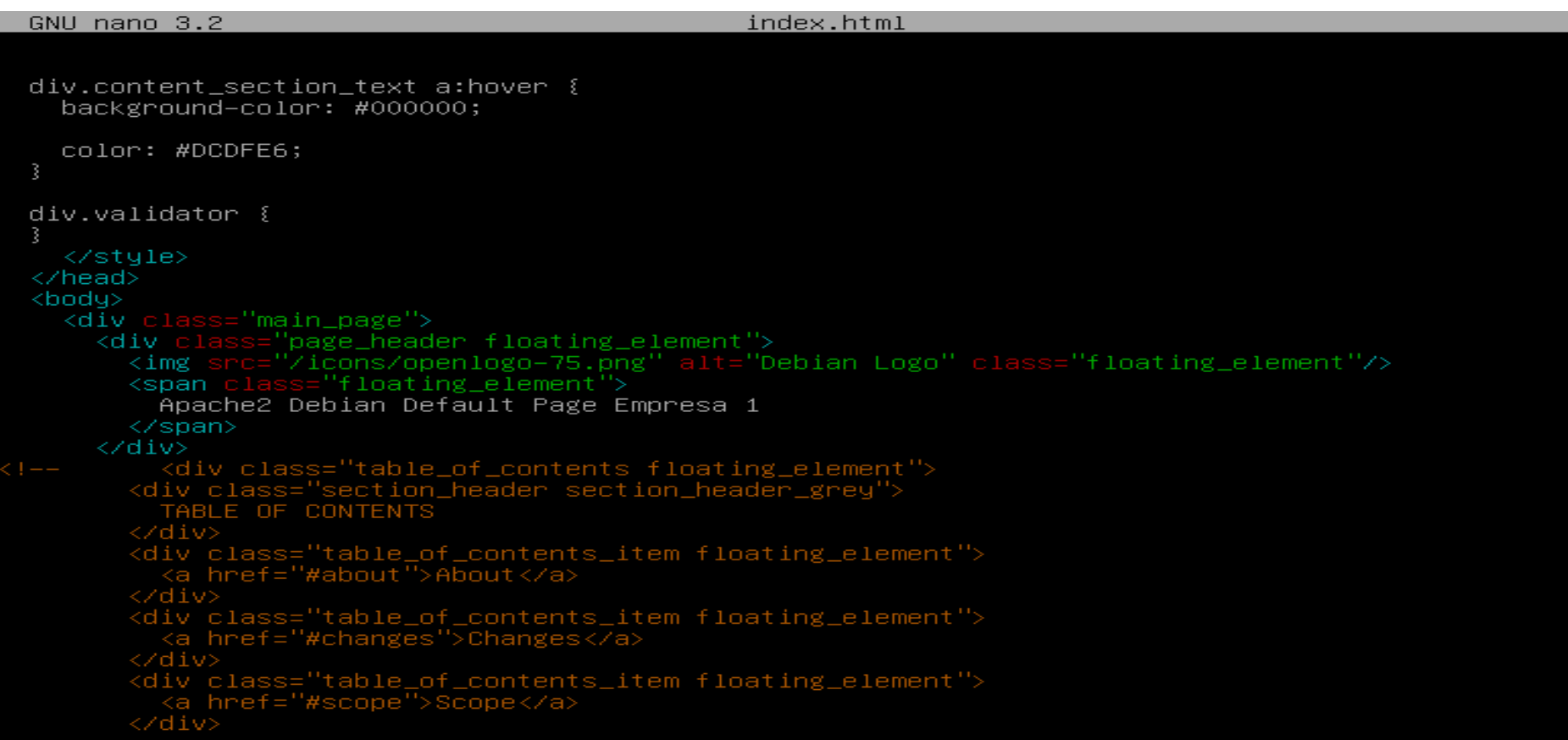
Una vez en el lugar, creamos dos carpetas: empresa1 y empresa2 con el comando:



Ahora, vamos a copiar el fichero index.html a las dos carpetas con el comando:



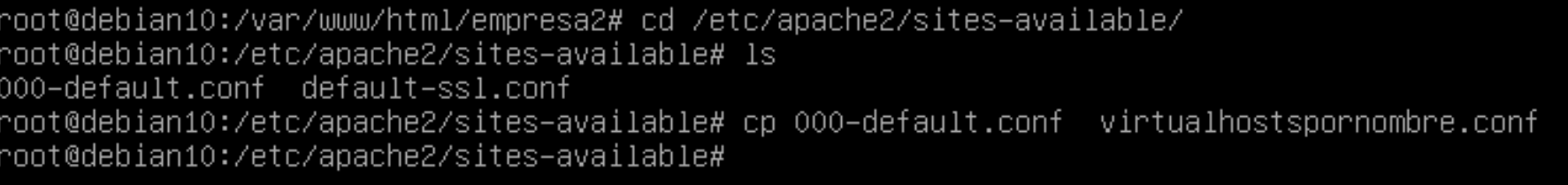
Una vez creadas las carpetas , vamos a cambiar los index.html y agregamos la frase Empresa 1 para indicar que son uno de empresa1 y otro de empresa2.





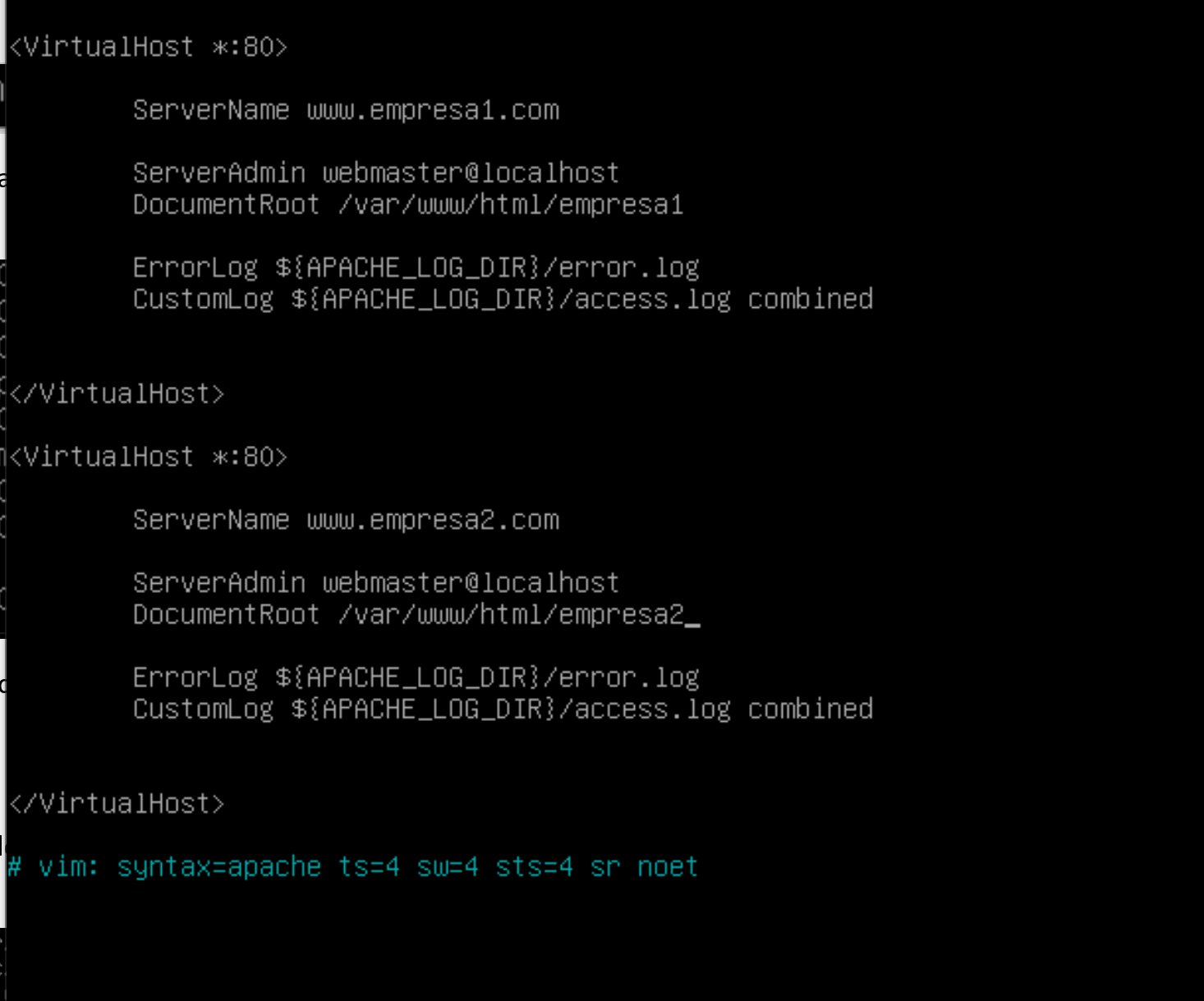
Modificados los ficheros y nos vamos a la siguiente ruta:

/etc/apache2/sites-available/



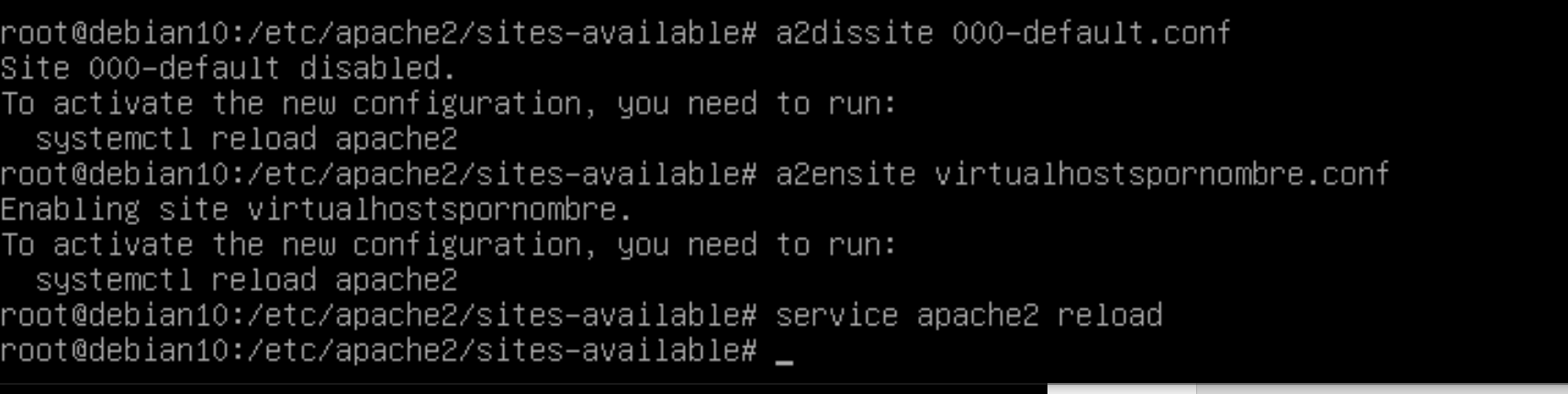
En esta ruta, copiamos el fichero por defecto y creamos un nuevo fichero, lo llamaremos virtualhostspornombre.conf

Deberemos dejar el fichero como éste, con el servername y la ruta de cada empresa.



Una vez tengamos el fichero guardado, vamos a desactivar el sitio web por defecto con el comando:

a2dissite 000-default.conf



Y activamos el que hemos creado:

a2ensite virtualhostspornombre.conf

Por último, ponemos el comando

service apache2 reload

que hará que el servidor vuelva a leer sus ficheros de configuración.

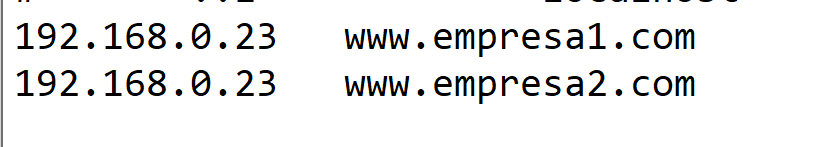
Ahora deberemos modificar el fichero hosts del SO donde estemos realizando las pruebas para indicarle que las urls son de nuestro servidor web.

En windows se encuentra en

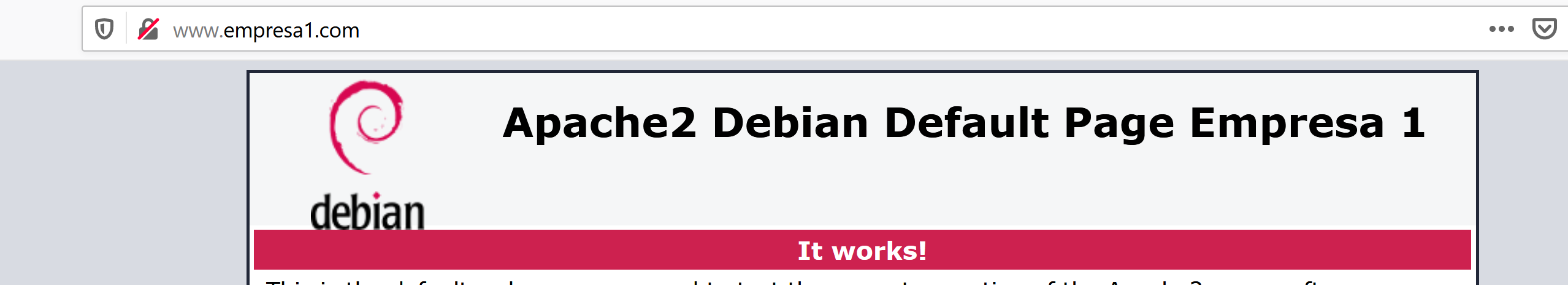
C:\Windows\System32\drivers\etc\hosts

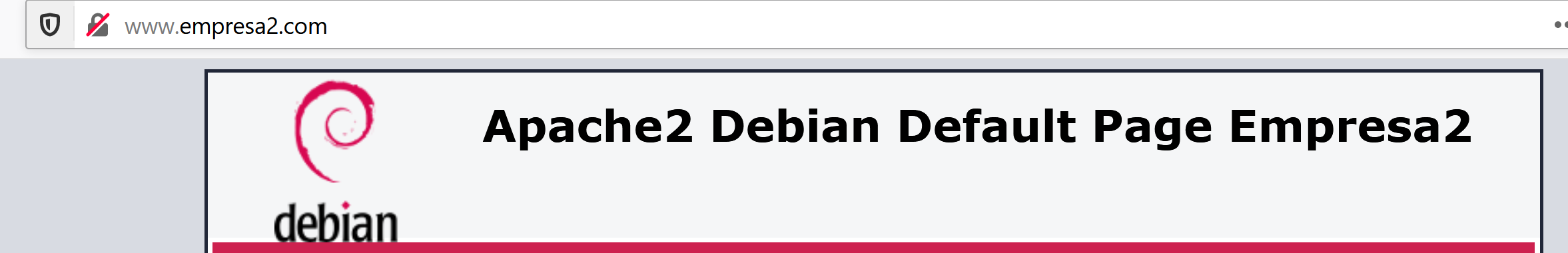
En debian

/etc/hosts



Ahora vamos a comprobarlo poniendo en el navegador web del equipo local las y comprobamos.





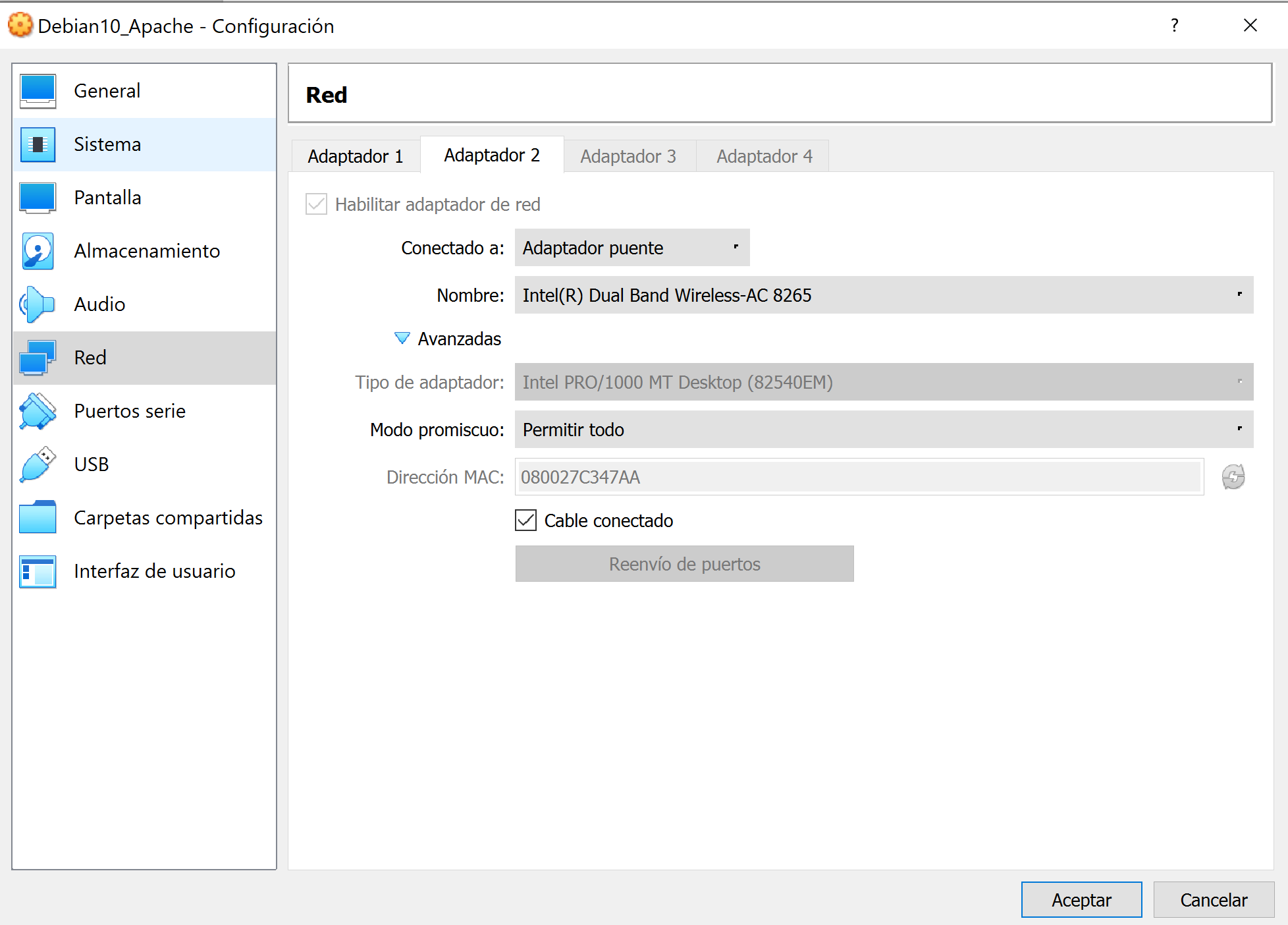
Como vemos, el servidor está funcionando correctamente.

## 1.2.2 VirtualHosts basados Ip´s y puertos

Apagamos la máquina virtual y vamos a configuración, al apartado "Red".

En el apartado "Conectado a" seleccionamos "Adaptador puente" y en "Nombre" quedamos la opción que aparece, que debe ser el adaptador de red (tarjeta de red de nuestro ordenador).

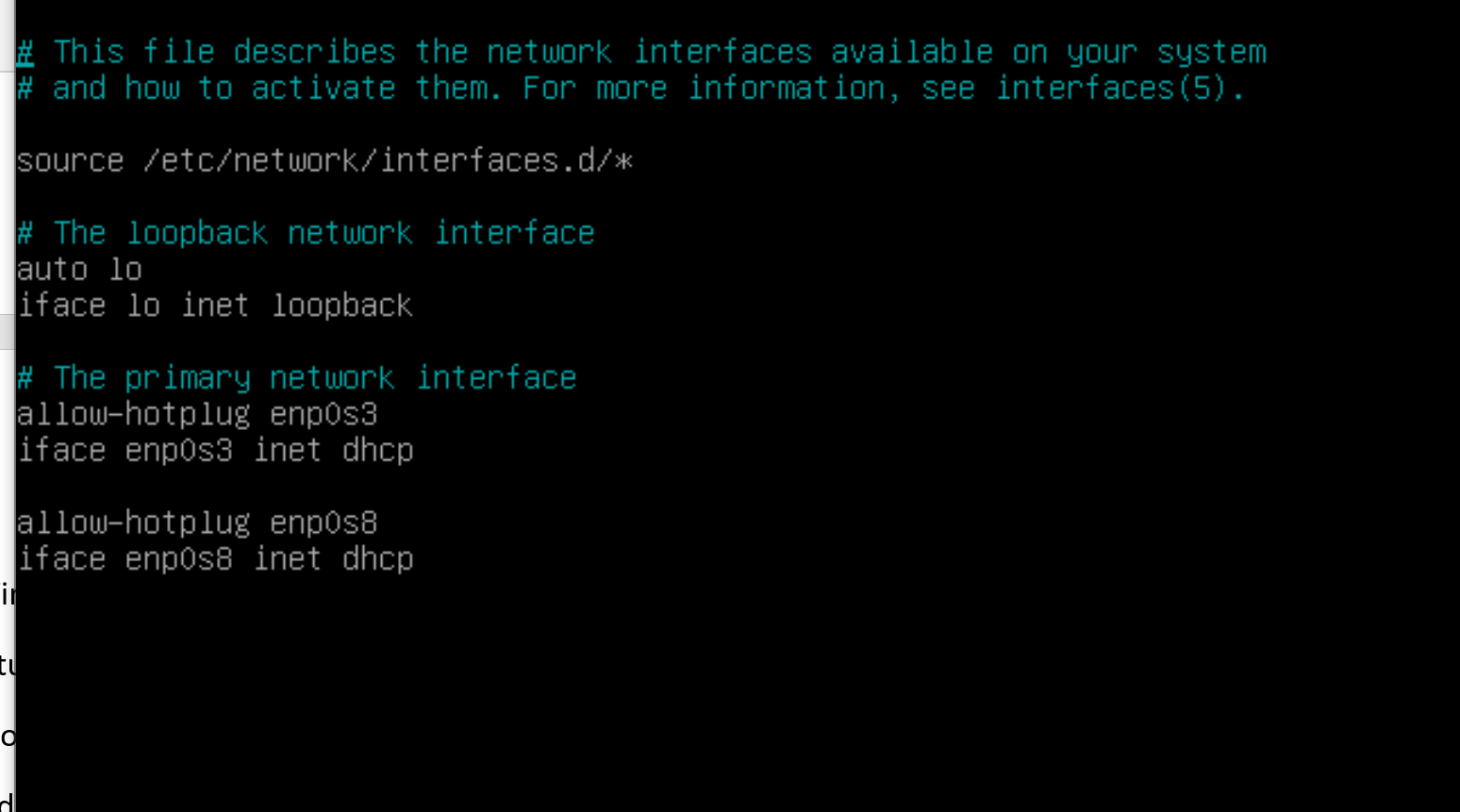
En nuestro caso necesitamos dos adaptadores, vamos al segundo (solapa Adaptador 2), lo habilitamos y volvemos a hacer los mismos pasos que antes.



Arrancamos la máquina virtual. Una vez dentro de ella vamos a establecer las direcciones IP's para cada uno de los adaptadores.

En el fichero /etc/network/interfaces

Copiamos en el fichero la parte de enp0s3 por y cambiamos en enp0s8.

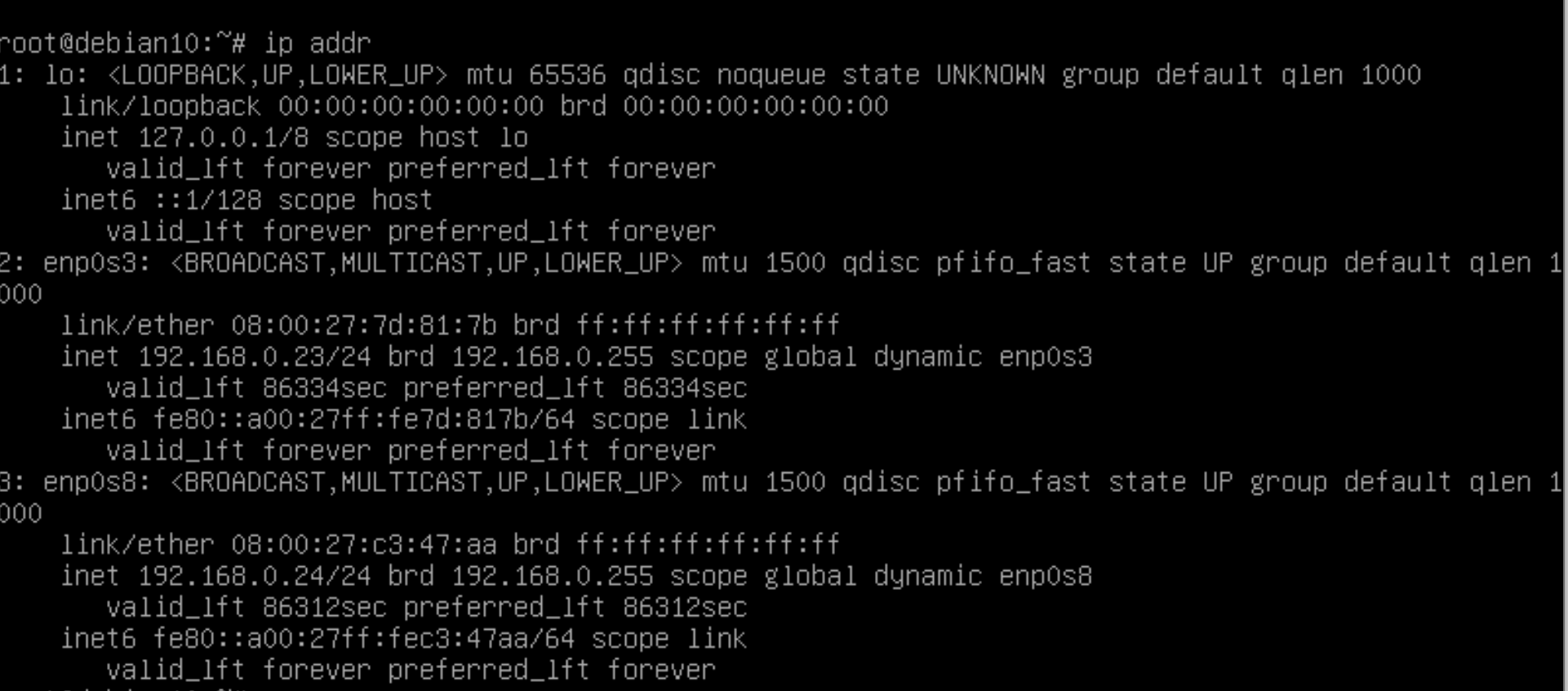


Luego ponemos los comandos

ifup enp0s8

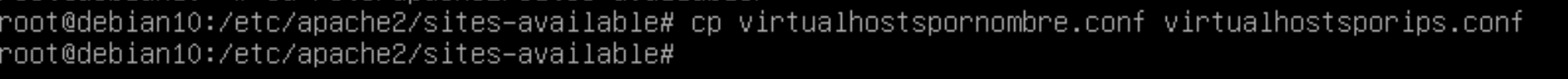
ifup enp0s3

Y comprobamos las direcciones con el comando ip addr

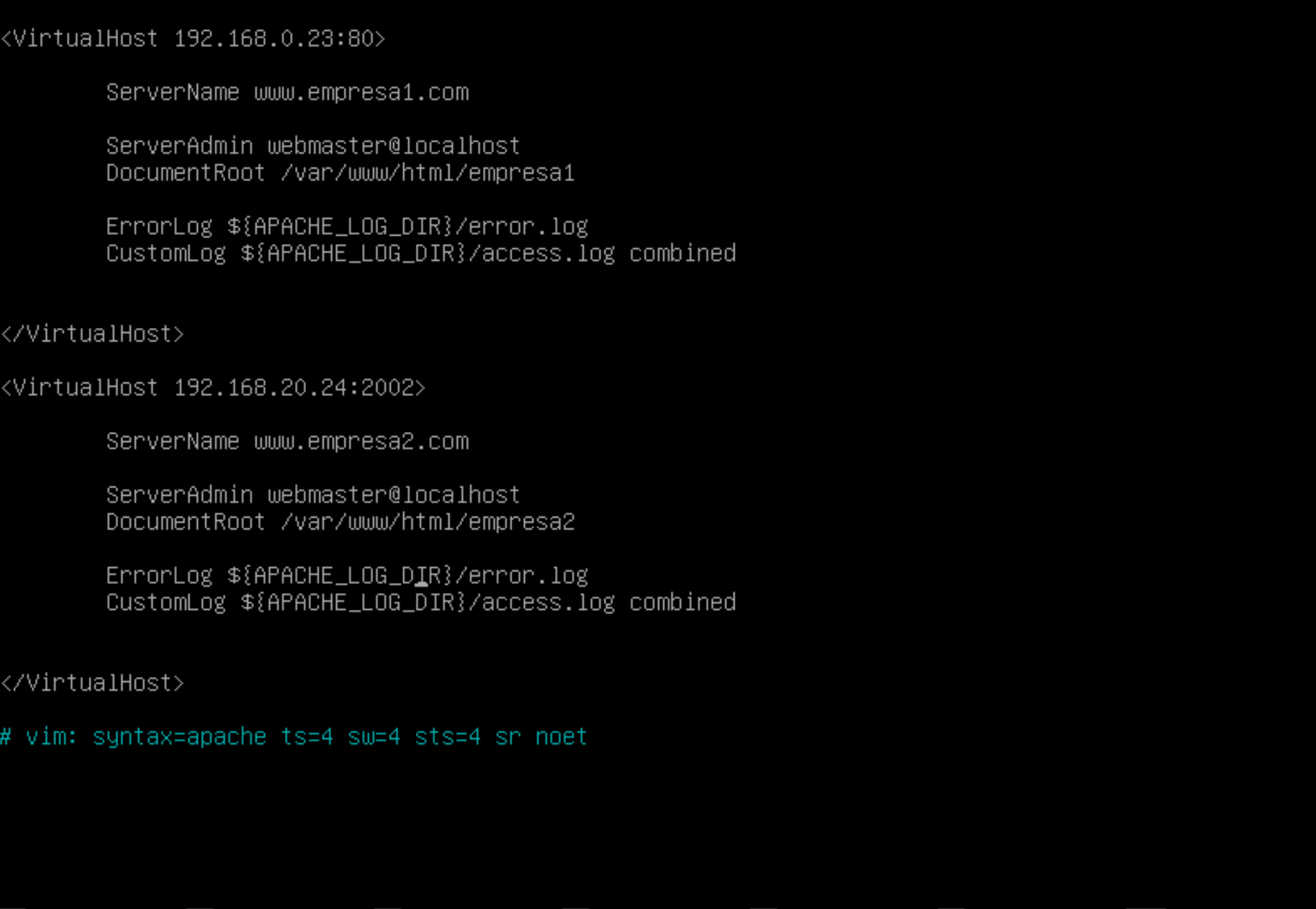


Una vez que tengamos las dos interfaces funcionando nos pondremos a configurar el servidor. Para ello, deberemos realizar las siguientes configuraciones:

En primer lugar, volvemos a ir a la ruta /etc/apache2/sites-available y creamos un nuevo fichero como hicimos anteriormente.



A continuación, copiamos el fichero y creamos uno nuevo para cambiar lo siguiente:



Como vemos hemos puesto la dirección IP y hemos cambiado el puerto de escucha de empresa2.

Ahora desactivamos el sitio que teníamos activo y activamos el que acabamos de configurar.

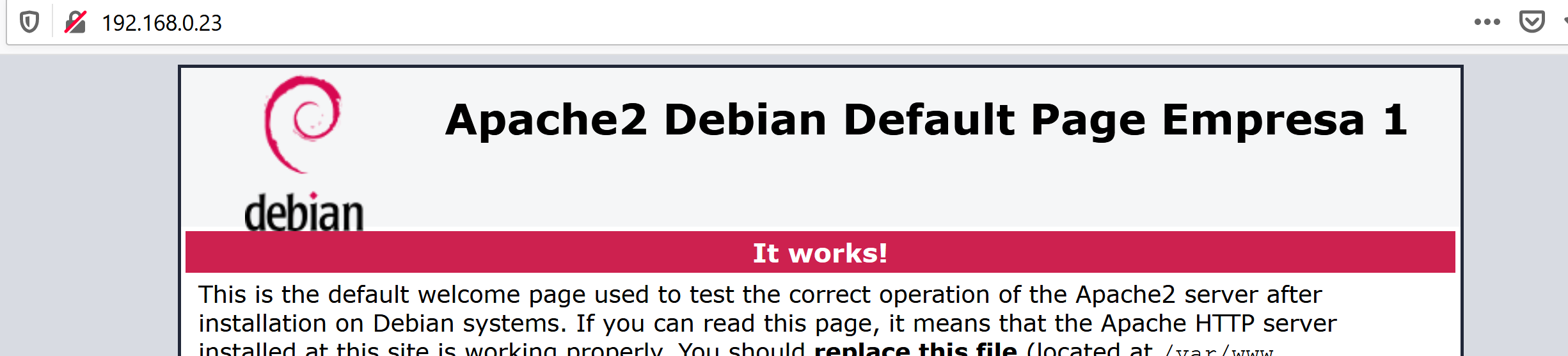
a2dissite virtualhostspornombre.conf

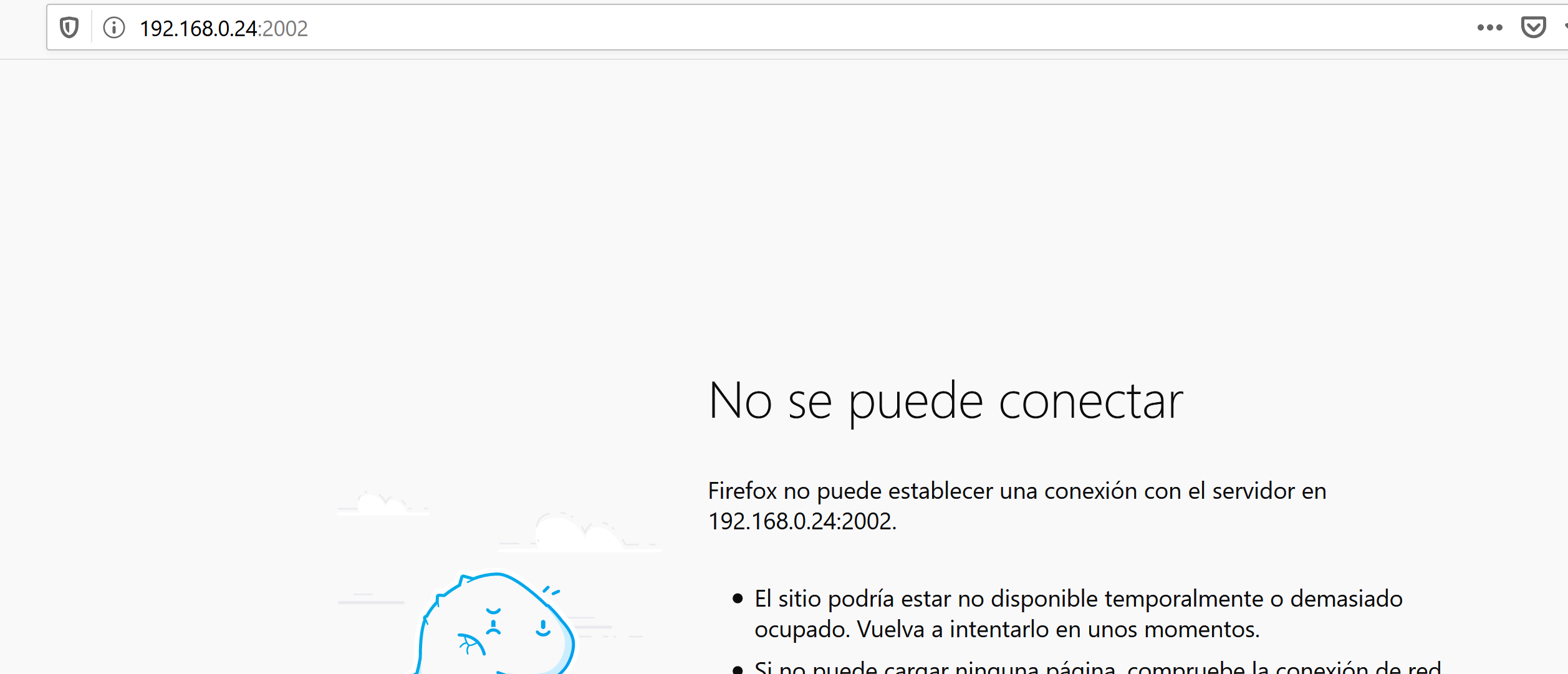
a2ensite virtualhostsporips.conf

Siempre debemos recargar el servidor con el comando

service apache2 reload

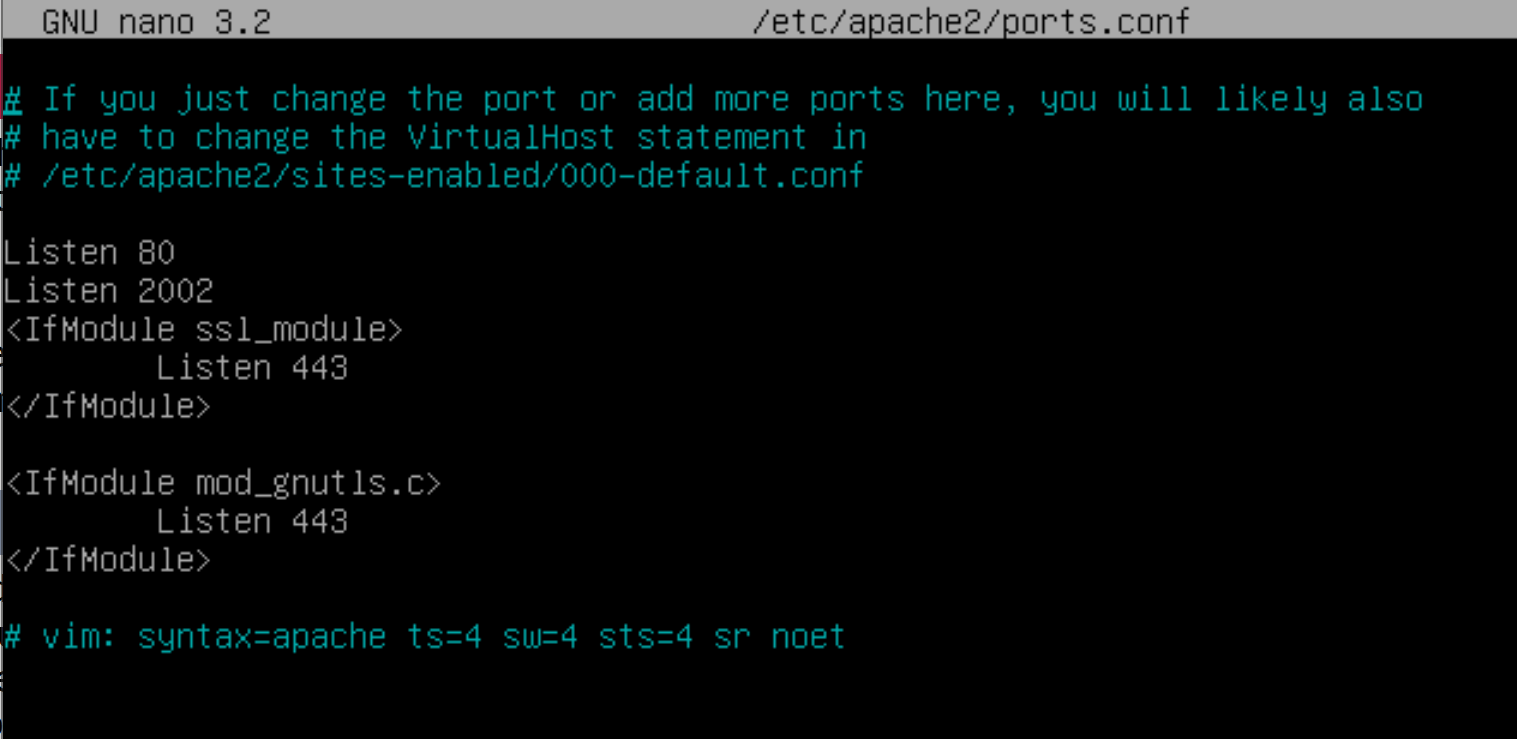
Y comprobamos:



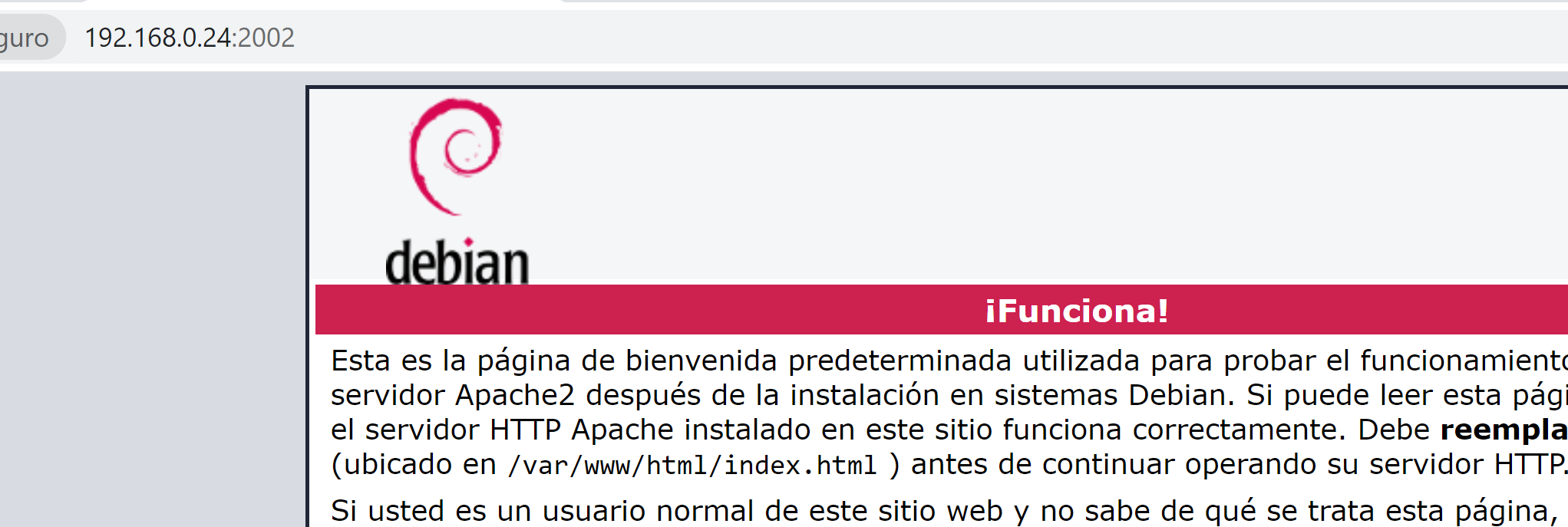


Al intentar acceder a apache por el puerto que no es el 80(puerto por defecto de los servicios web) nos da error. Esto es debido a que no hemos configurado correctamente el servicio, nos falta una pequeña configuración que deberemos hacer.

Para ello debemos modificar el siguiente fichero:



Al incluir en el fichero /etc/apache2/ports.conf la línea Listen 2002 debería funcionar. Recargamos el servicio apache2:



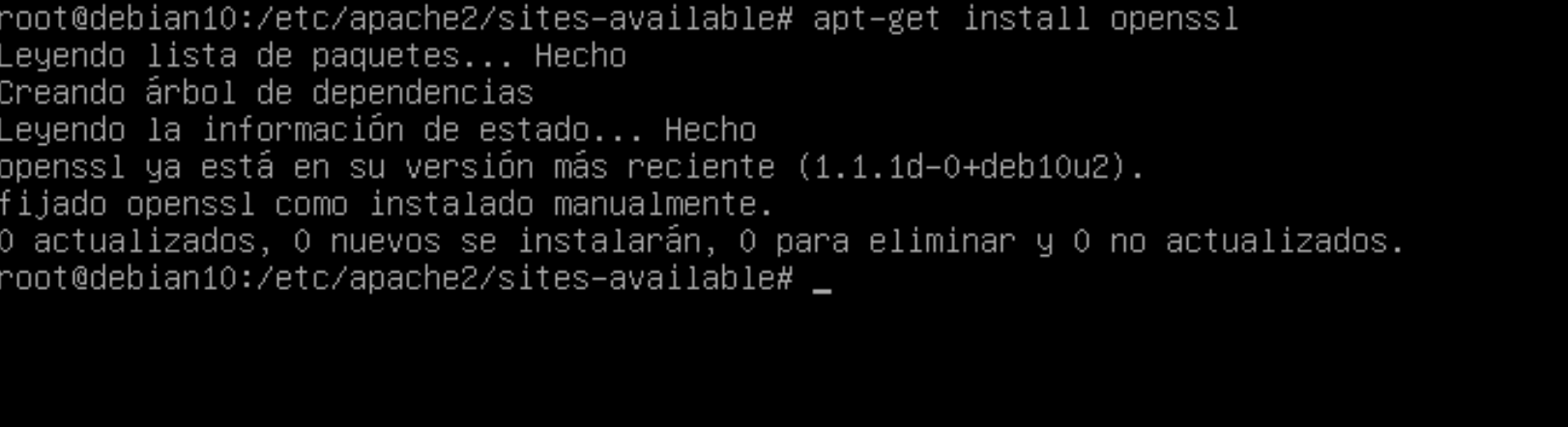
Como podemos comprobar, ya funciona tanto el atender por dos IP´s como el atender por distintos puertos.

# 1.4 Módulo ssl para Apache

Vamos a realizar ahora la configuración del servicio ssl para Apache.

Esto nos establece una conexión segura entre el servidor y el cliente, ya que la información viaja encriptada. Para ello, lo primero que deberemos hacer es crear una certificación, la cual conseguiremos instalando lo siguiente:

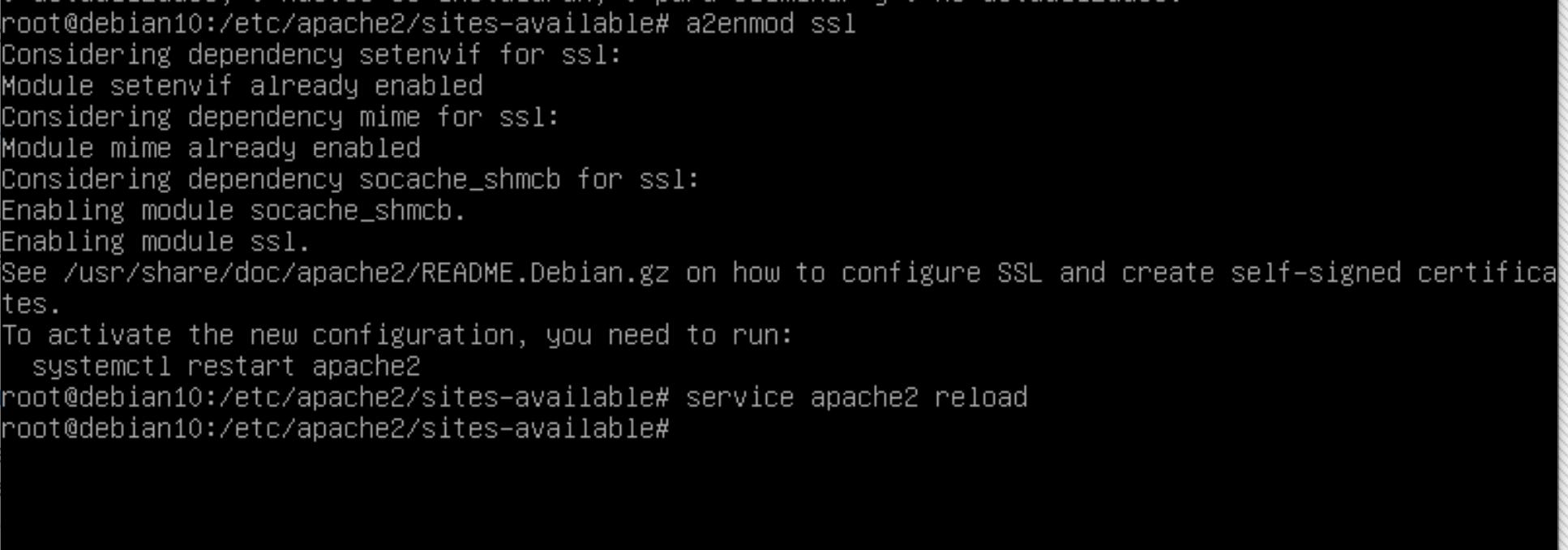
apt-get install openssl



Nosotros ya lo tenemos instalado, por lo tanto, vamos a pasar el siguiente paso.

Este módulo viene instalado por defecto en apache2, pero esta desactivado, así tendremos que activarlo. Utilizaremos el siguiente comando:

A2enmod ssl



Habilitado el servicio ssl, deberemos crear el certificado digital, ya que lo que vamos a hacer es autocertificarnos.

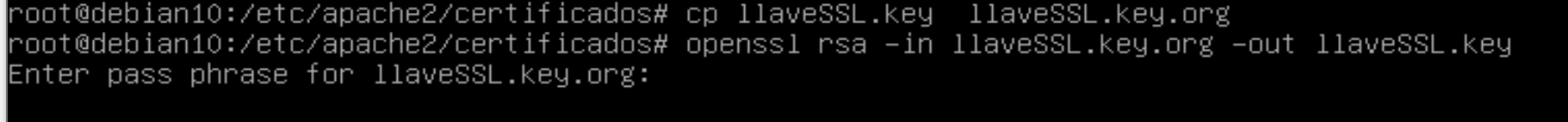
Lo primero que haremos será crear una carpeta donde guardar nuestro certificado:



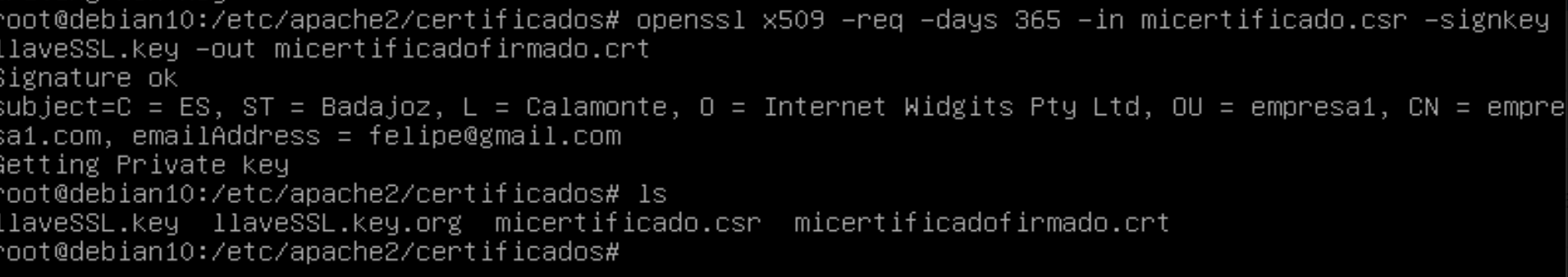
Luego crearemos una clave que nos pedirá dos veces una contraseña:



Rellenos todos los datos, ya tendremos nuestros certificados.

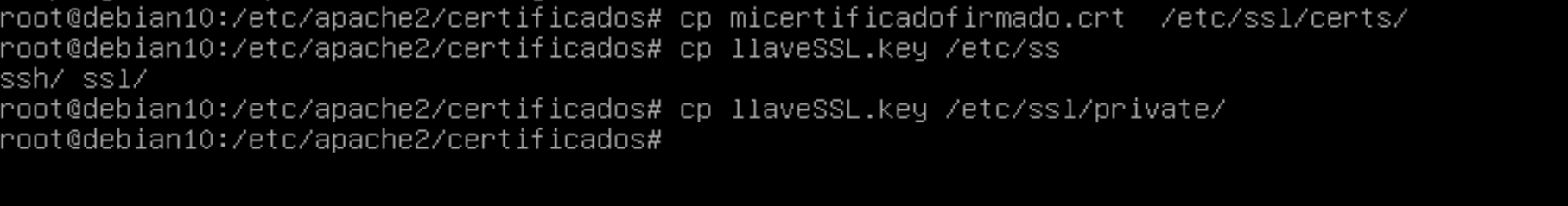


Ahora deberemos ocultar la llave, para que también se cifre la contraseña, ya que, si no, sería una vulnerabilidad de nuestro sistema.



Ahora deberemos autofirmar el certificado.

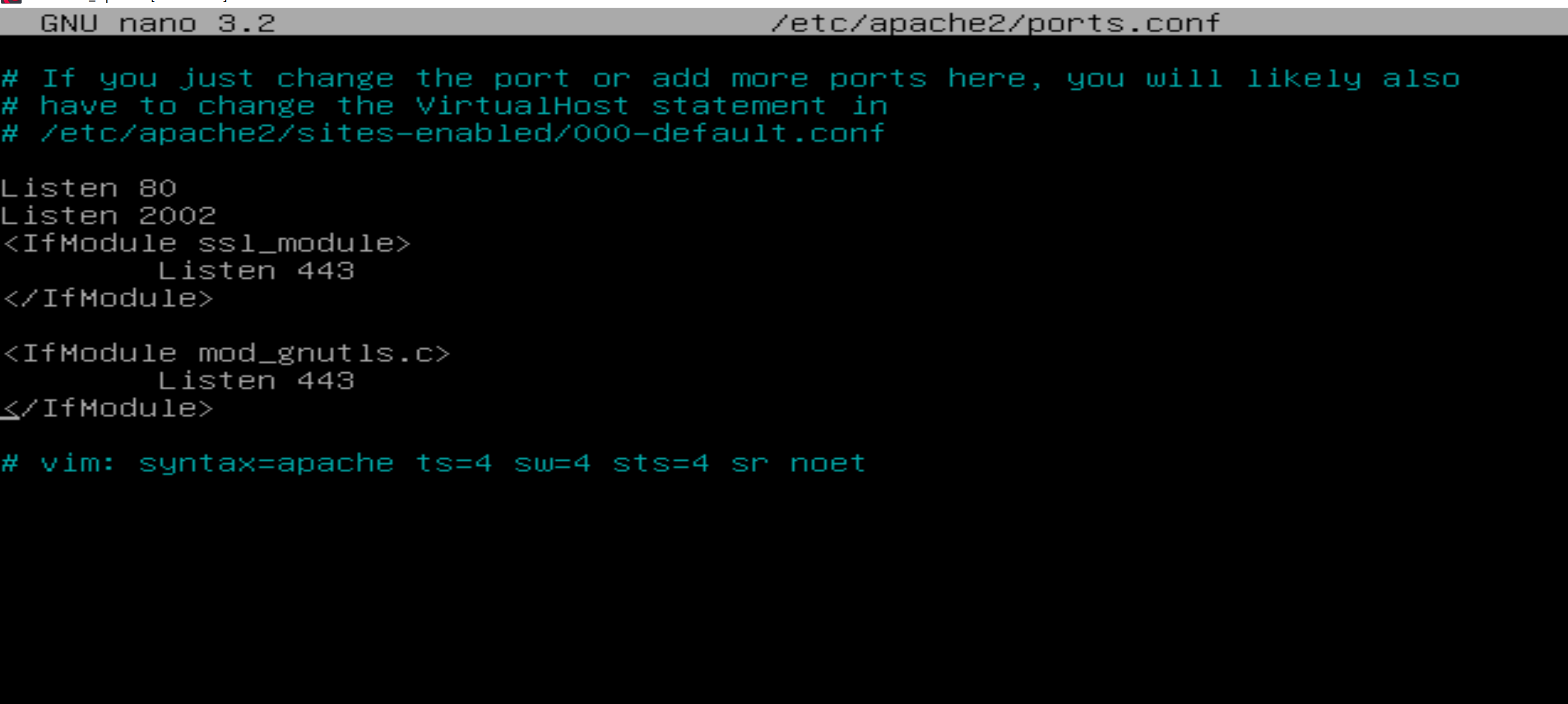
A continuación, vamos a copiar y habilitar nuestros certificados en apache. Lo haremos copiando los ficheros en los lugares donde apache suele buscarlos.



Una vez terminado, nos vamos a dirigir a la carpeta de empresa1 en la dirección: /var/www/html/empresa1

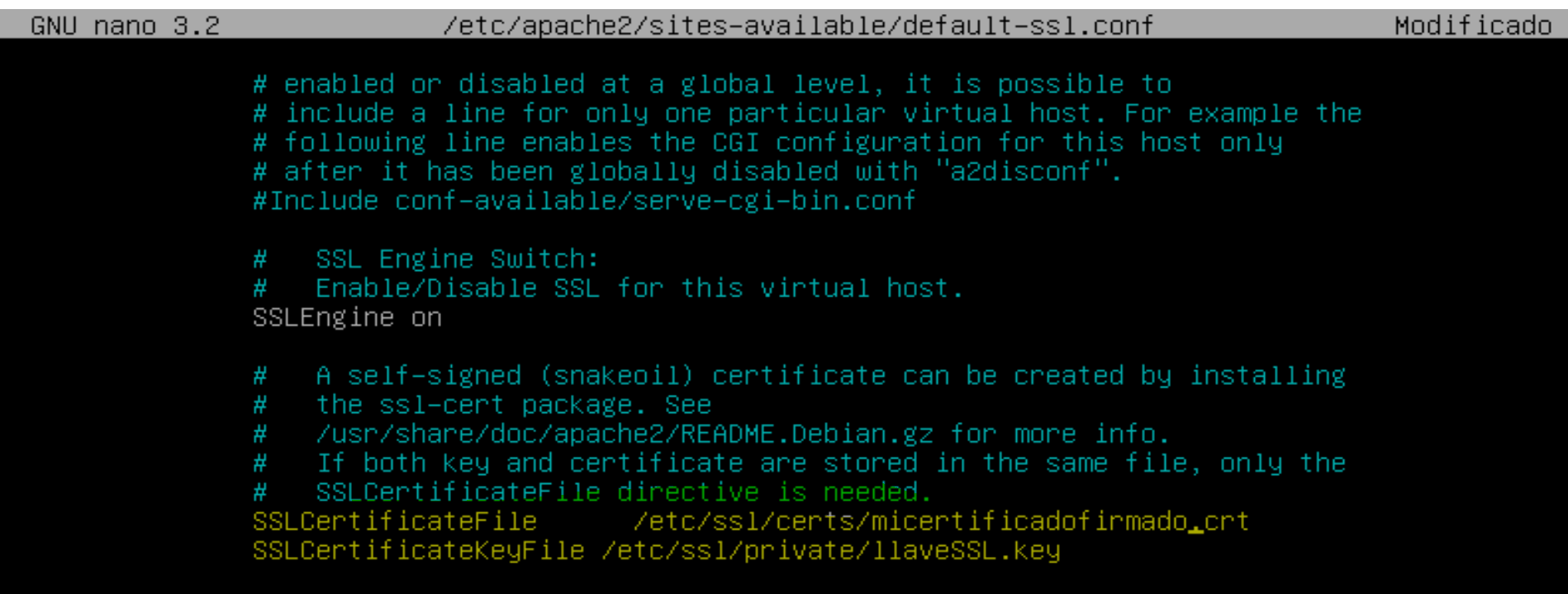
Vamos a crear una carpeta llamada segura. Copiamos el index.html de la carpeta 1 dentro de la carpeta creada y lo modificamos.

Ahora deberemos comprobar que el puerto 443(puerto que utiliza el protocolo https) esté habilitado en el ports.conf:



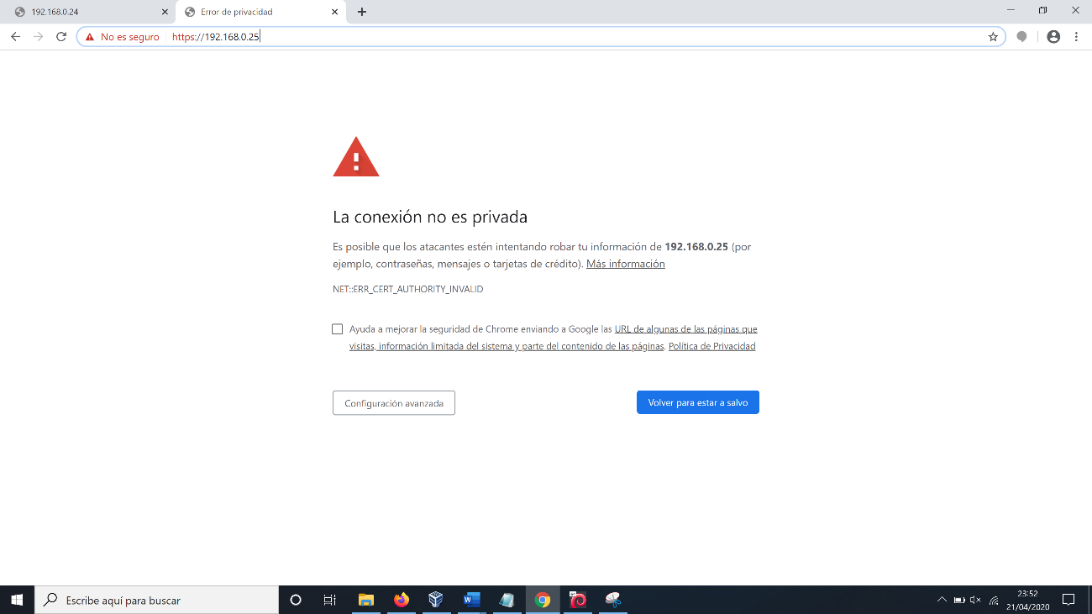
Como podemos comprobar, está habilitado. Por tanto, no tenemos que hacer nada.

El siguiente paso consiste en cambiar los certificados originales que trae apache por los nuestros.

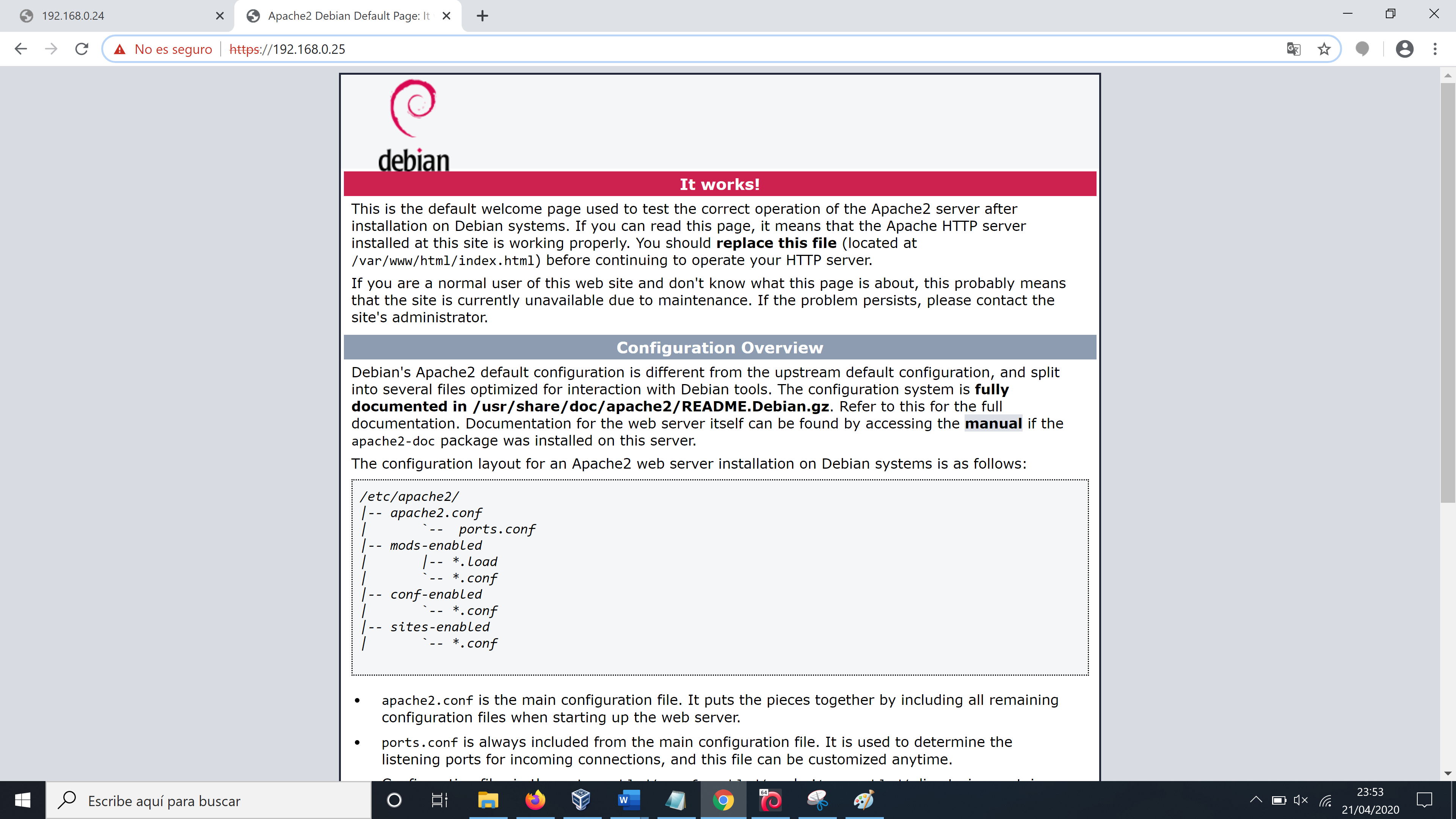


Deberemos modificar el fichero default-ssl.conf y cambiar los ficheros que vienen por los nuestros.

Ahora reiniciamos el servidor y comprobamos :



Vamos a intentar acceder :



Al acceder al sitio web nos indica que no es seguro, ya que el archivo está autofirmado y, por tanto, no está certificado por una autoridad. Pero como vemos, accede por protocolo https.

# 1.5 Autenticación y control de acceso

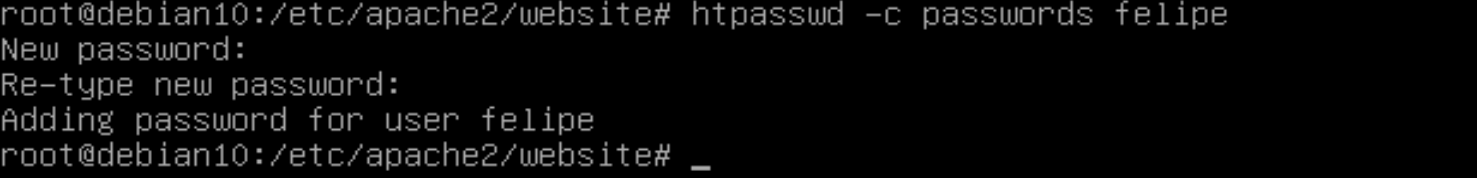
## 1.5.1 autenticación BASIC

Vamos a comprobar la autentificación. Para ello, vamos a utilizar los dos métodos que existen: basic y digest. En ambos, lo primero que deberemos hacer es crear tanto en empresa1, como en empresa 2, las carpetas privadas. Utilizaremos para tal fin el comando mkdir.

Ahora vamos a crear una carpeta en el fichero de /etc/apache2 que será donde guardaremos los usuarios que pueden acceder a las carpetas.

Una vez en website, ponemos el siguiente comando:

htpasswd –c passwords felipe

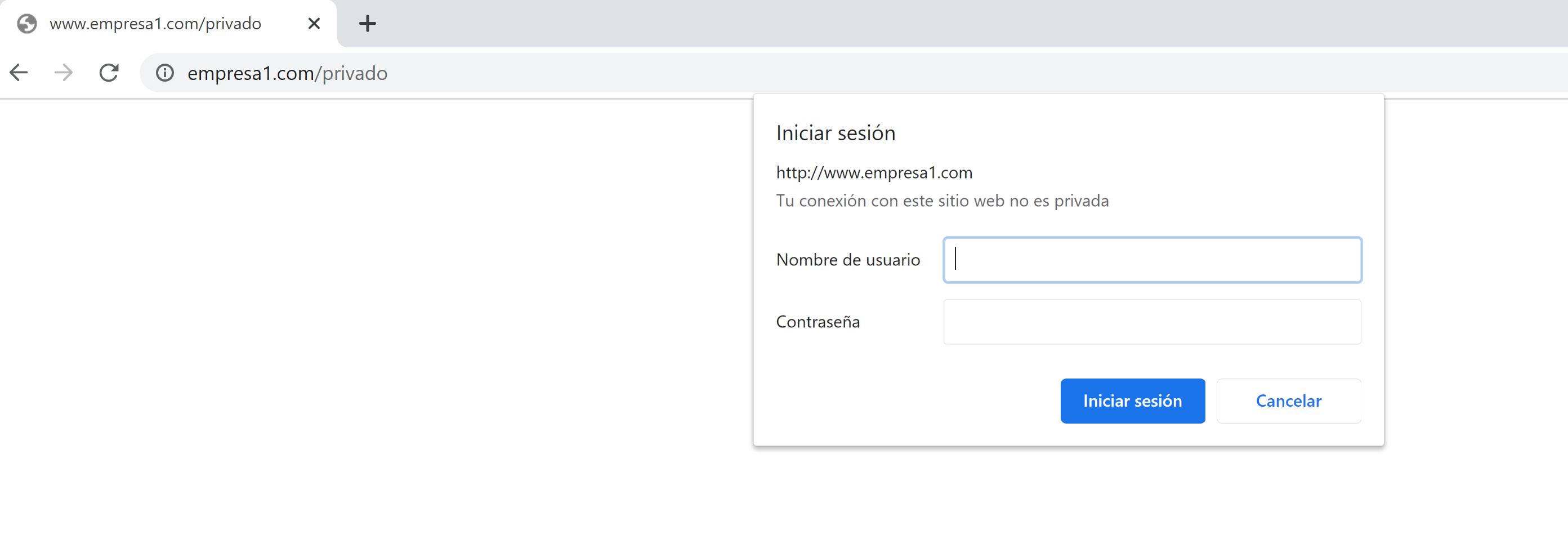


Creado el fichero en el que guardamos a los usuarios, vamos a crear un nuevo virtualhost para acceder a él.

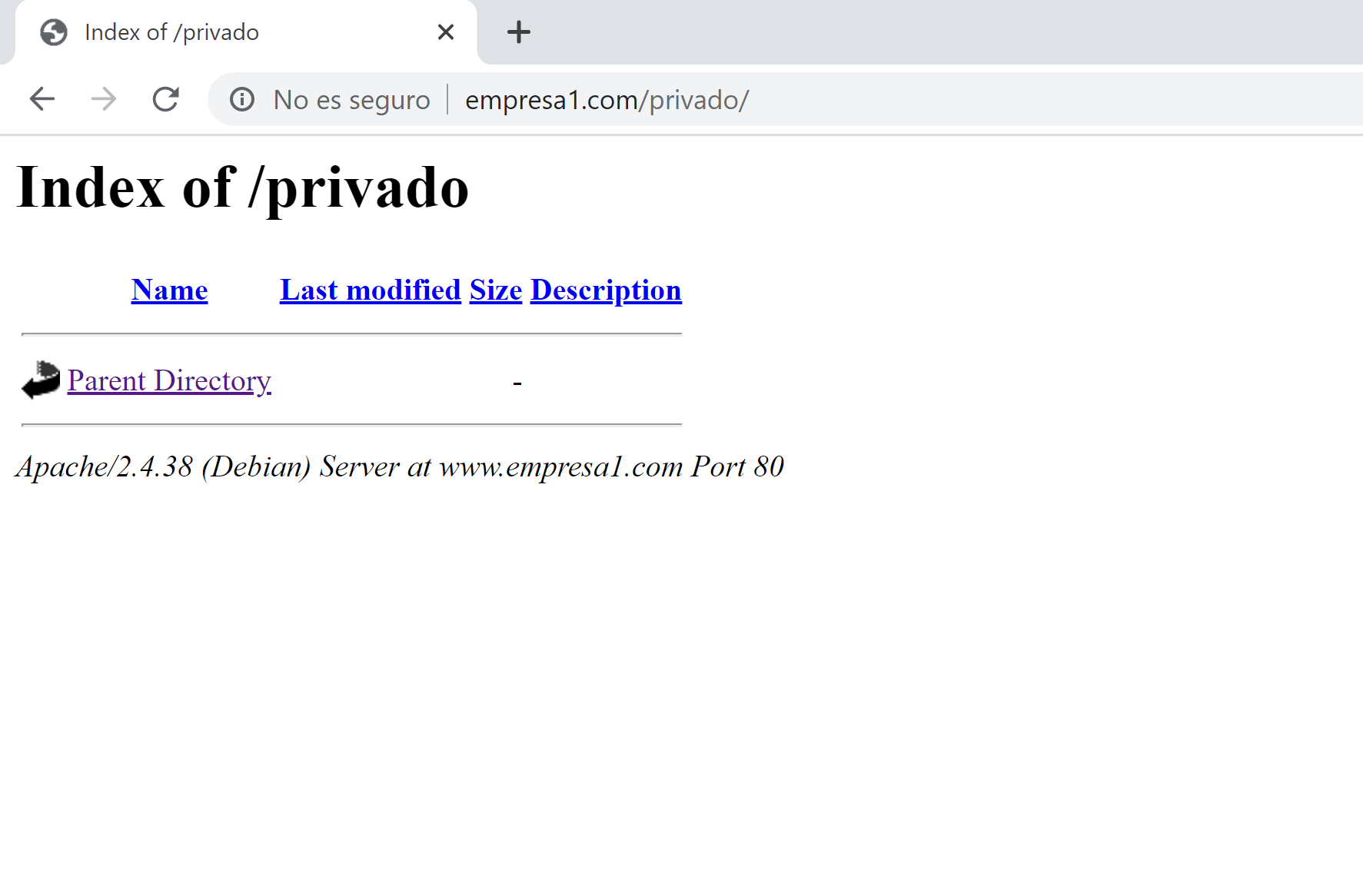


Incluimos estas líneas en el fichero virtualhostspornombre.conf y una vez finalizado, reiniciamos el servidor y actualizamos.

Por último, comprobamos si ha funcionado.



Como podemos ver, la autentificación ha funcionado. Vamos a comprobar si podemos acceder con el usuario felipe

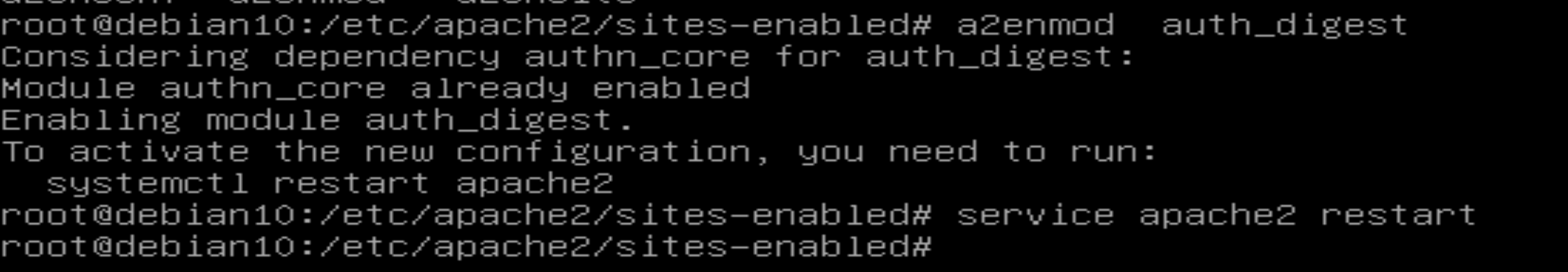


## 1.5.2 Autenticación DIGEST

Ahora vamos a utilizar el otro método de autenticar

Vamos a crear una carpeta en empresa2 para autenticar con digest.

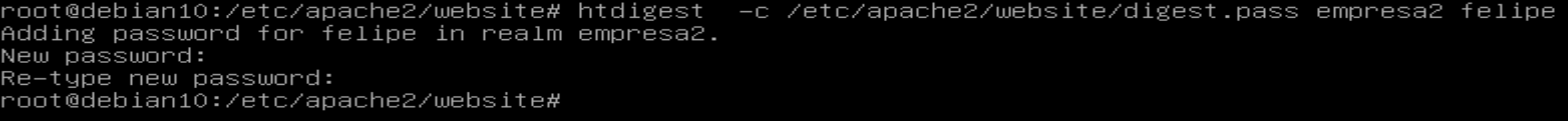
Lo primero que debemos hacer es habilitar el módulo de autenticación con digest .



Ahora deberemos de incluir los siguiente en el fichero de configuración del virtualhost:



Nos queda crear el fichero de autenticación:

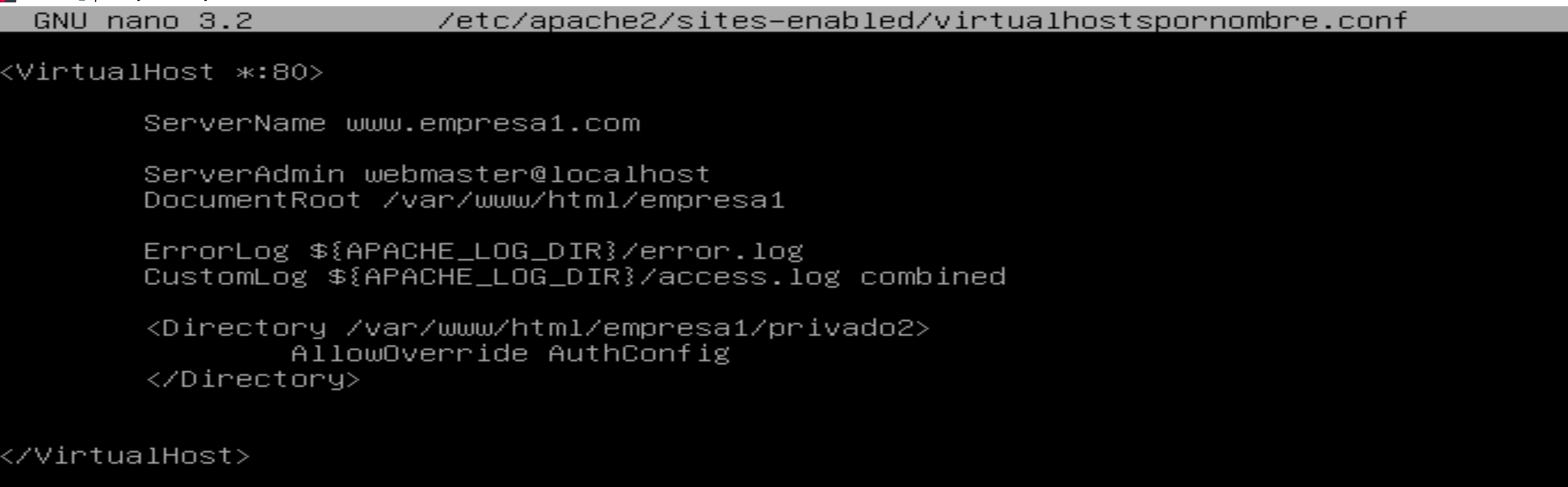


Por último, comprobamos si funciona:



Ahora vamos a configurar el fichero .htaccess. Éste es un fichero donde entraremos las directivas para la autenticación, en vez de hacerlo directamente en el fichero de hostsvirutales.

Lo primero que haremos será crear un nuevo directorio en /var/www/html/empresa1/privado2



A continuación, dentro del hostvirtual empresa1 ponemos el directory que es la directiva para configurar una carpeta y su configuración.

Ahora creamos un fichero .htaccess dentro de la carpeta privado2 y agregamos las siguientes directivas:



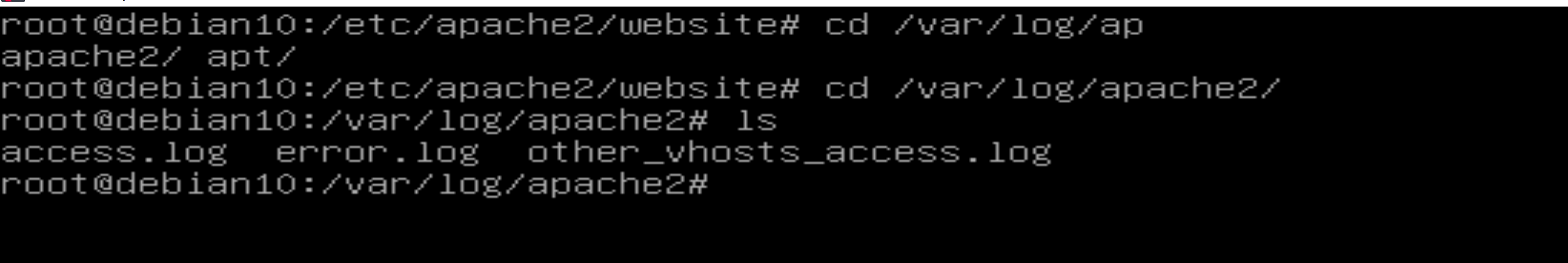
Reiniciamos el servidor y comprobamos:



# 1.6 Logs

Por último, vamos a trabajar un poco con los logs.

Vamos a crear nuestros propios logs y vamos a rotarlo. Para ello, nos dirigiremos /var/log/apache2



Ahora vamos a crear nuestros propios logs. Para conseguirlo, crearemos cuatro ficheros: dos ficheros para el acceso tanto de empresa 1 como empresa 2 que sean independientes, y otros dos ficheros con errores para las empresas

Los ficheros se crean automáticamente por el servidor.

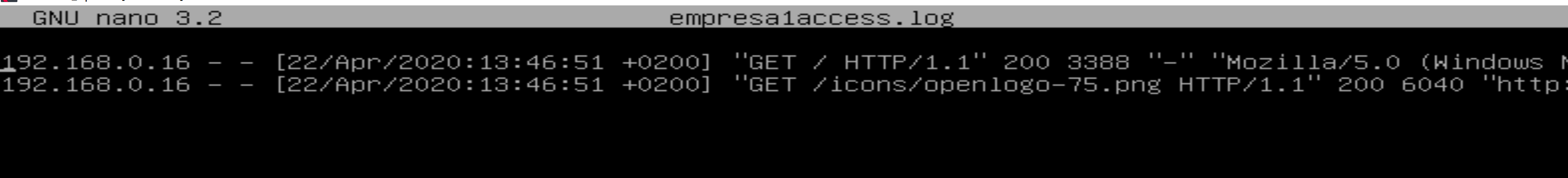


Una vez creados los ficheros nos vamos a nuestro fichero de hostvirtuales y modificamos lo siguiente:



De esta manera lo que hacemos al ponerle los nombre es poder identificar los errores y los accesos individualmente para cada host virtual.

Ahora sólo nos queda reiniciar el servidor y comprobar si se escribe bien los datos.



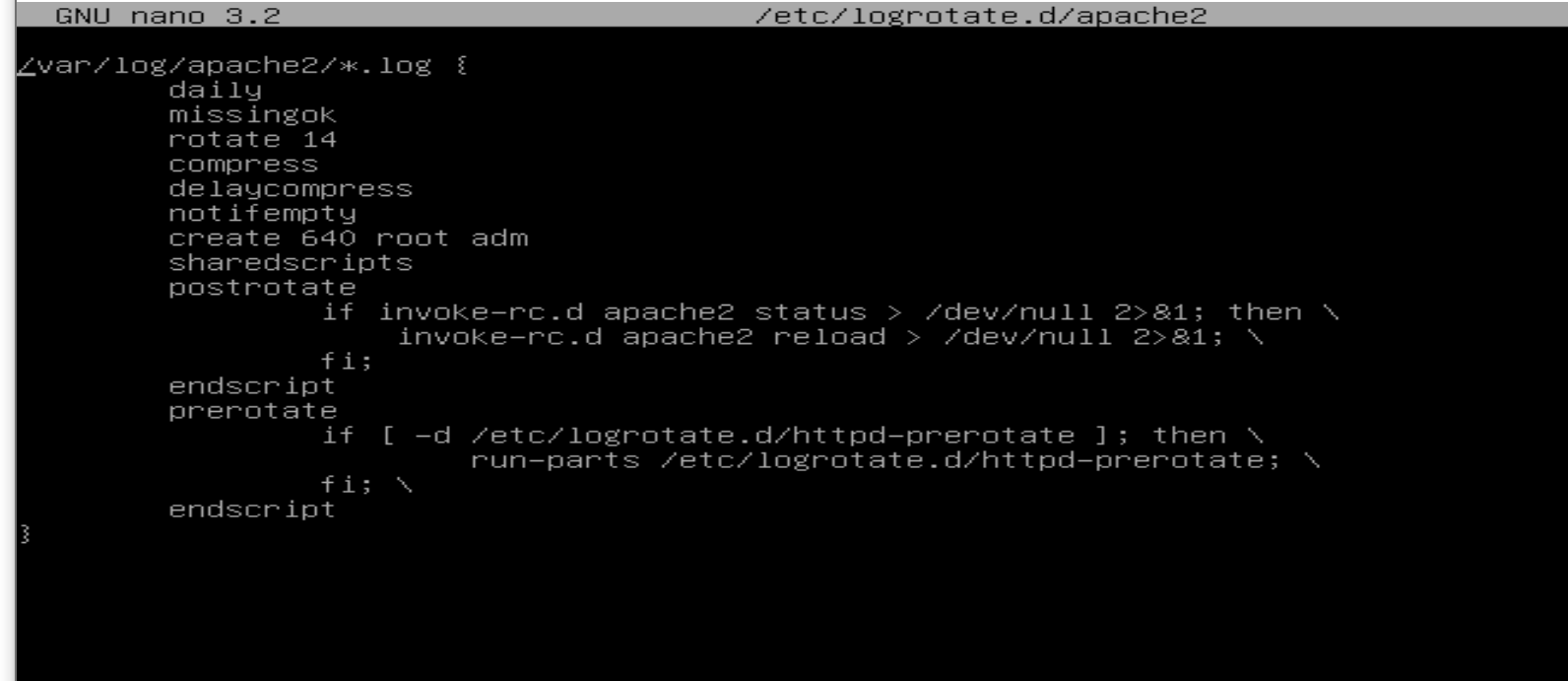
Como vemos funciona correctamente.

1.7 LOGROTATE

Ahora vamos a realizar la rotación de logs esto es una configuración para que los ficheros se vayan cambiando cada cierto tiempo.

Vamos a ver como configurar los ficheros para ello deberemos de acceder al siguiente fichero

/etc/logrotate.d/apache2



Como vemos en este fichero la rotación esta puesta daily(las opciones son daily, weekly o monthly), para que roten los ficheros cada dia, cada semana o cada mes.