UD 4 CONFIGURACIÓN Y ADMINISTRACIÓN DE SERVIDORES

CLARA MONTAÑO RODRÍGUEZ

índice

INTRODUCCIÓN

En este proyecto se desarrolla una **configuración completa y progresiva** de un entorno de **servidores web utilizando Apache sobre una máquina Linux**. A lo largo de las distintas prácticas se ha trabajado con múltiples aspectos fundamentales de la **administración de servidores web**, como la **gestión de puertos**, la **creación de servidores virtuales**, el uso de **alias** y **redirecciones**, la configuración de **documentos de error personalizados**, la **autenticación** de **usuarios** y la aplicación de **restricciones** de acceso **por dirección IP**.

El objetivo principal del trabajo ha sido **comprender el funcionamiento interno** de **Apache** y aprender a controlar su **comportamiento** mediante el **uso de directivas de configuración**, evitando soluciones automáticas o simplificadas. Todas las configuraciones se han realizado directamente desde los **ficheros de configuración del servidor**, sin recurrir al uso de ficheros **.htaccess**, con el fin de aplicar una administración más centralizada, segura y controlada.

Durante el desarrollo del proyecto se han planteado situaciones reales que requieren analizar cómo interactúan entre sí distintas directivas, como **Alias**, **Redirect**, **DirectoryIndex**, **ErrorDocument** o los **mecanismos de autenticación**. Esto ha permitido no solo aplicar configuraciones funcionales, sino también detectar y resolver conflictos derivados de reglas solapadas, herencia de permisos y redirecciones previas.

Este trabajo no se limita a comprobar que los servicios funcionan correctamente, sino que **pone especial énfasis en la seguridad**, la organización de los contenidos y la **correcta documentación de cada paso realizado**. De este modo, el proyecto refleja un proceso de aprendizaje práctico orientado a la administración real de servidores web, siguiendo criterios de orden, coherencia y buenas prácticas en la gestión de Apache.

PRÁCTICA 4. Cambio del puerto del servidor nº 1

Objetivo

Cambiar el puerto de escucha del servidor nº 1 para que escuche por el puerto 7000.

Configuración realizada

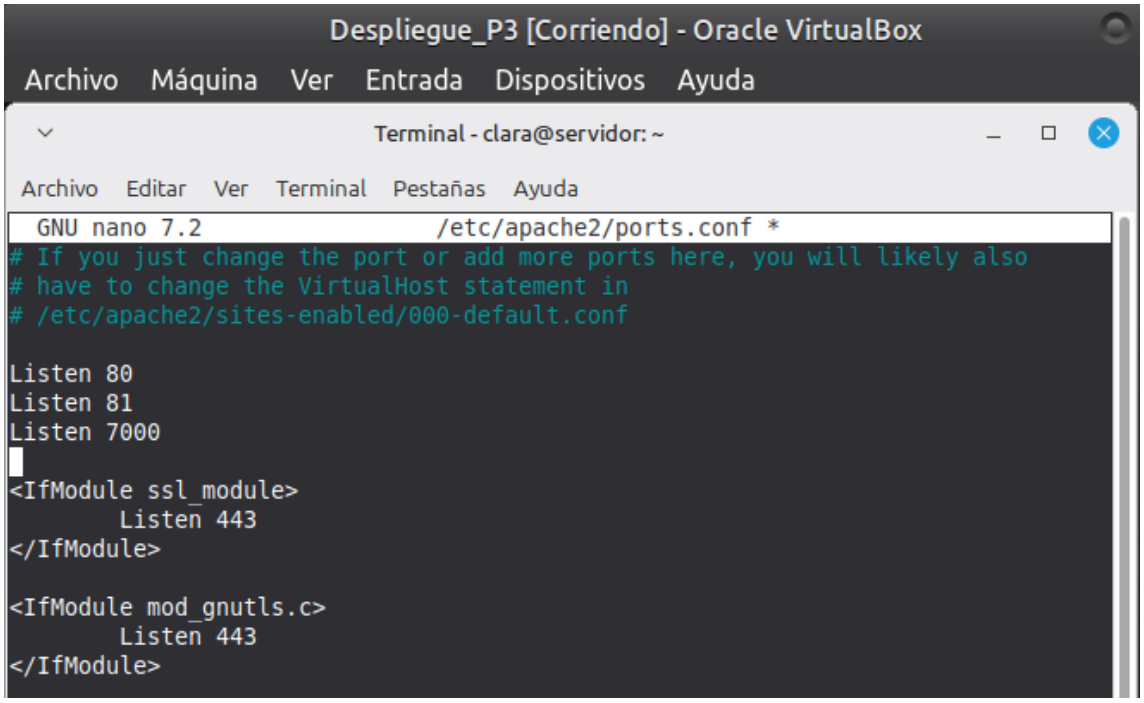
Paso 1: Modificación de los puertos de escucha de Apache.

Editar el fichero de configuración de puertos de Apache:

```
/etc/apache2/ports.conf
```

Añadir la siguiente directiva para habilitar el nuevo puerto:

```
Listen 7000
```



No se deben eliminar los puertos **81** ni **443**, ya que están siendo utilizados por otros servidores web de la infraestructura.

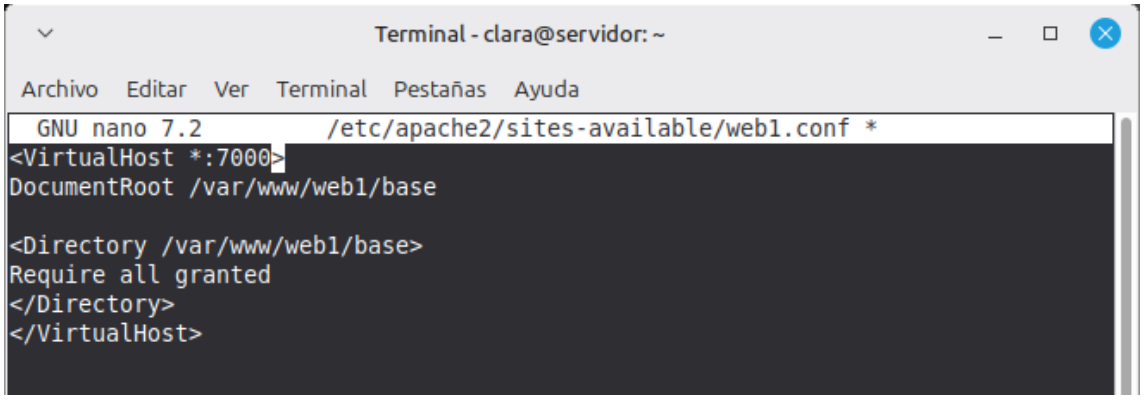
Paso 2: Modificación del VirtualHost del servidor nº 1

Editar el fichero del VirtualHost correspondiente al servidor nº 1:

```
/etc/apache2/sites-available/web1.conf
```

Modificar la directiva `<VirtualHost>` para que el servidor escuche en el nuevo puerto, sustituyendo:

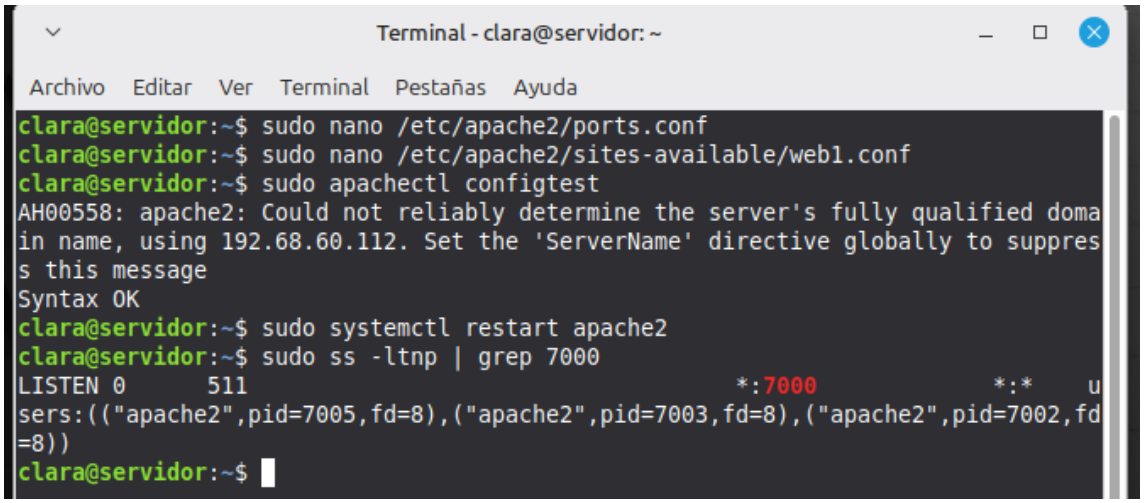
`<VirtualHost *:80>` por: `<VirtualHost *:7000>`



Paso 3: Aplicación de los cambios

Comprobar la sintaxis de la configuración y reiniciar el servicio Apache:

```
sudo apachectl configtest

sudo systemctl restart apache2
```



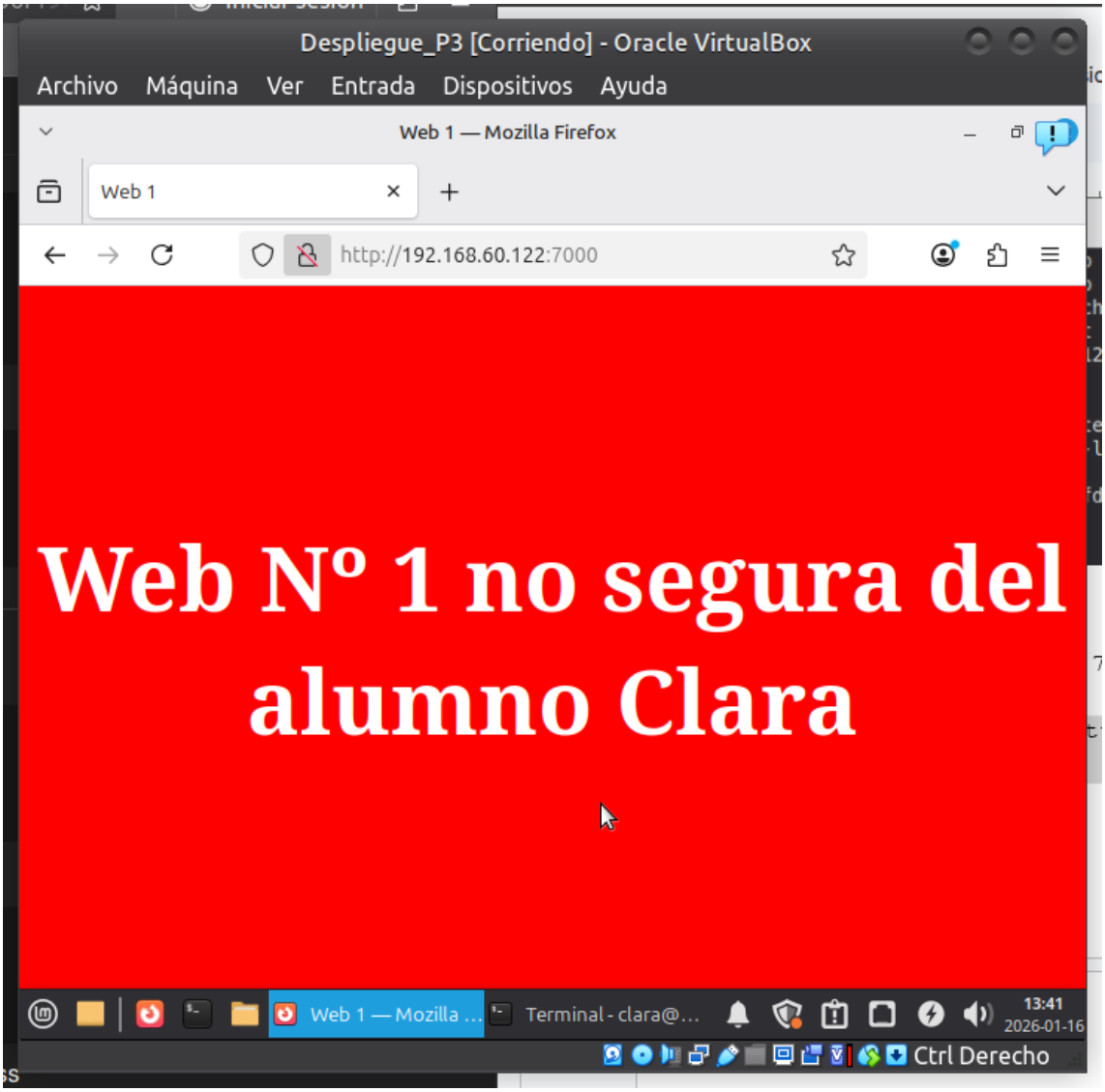
Verificación del funcionamiento

Comprobar que Apache está escuchando en el puerto configurado:

```
sudo ss -ltnp | grep 7000
```

Acceder desde un navegador web a la siguiente dirección:

```
http://192.168.60.122:7000
```



Justificación técnica y de seguridad

La modificación del puerto de escucha permite separar servicios web dentro de una misma máquina y reducir la exposición directa del servicio en el puerto estándar 80.

Esta medida mejora la organización del sistema y facilita una administración más segura y flexible del servidor web

PRÁCTICA 5. Cambio del puerto del servidor web nº 3 (HTTPS)

Objetivo

Cambiar el puerto de escucha del servidor nº 3 (seguro) para que escuche por el puerto 45678.

Configuración realizada

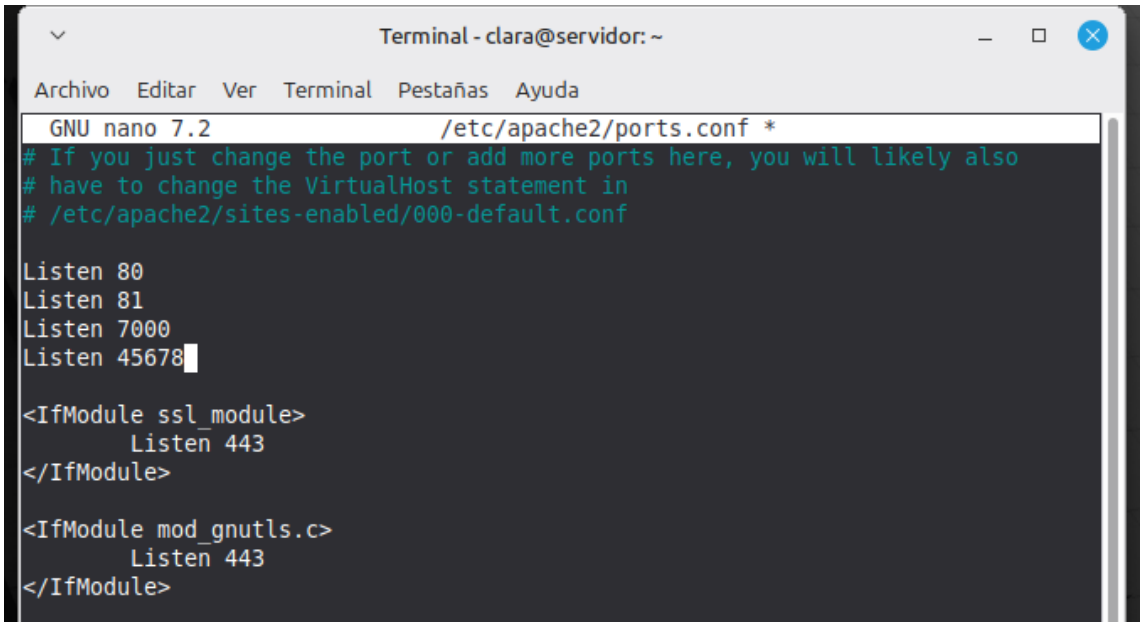
Paso 1: Modificación de los puertos de escucha de Apache

Editar el fichero de configuración de puertos de Apache:

`/etc/apache2/ports.conf`

Añadir la siguiente directiva para habilitar el nuevo puerto HTTPS:

`Listen 45678`



Este cambio se realiza sin eliminar otros puertos, ya que pueden estar siendo utilizados por otros servicios de la infraestructura.

Paso 2: Modificación del VirtualHost HTTPS del servidor nº 3

Editar el fichero del VirtualHost seguro correspondiente al servidor web nº 3:

`sudo nano /etc/apache2/sites-available/web3-ssl.conf`

Modificar la directiva `<VirtualHost>` para que el servidor HTTPS escuche en el nuevo puerto, sustituyendo:

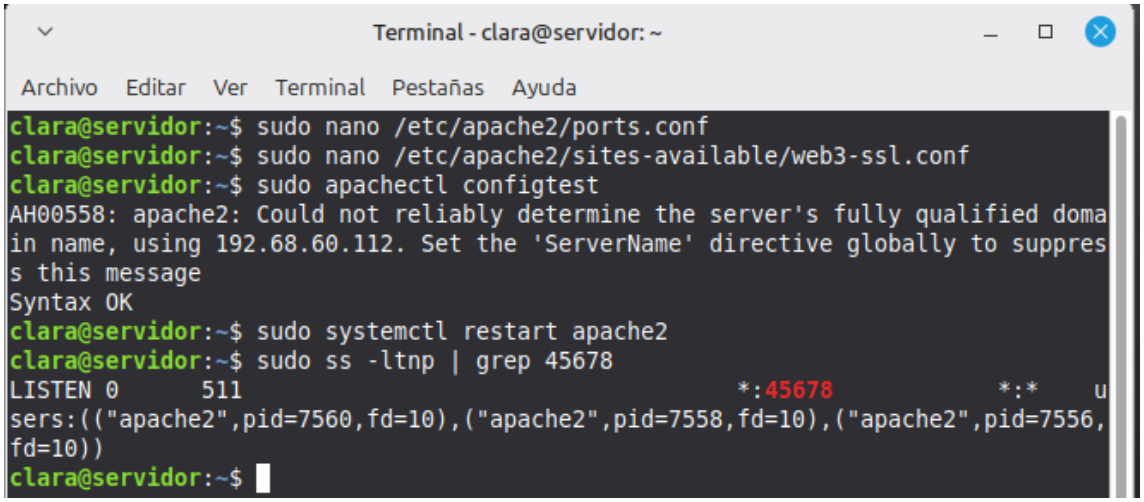
`<VirtualHost *:443>` por: `<VirtualHost *:45678>`

El resto de la configuración SSL (certificados y parámetros de seguridad) se mantiene sin cambios.

Paso 3: Aplicar los cambios

```
sudo apachectl configtest
```

```
sudo systemctl restart apache2
```



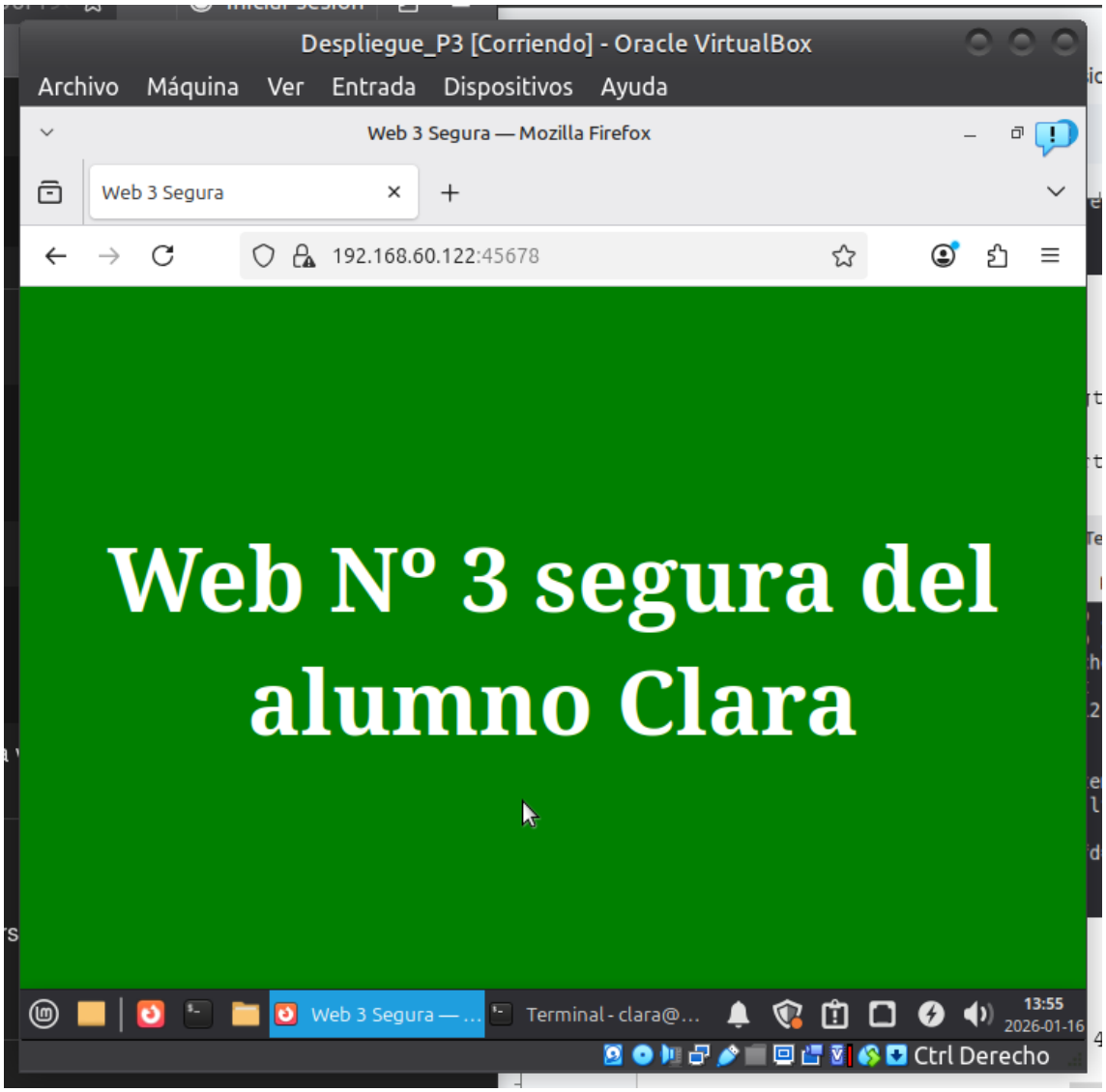
Verificación del funcionamiento

Comprobar que Apache está escuchando en el nuevo puerto HTTPS mediante el comando:

```
sudo ss -ltnp | grep 45678
```

Acceder desde un navegador web a la siguiente dirección:

`https://192.168.60.122:45678`



Si la página del servidor web nº 3 se muestra correctamente y el navegador indica una conexión segura, la configuración se ha aplicado correctamente.

Justificación técnica y de seguridad

El uso de un puerto HTTPS personalizado permite separar servicios seguros dentro de la misma máquina, facilitando la administración y evitando conflictos con otros servicios que puedan utilizar el puerto 443.

Además, esta configuración reduce la exposición directa del servicio HTTPS frente a accesos automatizados y contribuye a una mejor organización de la infraestructura web.

PRÁCTICA 6. Configuración del documento inicial del servidor nº 1

Objetivo

Al acceder al directorio base del servidor nº 1, redireccionar automáticamente al fichero: /documentos/empiezo_aqui.html, que mostrará el mensaje: "Estoy empezando desde este lugar"

Configuración realizada

Paso 1: Comprobación del documento índice existente

Se comprueba si existe un fichero index.html en el directorio raíz del servidor web nº 1.

En caso de existir, se renombra para evitar que Apache lo cargue como documento inicial por defecto y poder verificar correctamente la configuración aplicada.

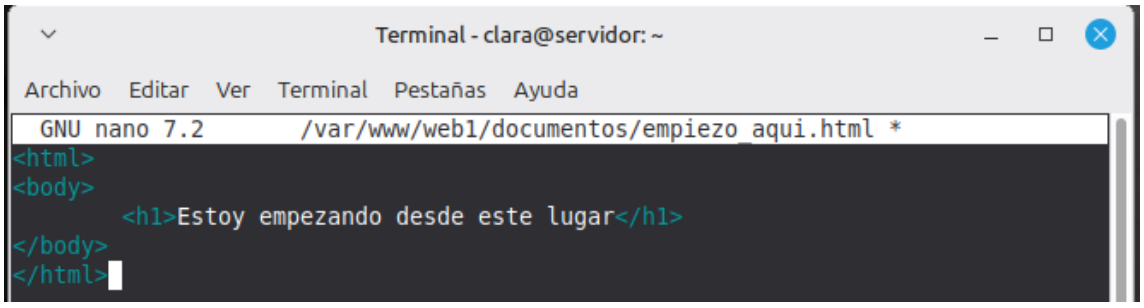
Paso 2: Creación del fichero HTML

Se crea el fichero empiezo_aqui.html en el directorio correspondiente:

```
/var/www/web1/documentos/empiezo_aqui.html
```

El contenido del fichero debe ser el siguiente:

```
<html>
    <body>
      <h1>Estoy empezando desde este lugar</h1>
    </body>
</html>
```



Paso 3: Modificación del VirtualHost del servidor nº 1

Editar el fichero de configuración del VirtualHost del servidor web nº 1:

```
/etc/apache2/sites-available/web1.conf
```

Dentro del bloque `<VirtualHost>`, añadir la siguiente directiva:

```
DirectoryIndex empiezo_aqui.html
```

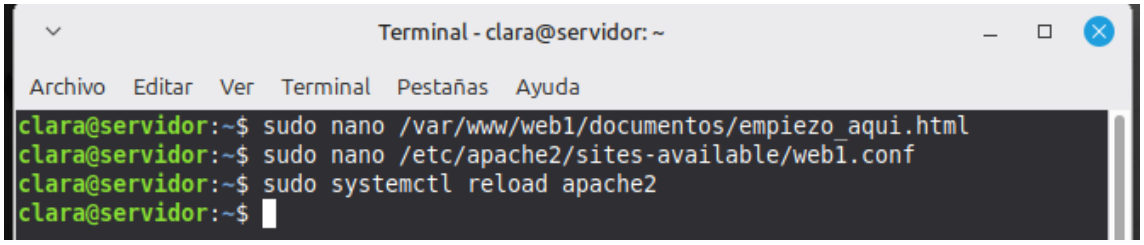
Con esta directiva se indica a Apache qué fichero debe servir como documento inicial al acceder a la raíz del servidor.

Para que Apache pueda cargar correctamente este fichero como documento inicial, el archivo empiezo_aqui.html se encuentra accesible desde el directorio base del servidor web nº 1, correspondiente a su DocumentRoot.

Paso 4: Aplicación de los cambios

```
sudo apachectl configtest
```

```
sudo systemctl restart apache2
```



Verificación del funcionamiento

Acceder desde un navegador web a la dirección del servidor nº 1:

http://192.168.60.122:7000

El servidor muestra automáticamente el contenido del fichero empiezo_aqui.html, confirmando que el documento inicial se ha configurado correctamente.

Justificación técnica y de seguridad

La configuración del documento inicial permite controlar el punto de entrada al servidor web, guiando al usuario hacia el recurso deseado.

Esta medida facilita la organización de los contenidos y mejora la administración del servidor, evitando accesos no deseados a otros ficheros o directorios.

PRÁCTICA 7. Redirección al servidor web nº 3 (directorio imágenes)

Objetivo

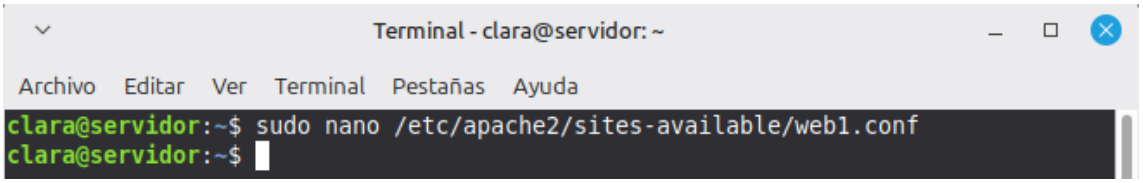
Al acceder al directorio base del servidor nº 1, redireccionar automáticamente al directorio imágenes del servidor nº 3, mostrando el listado de su contenido.

Configuración realizada

Paso 1: Configuración de la redirección en el servidor nº 1

Editar el fichero del VirtualHost del servidor web nº 1:

```
sudo nano /etc/apache2/sites-available/web1.conf
```

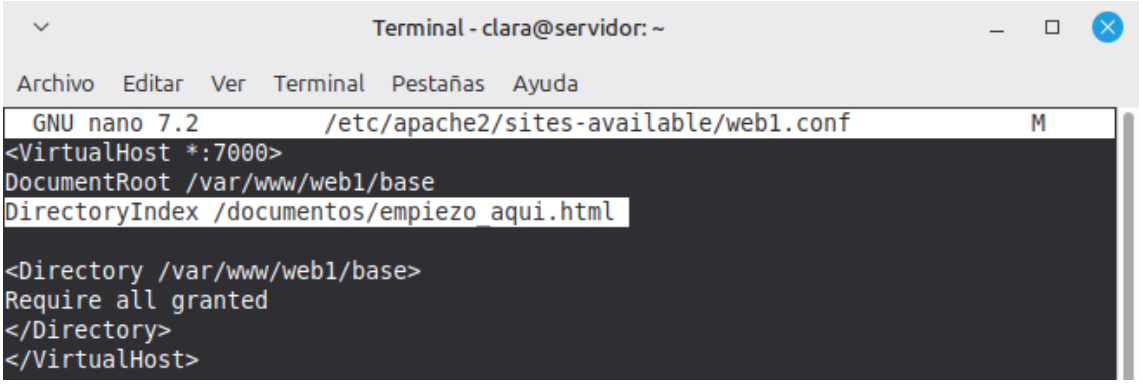


Dentro del bloque `<VirtualHost *:7000> ... </VirtualHost>`, eliminar la siguiente línea de configuración utilizada en la práctica anterior:

```
DirectoryIndex /documentos/empiezo_aqui.html
```

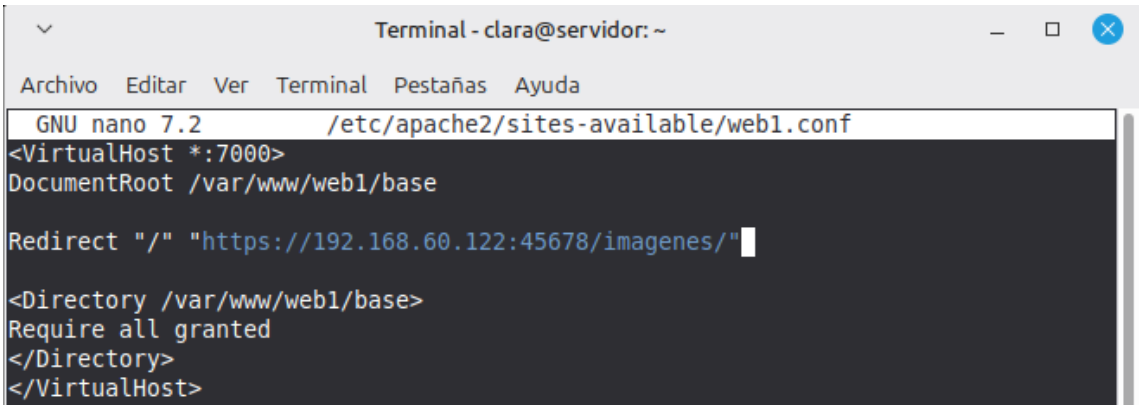
A continuación, añadir la siguiente directiva de redirección, situada después de `DocumentRoot`:

```
DirectoryIndex /documentos/empiezo_aqui.html
```



A continuación, añadir la siguiente directiva debajo de DocumentRoot:

```
Redirect "/" "https://192.168.60.122:45678/imagenes/"
```



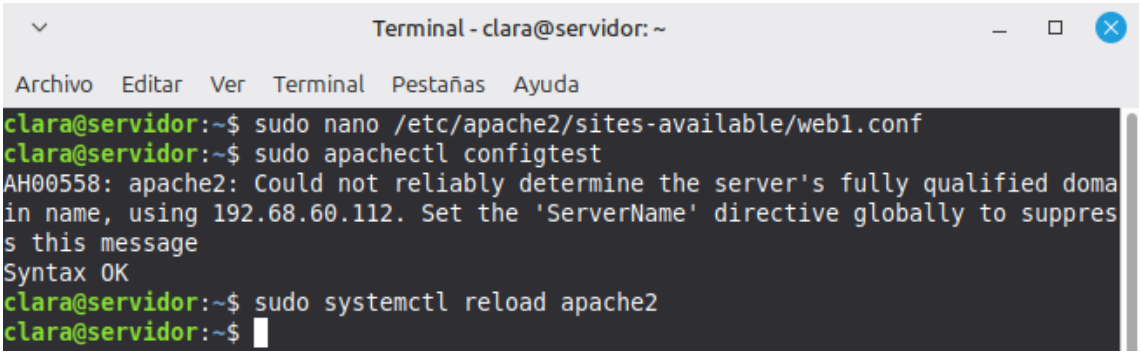
Con esta directiva se fuerza la redirección automática desde el servidor web nº 1 hacia el servidor web nº 3 mediante HTTPS.

Paso 2: Comprobación de la sintaxis y aplicación de los cambios

Verificar que la configuración no contiene errores y recargar el servicio Apache:

```
sudo apachectl configtest
```

```
sudo systemctl reload apache2
```



Verificación del funcionamiento

Acceder desde el navegador web a la siguiente dirección:

```
http://192.168.60.122:7000
```

El acceso debe redirigir automáticamente a:

```
https://192.168.60.122:45678/imagenes/
```

En este punto, el navegador puede mostrar inicialmente un error **404 Not Found**.

Análisis del problema

El error 404 no se debe a un fallo en la redirección, ya que:

- El servidor web nº 3 está operativo.
- El puerto HTTPS 45678 se encuentra activo y accesible.
- La redirección desde el servidor web nº 1 se realiza correctamente.

El problema se produce porque, aunque el directorio físico:

`/var/www/web3/imagenes`

Solución aplicada

Paso 3: Edición del VirtualHost HTTPS del servidor nº 3

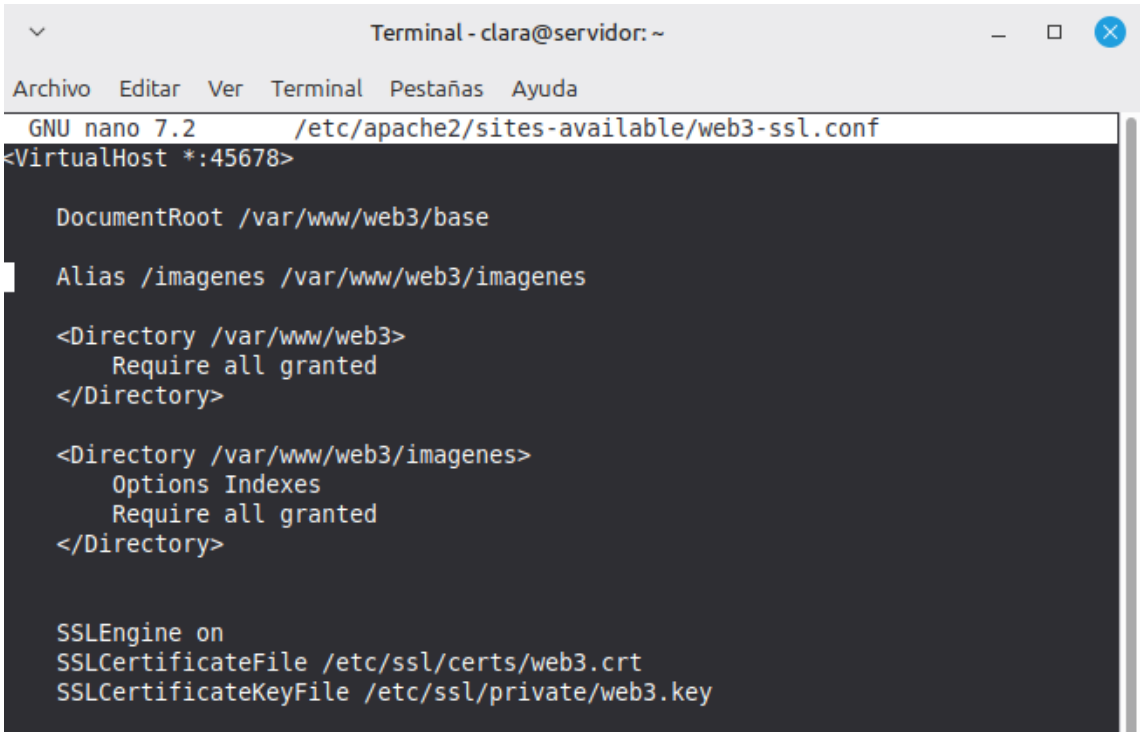
`Editar el fichero de configuración del servidor web seguro:`

`sudo nano /etc/apache2/sites-available/web3-ssl.conf`

Paso 4: Asociación de la URL y habilitación del listado de directorios

Dentro del bloque `<VirtualHost *:45678>`, añadir la siguiente configuración, sin modificar el resto del VirtualHost:

```
Alias /imagenes /var/www/web3/imagenes
<Directory /var/www/web3/imagenes>
      Options Indexes
      Require all granted
</Directory>
```



Con esta configuración:

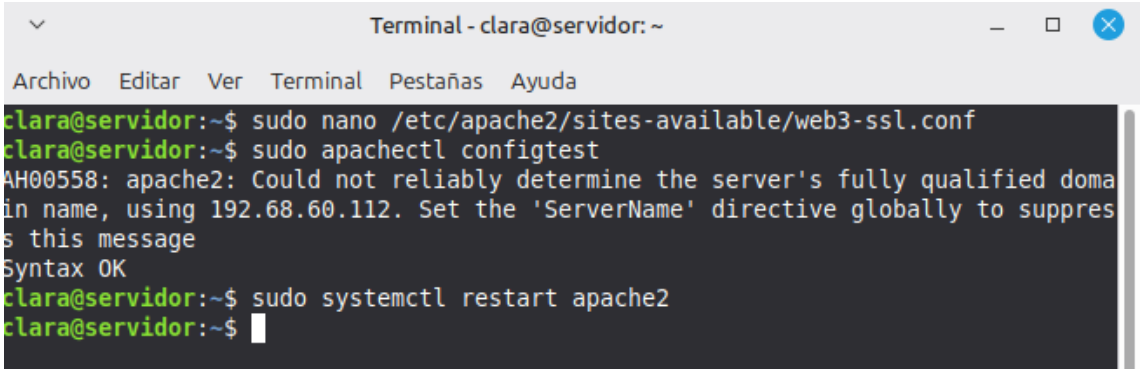
- La directiva Alias asocia la URL /imagenes al directorio físico correspondiente.
- La opción Indexes permite mostrar el listado del contenido del directorio cuando no existe un fichero index.html.

Paso 5: Comprobación y aplicación de los cambios

Comprobar la configuración y reiniciar el servicio Apache:

```
sudo apachectl configtest
```

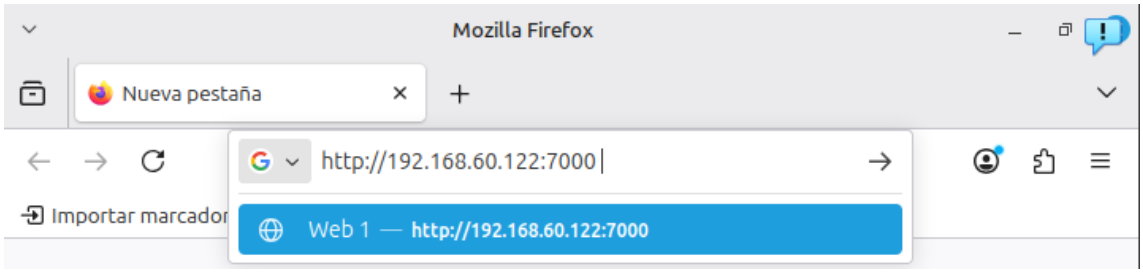
```
sudo systemctl reload apache2
```



Verificación final

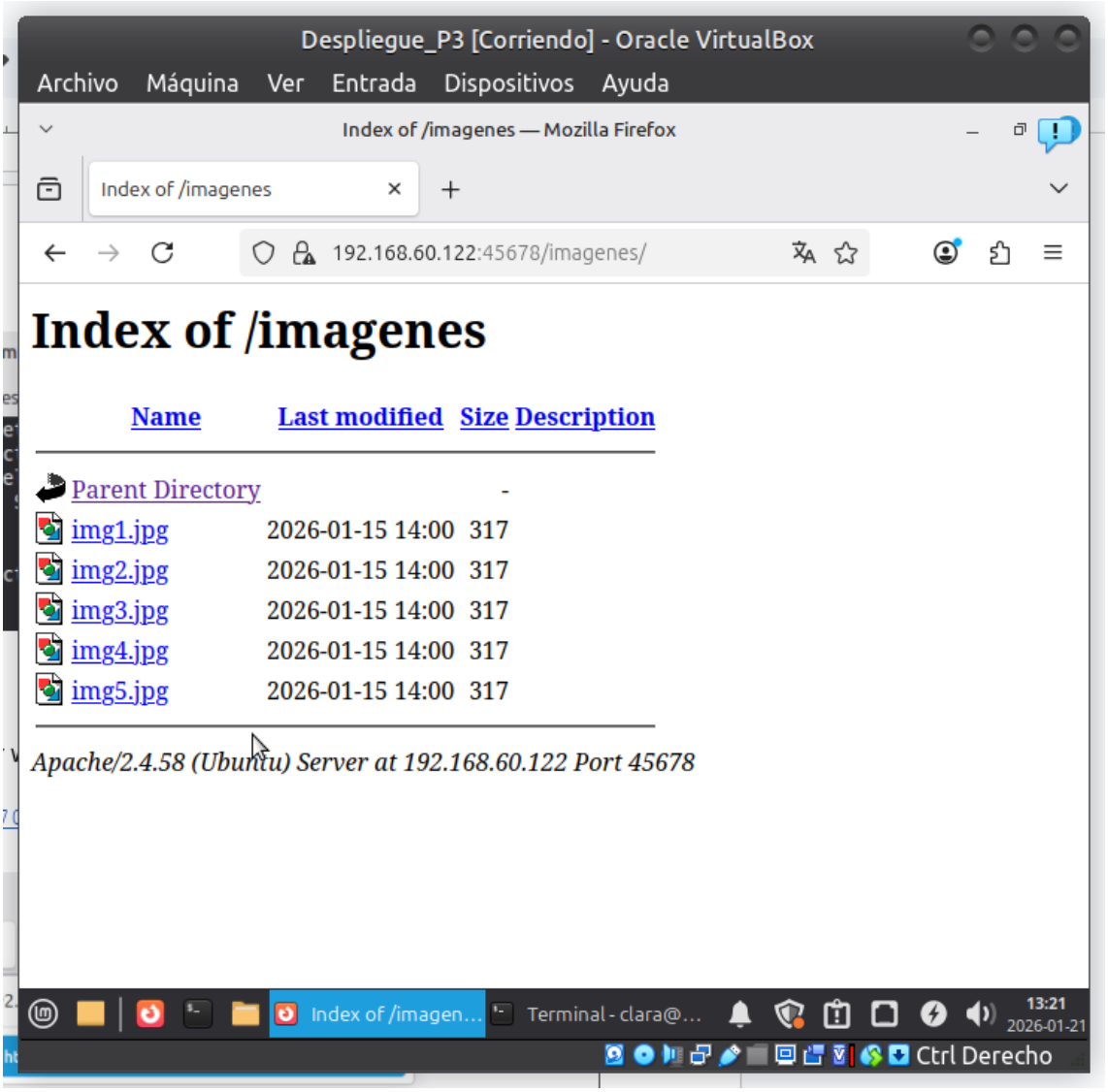
Acceder de nuevo desde el navegador a:

```
http://192.168.60.122:7000
```



El sistema redirige automáticamente a:

```
https://192.168.60.122:45678/imagenes/
```



Justificación técnica y de seguridad

Esta configuración permite centralizar los recursos gráficos en el servidor web seguro, forzando el acceso mediante HTTPS y garantizando una transmisión cifrada de los datos.

El uso de Alias evita la duplicación de contenidos y mejora la organización del sistema, mientras que la habilitación del listado de directorios se limita únicamente al recurso necesario, manteniendo un equilibrio entre funcionalidad y seguridad.

PRÁCTICA 8. Permitir el acceso al servidor web mediante IP

Objetivo

Permitir que todos los puestos de la clase, incluido el puesto del profesor, puedan acceder al servidor web mediante IP.

Configuración realizada

Paso 1: Edición de los VirtualHost

Se comprueba que cada servidor web tenga definida una directiva de acceso que permita la conexión desde cualquier equipo de la red.

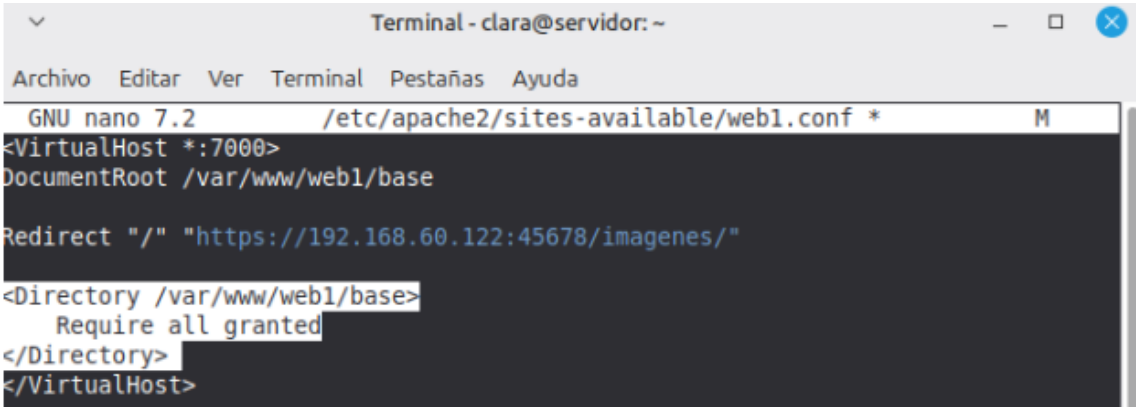
Servidor web nº 1

```
sudo nano /etc/apache2/sites-available/web1.conf
```

Editar el fichero de configuración del servidor web nº 1:

Dentro del bloque `<VirtualHost>`, comprobar o añadir la siguiente configuración:

```
<Directory /var/www/web1>
      Require all granted
</Directory>
```



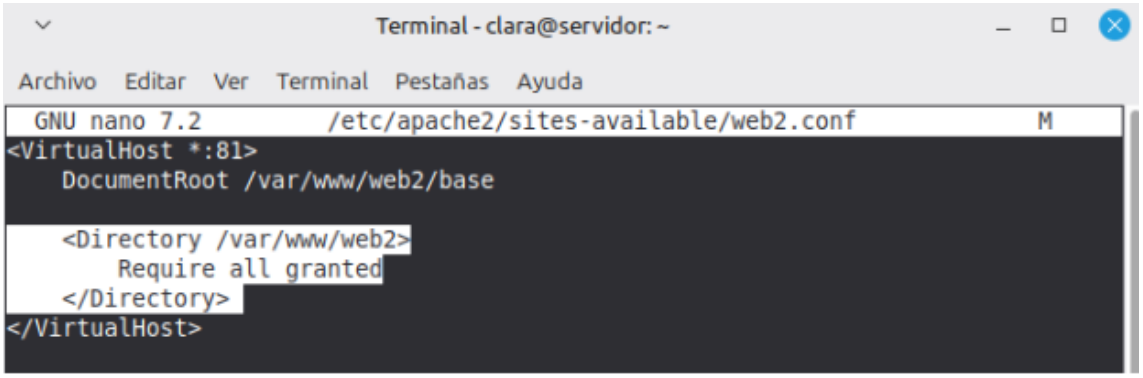
Servidor web nº 2

Editar el fichero de configuración del servidor web nº 2:

```
sudo nano /etc/apache2/sites-available/web2.conf
```

Dentro del bloque `<VirtualHost>`, comprobar o añadir la siguiente configuración:

```
<Directory /var/www/web2>
      Require all granted
</Directory>
```



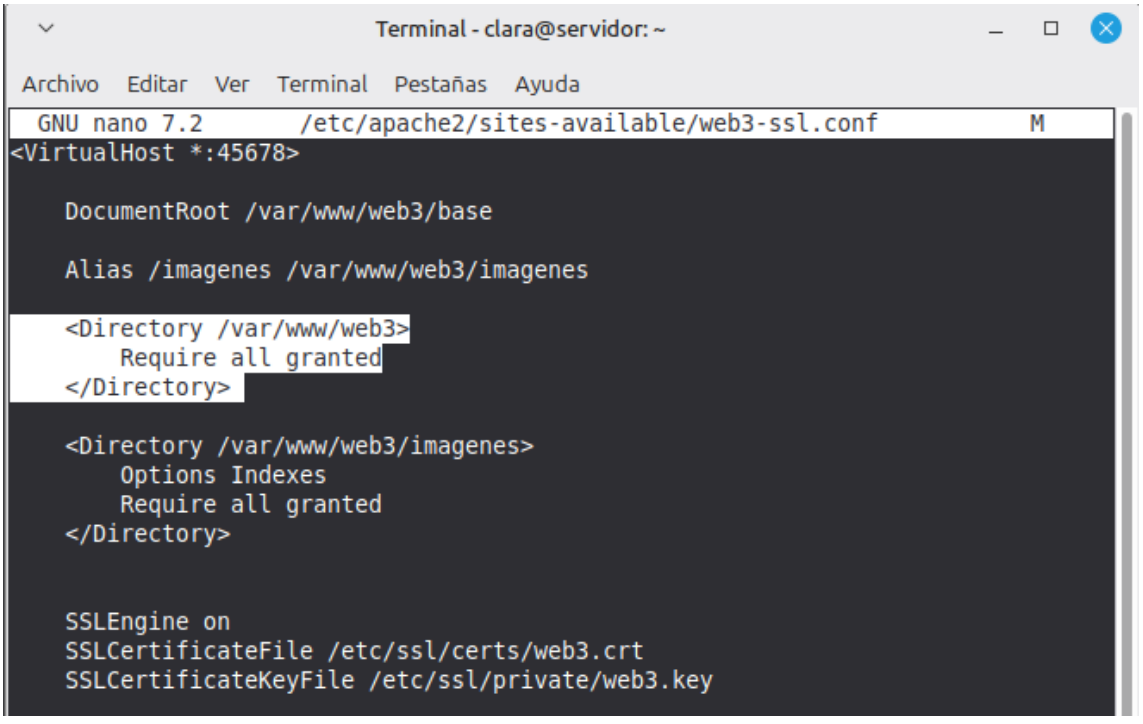
Servidor web nº 3

Editar el fichero de configuración del servidor web nº 3:

```
sudo nano /etc/apache2/sites-available/web3.conf
```

Dentro del bloque `<VirtualHost>`, comprobar o añadir la siguiente configuración:

```
<Directory /var/www/web3>
      Require all granted
</Directory>
```



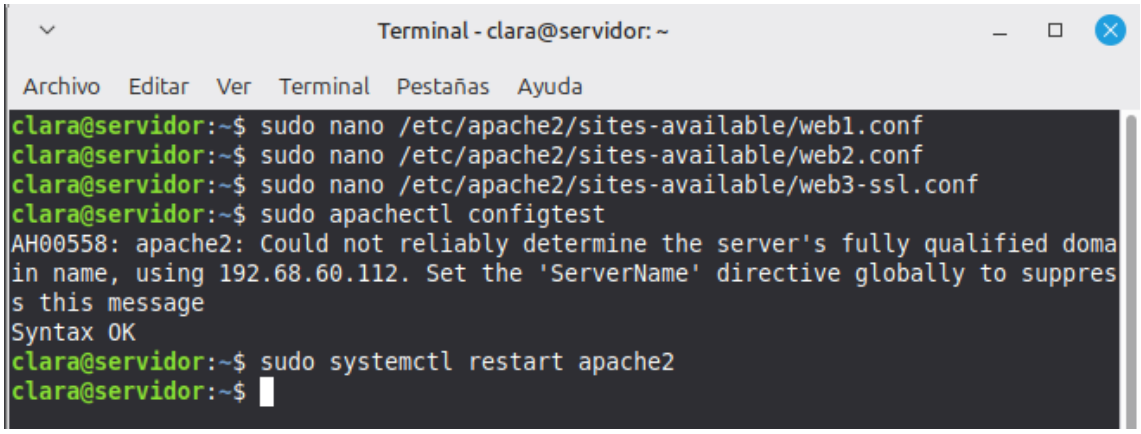
Esta directiva permite el acceso al contenido del servidor web desde cualquier dirección IP de la red.

Paso 2: Aplicación de los cambios

Comprobar la sintaxis de la configuración y reiniciar el servicio Apache:

```
sudo apachectl configtest
```

```
sudo systemctl restart apache2
```

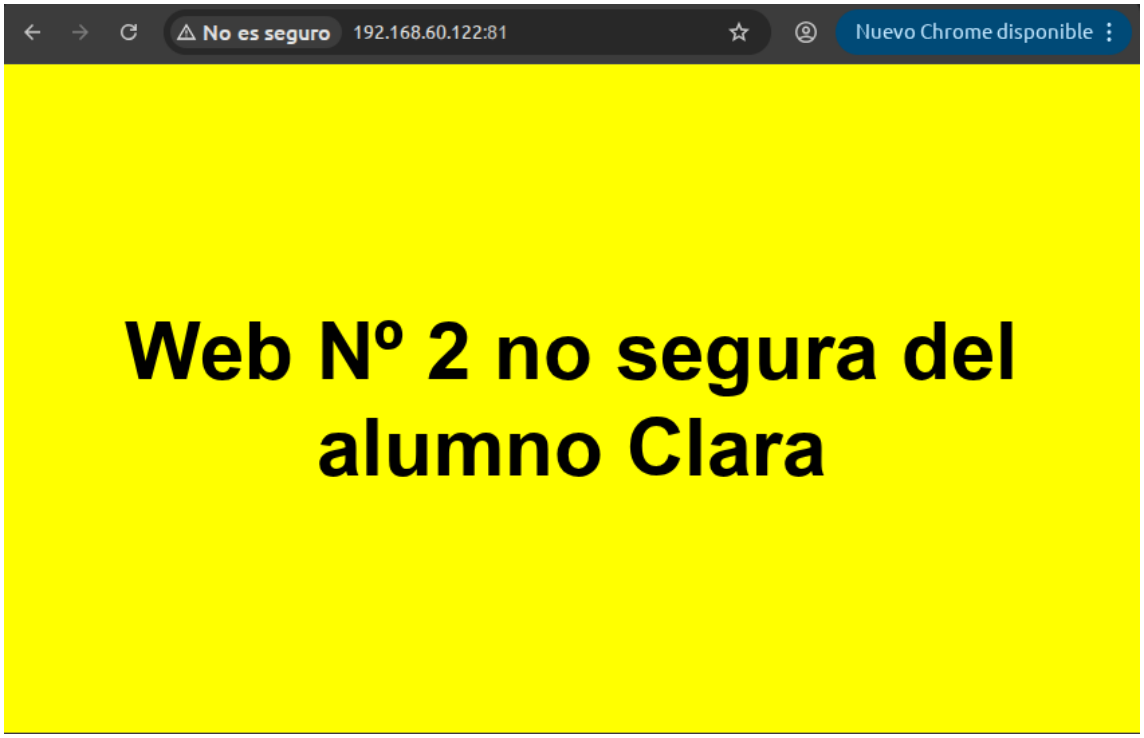


Verificación del funcionamiento

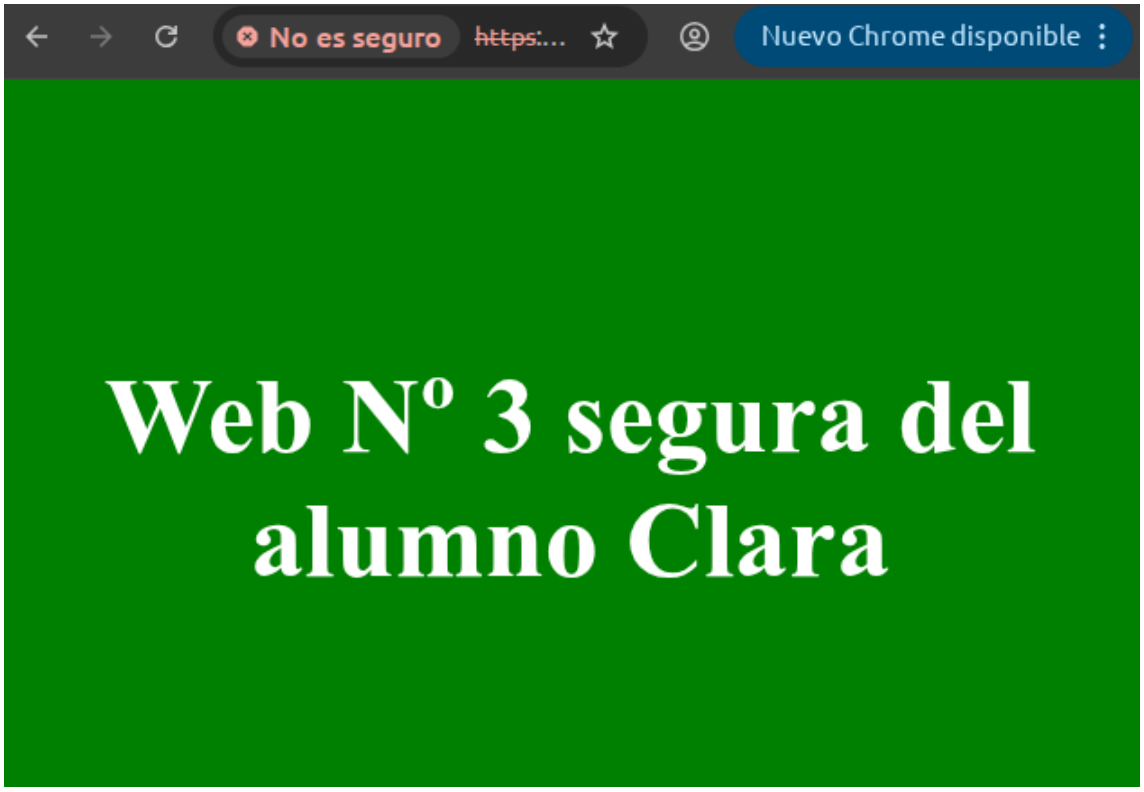
Desde distintos equipos de la red (puestos de alumnos y puesto del profesor), se accede mediante navegador a las siguientes direcciones:

```
http://192.168.60.122:7000
```

`http://192.168.60.122:81`



`https://192.168.60.122:45678`



En este caso, la comprobación se realiza desde la máquina local.
El acceso se realiza correctamente desde cualquier IP de la red, sin mostrarse mensajes de bloqueo o restricción.

Justificación técnica y de seguridad

Esta configuración garantiza la accesibilidad completa de los servidores web dentro de la red de la clase, permitiendo que todos los puestos autorizados accedan a los servicios web sin restricciones innecesarias.

El control de acceso se gestiona de forma centralizada desde los ficheros de configuración de Apache, lo que mejora la administración del servidor y evita el uso de configuraciones dispersas o potencialmente inseguras mediante ficheros .htaccess.

PRÁCTICA 9. Ajuste del ancho del listado de directorios en el servidor nº 2

Objetivo

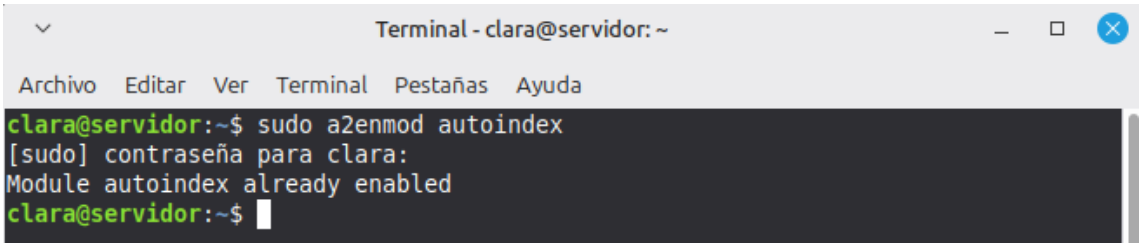
Ajustar el ancho de presentación del listado del directorio documentos del servidor nº 2, de forma que se muestren correctamente los nombres de los ficheros.

Configuración realizada

Paso 1: Activación del módulo autoindex

Activar el módulo autoindex, necesario para mostrar listados de directorios en Apache:

```
sudo a2enmod autoindex
```



Paso 2: Edición del VirtualHost del servidor web nº 2

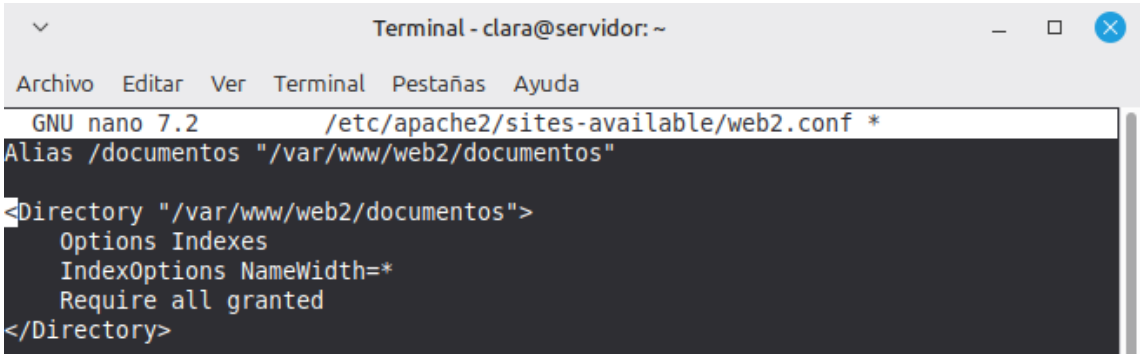
Editar el fichero de configuración del servidor web nº 2:

Dentro del bloque `<VirtualHost *:81>`, añadir la siguiente configuración sin modificar ni eliminar ninguna directiva existente, destinada a permitir el acceso al directorio documentos y ajustar el ancho de presentación del listado:

```
Alias /documentos "/var/www/web2/documentos"
<Directory "/var/www/web2/documentos">
    Options Indexes
    IndexOptions NameWidth=*
    Require all granted
</Directory>
```

Esta configuración realiza las siguientes acciones:

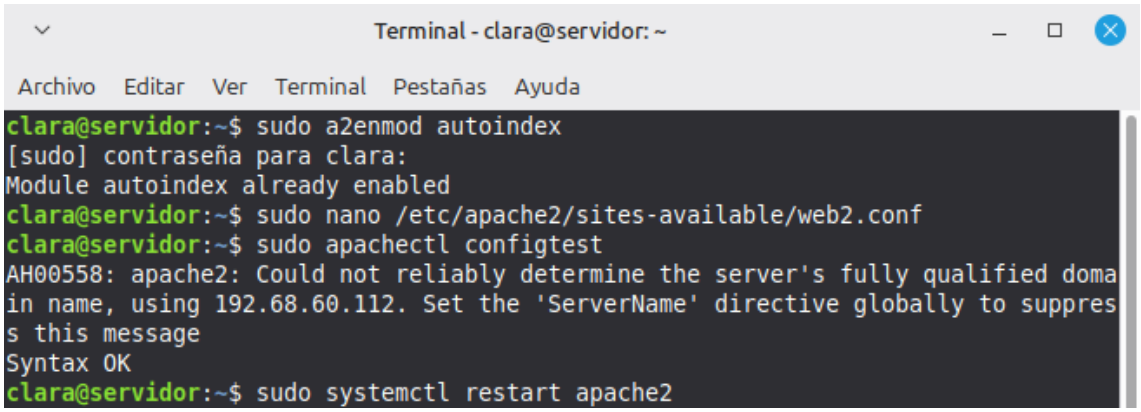
- Asocia la URL `/documentos` con el directorio físico `/var/www/web2/documentos`.
- Habilita el listado del contenido del directorio.
- Ajusta el ancho de los nombres de los ficheros para que se muestren completos.
- Permite el acceso al recurso desde cualquier IP de la red.



Paso 3: Comprobación y reinicio del servicio Apache

Verificar que la configuración es correcta y reiniciar Apache para aplicar los cambios:

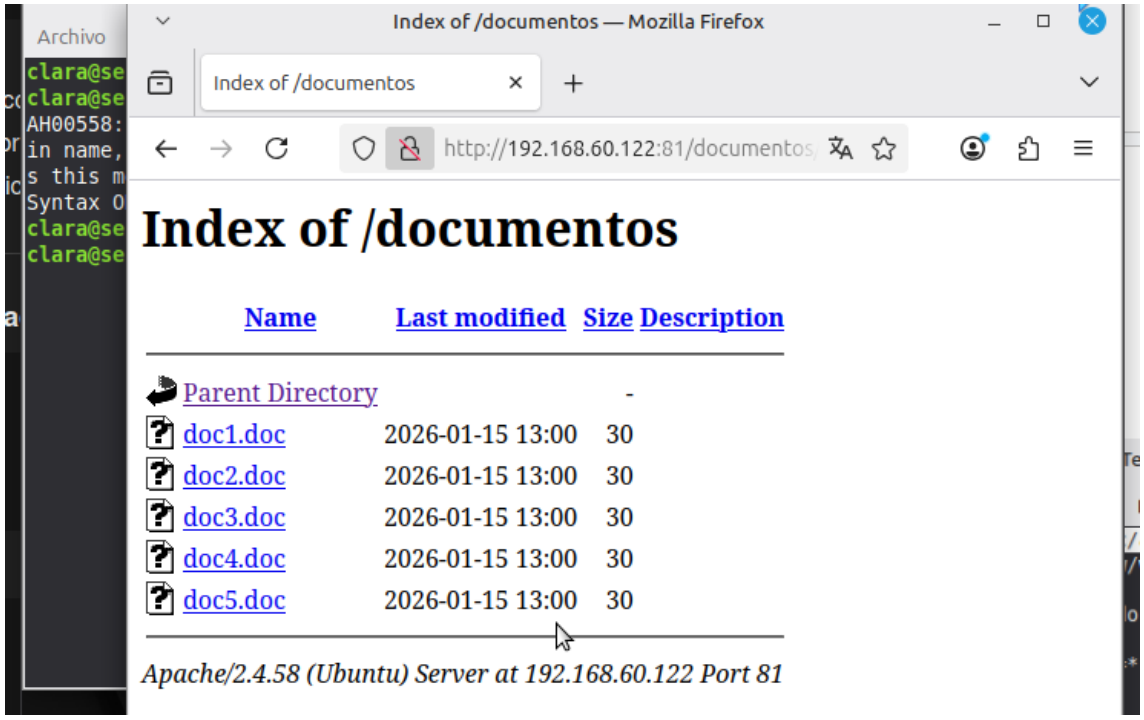
```
sudo apachectl configtest

sudo systemctl restart apache2
```



Verificación del funcionamiento

Desde el navegador web, acceder al servidor web nº 2 y a su directorio documentos:

```
http://192.168.60.122:81/documentos/
```



El listado del directorio se muestra correctamente, con los nombres de los ficheros completos y sin cortes ni solapamientos.

Justificación técnica y de seguridad

El uso de la directiva IndexOptions NameWidth=* mejora la usabilidad y legibilidad del servidor web al mostrar correctamente los nombres de los ficheros en los listados de directorios.

Esta configuración se aplica de forma controlada y explícita desde los ficheros de configuración de Apache, manteniendo una administración clara y segura del servidor web y evitando soluciones improvisadas.

PRÁCTICA 10. Protección del directorio fotos_protegidas (Servidor nº 3)

Objetivo

Proteger el acceso al directorio imagenes/fotos_protegidas del servidor nº 3, que deberá crearse previamente, usando:

- Usuario: doraemon
- Contraseña: nobita2014#(12@

Este directorio contendrá 5 imágenes de Doraemon y Nobita.

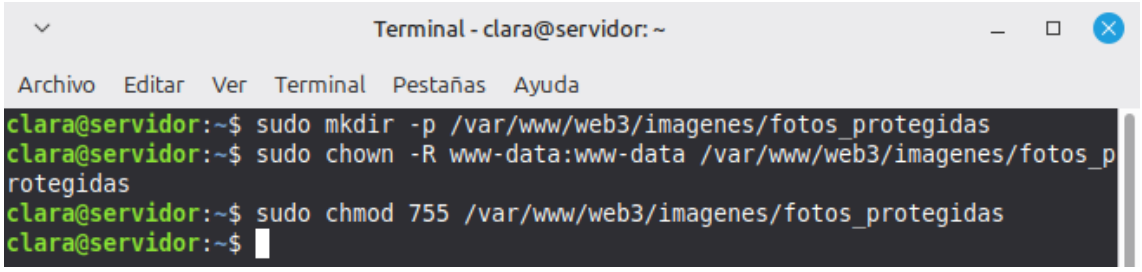
Configuración realizada

Paso 1: Creación del directorio fotos_protegidas

```
sudo mkdir -p /var/www/web3/imagenes/fotos_protegidas
```

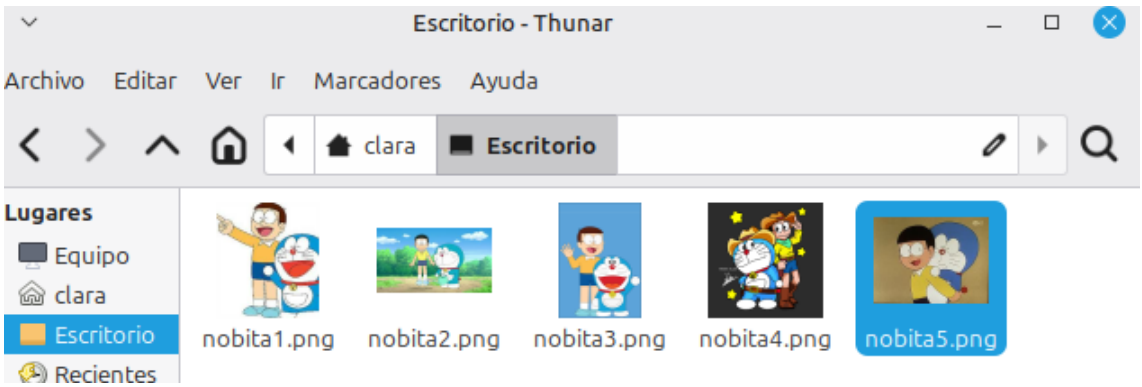
Asignar los permisos adecuados en caso de que no se hereden correctamente:

```
sudo chown -R www-data:www-data
/var/www/web3/imagenes/fotos_protegidas

sudo chmod 755 /var/www/web3/imagenes/fotos_protegidas
```



Paso 2: Añadir las imágenes obligatorias

Descargar en el escritorio cinco imágenes de Doraemon y Nobita.



Copiar las imágenes al directorio protegido sin eliminar los archivos originales:

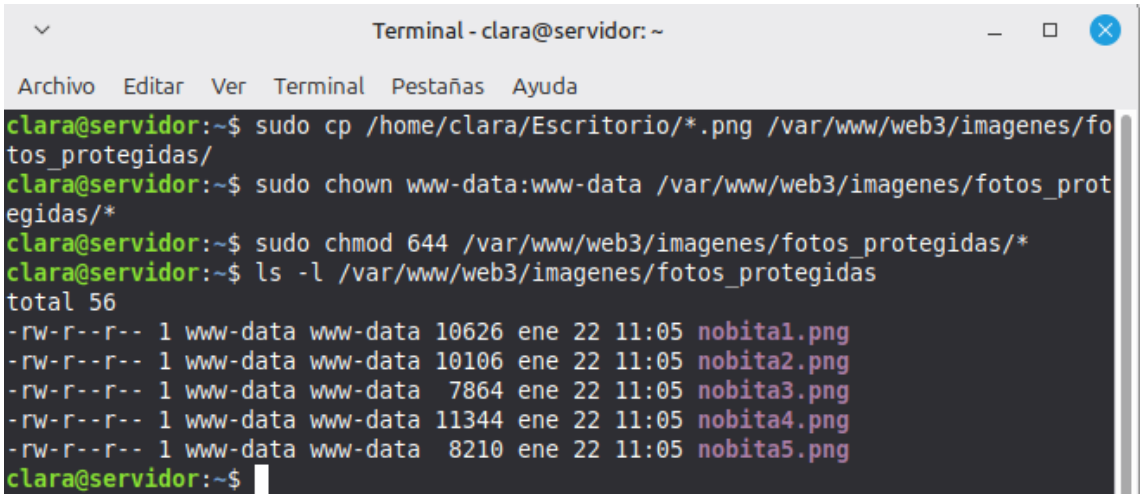
```
sudo cp /home/clara/Escritorio/*.png
/var/www/web3/imagenes/fotos_protegidas/
```

Ajustar los permisos de los ficheros copiados:

```
sudo chown www-data:www-data /var/www/web3/imagenes/fotos_protegidas/*

sudo chmod 644 /var/www/web3/imagenes/fotos_protegidas/*
```

Comprobar que las imágenes existen en el directorio:

```
ls -l /var/www/web3/imagenes/fotos_protegidas
```



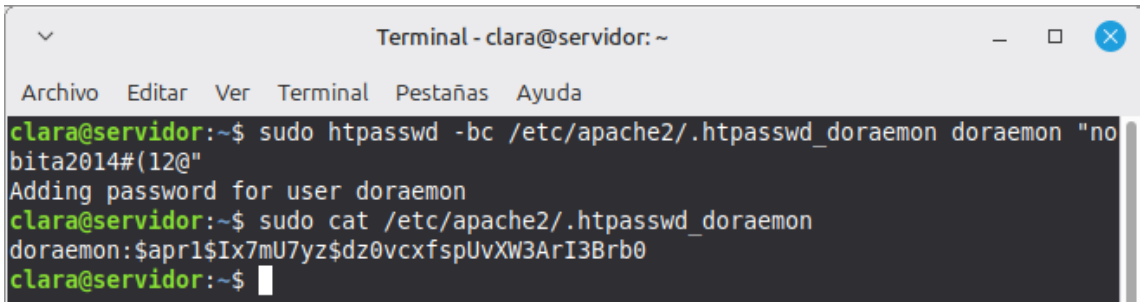
Paso 3: Crear el usuario y la contraseña

Para proteger el acceso al directorio, crear un fichero de autenticación con el usuario indicado en el enunciado:

```
sudo htpasswd -bc /etc/apache2/.htpasswd_doraemon doraemon
"nobita2014#(12@"
```

Comprobar el contenido del fichero de autenticación:

```
sudo cat /etc/apache2/.htpasswd_doraemon
```



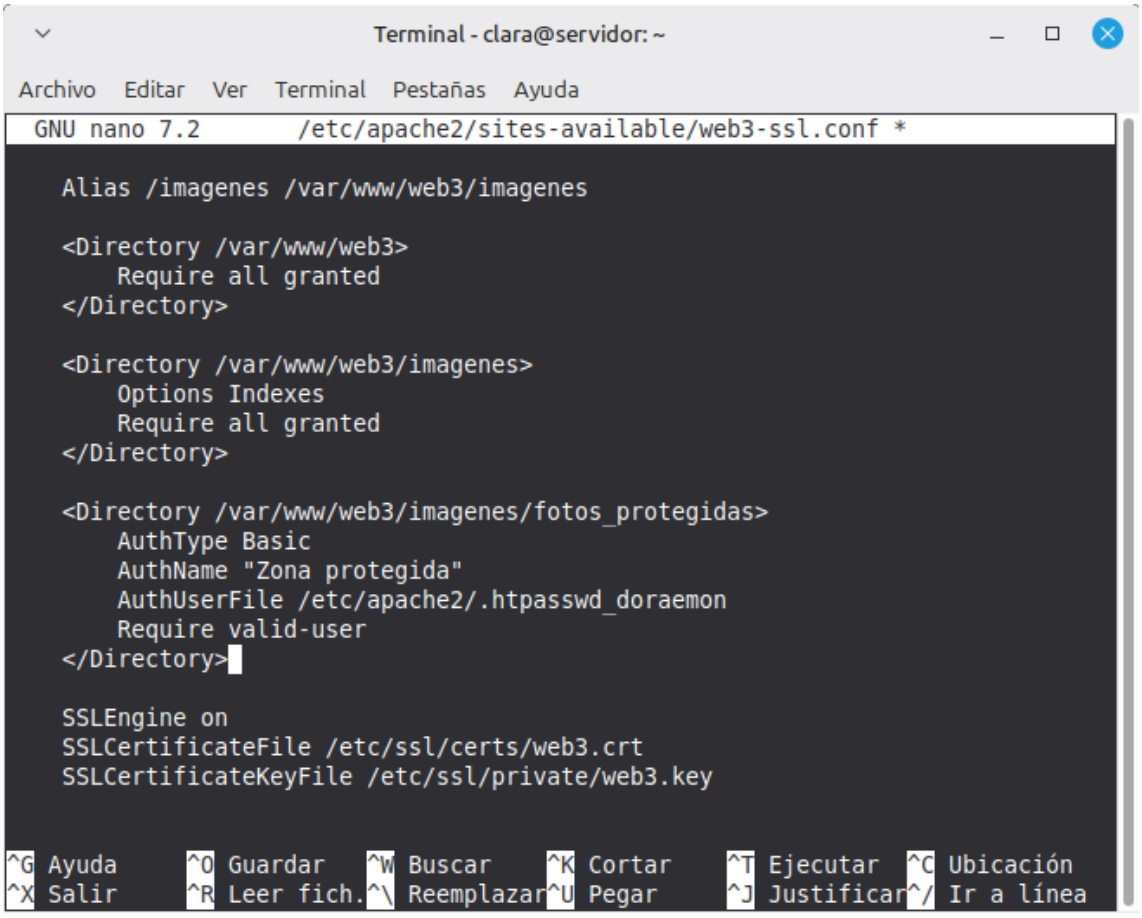
Paso 4: Protección del directorio desde Apache

Editar el VirtualHost seguro del servidor web nº 3:

```
sudo nano /etc/apache2/sites-available/web3-ssl.conf
```

Dentro del bloque `<VirtualHost *:45678>`, añadir el siguiente bloque **sin modificar el resto del fichero**:

```
<Directory /var/www/web3/imagenes/fotos_protegidas>
      AuthType Basic
      AuthName "Zona protegida"
      AuthUserFile /etc/apache2/.htpasswd_doraemon
      Require valid-user
</Directory>
```



Este bloque indica que:

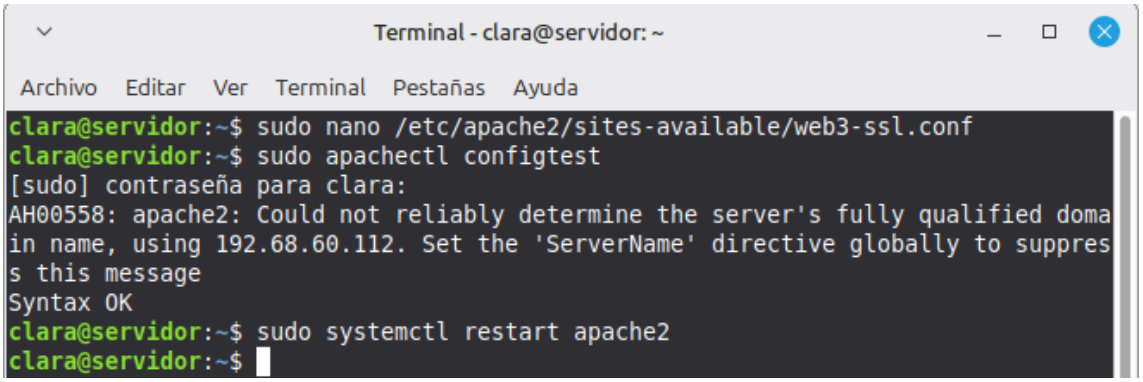
- Se utiliza autenticación básica.
- Se muestra el mensaje **“Zona protegida”** al solicitar las credenciales.
- Solo los usuarios definidos en el fichero .htpasswd_doraemon pueden acceder al directorio.

Paso 5: Aplicación de la configuración

Comprobar la sintaxis y reiniciar el servicio Apache:

```
sudo apachectl configtest
```

```
sudo systemctl restart apache2
```

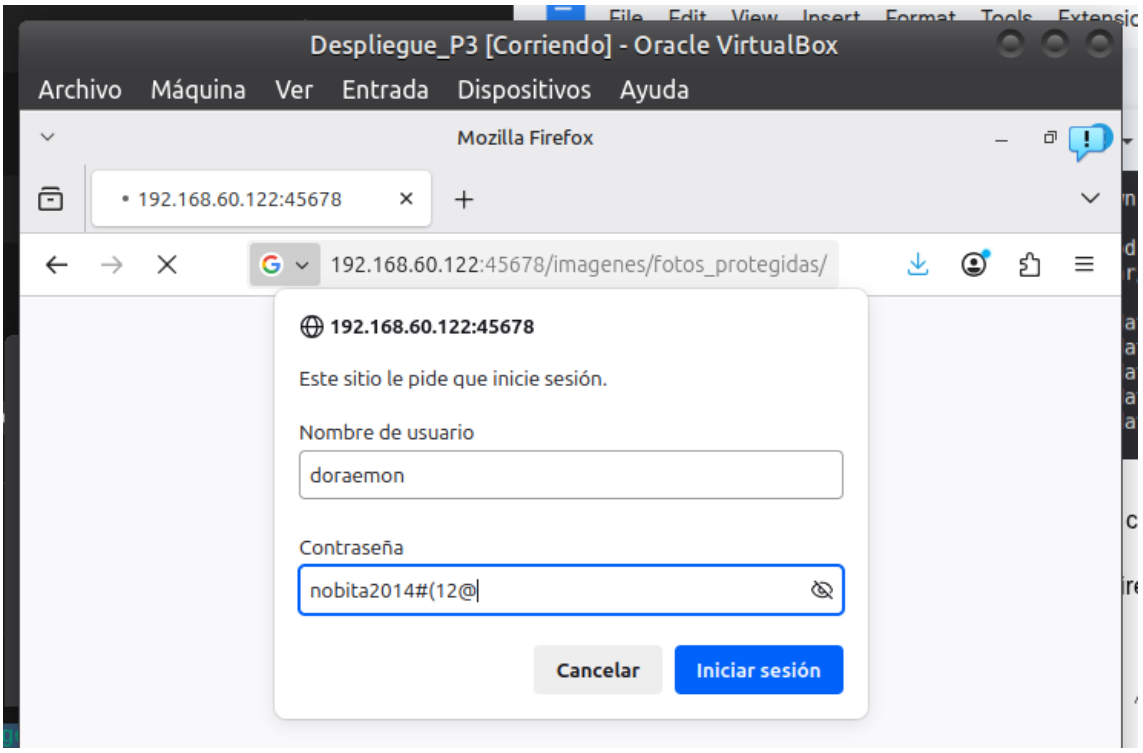


Verificación del funcionamiento

Acceder desde el navegador web a la siguiente dirección:

```
https://192.168.60.122:45678/imagenes/fotos_protegidas/
```

El navegador solicita usuario y contraseña.



Tras introducir las credenciales correctas, se muestra el contenido del directorio con las cinco imágenes.

Justificación técnica y de seguridad

La autenticación básica permite restringir el acceso a recursos sensibles del servidor web, evitando accesos no autorizados.

Al implementarse sobre HTTPS, las credenciales se transmiten cifradas, lo que incrementa la seguridad del sistema.

Esta medida cumple los criterios de administración segura exigidos en la práctica y demuestra el control de acceso a nivel de servidor sin utilizar ficheros .htaccess.

PRÁCTICA 11. Redirección temporal del servidor nº 2 al servidor seguro

Objetivo

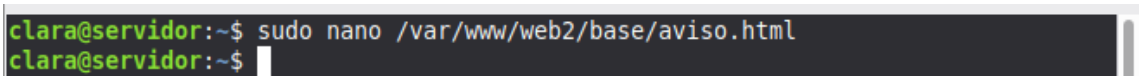
Al acceder al servidor nº 2, redireccionar automáticamente al servidor seguro, mostrando durante 5 segundos un aviso previo de redirección.

Configuración realizada

Paso 1: Creación de la página de aviso (aviso.html)

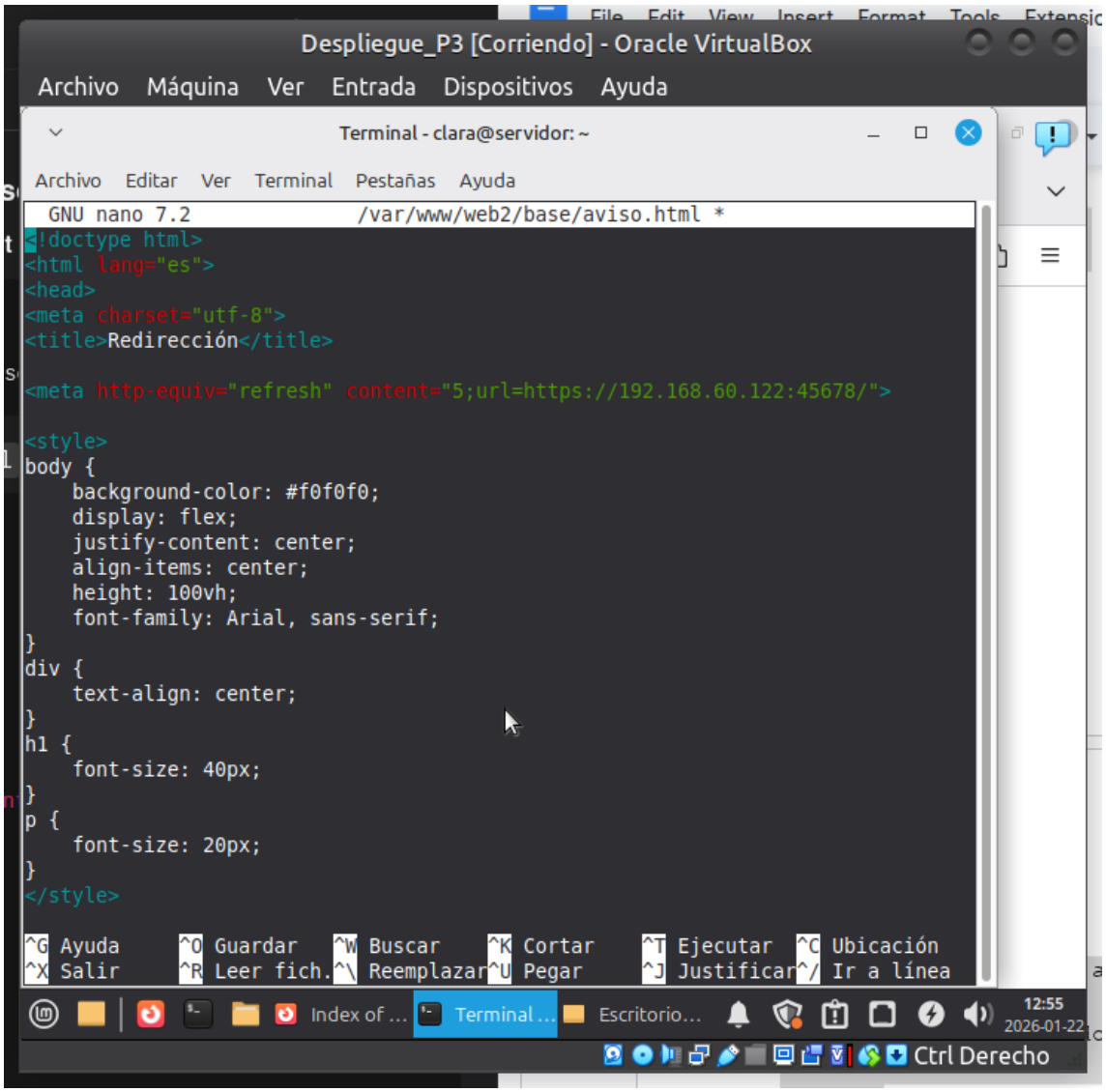
Crear el fichero aviso.html en el DocumentRoot del servidor web nº 2:

```
sudo nano /var/www/web2/base/aviso.html
```



Contenido del fichero aviso.html:

```
<!doctype html>
<html lang="es">
<head>
<meta charset="utf-8">
<title>Redirección</title>

<meta http-equiv="refresh"
content="5;url=https://192.168.60.122:45678/">

<style>
body {
    background-color: #f0f0f0;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
    font-family: Arial, sans-serif;
}
div {
    text-align: center;
}
h1 {
    font-size: 40px;
}
p {
    font-size: 20px;
}
</style>
</head>

<body>
<div>
    <h1>Redirigiendo al servidor seguro</h1>
    <p>En unos segundos será redirigido automáticamente…</p>
</div>
</body>
</html>
```



Esta página informa al usuario de la redirección y realiza el cambio automáticamente tras 5 segundos.

Paso 2: Edición del VirtualHost del servidor nº 2

Editar el fichero de configuración del servidor web nº 2:

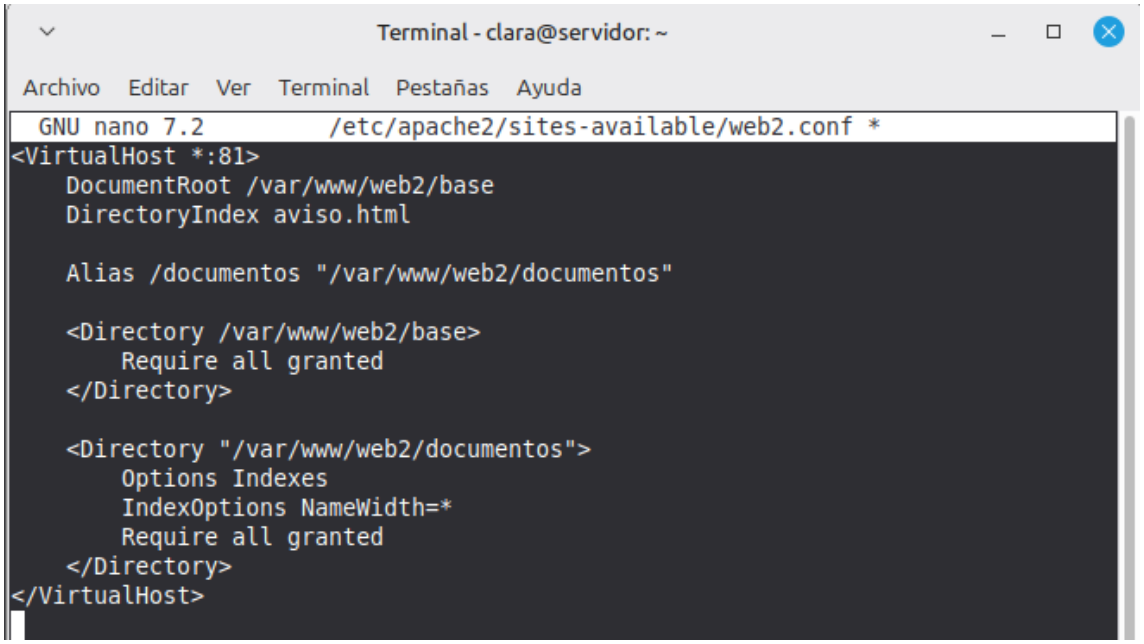
```
sudo nano /etc/apache2/sites-available/web2.conf
```

Dentro del bloque `<VirtualHost *:81>`, añadir la siguiente directiva **sin eliminar ninguna configuración existente**:

```
DirectoryIndex aviso.html
```

El bloque debe quedar configurado de la siguiente forma:

```
<VirtualHost *:81>
	DocumentRoot /var/www/web2/base
	DirectoryIndex aviso.html

	Alias /documentos "/var/www/web2/documentos"

	<Directory /var/www/web2/base>
		Require all granted
	</Directory>

	<Directory "/var/www/web2/documentos">
		Options Indexes
		IndexOptions NameWidth=*
		Require all granted
	</Directory>
</VirtualHost>
```



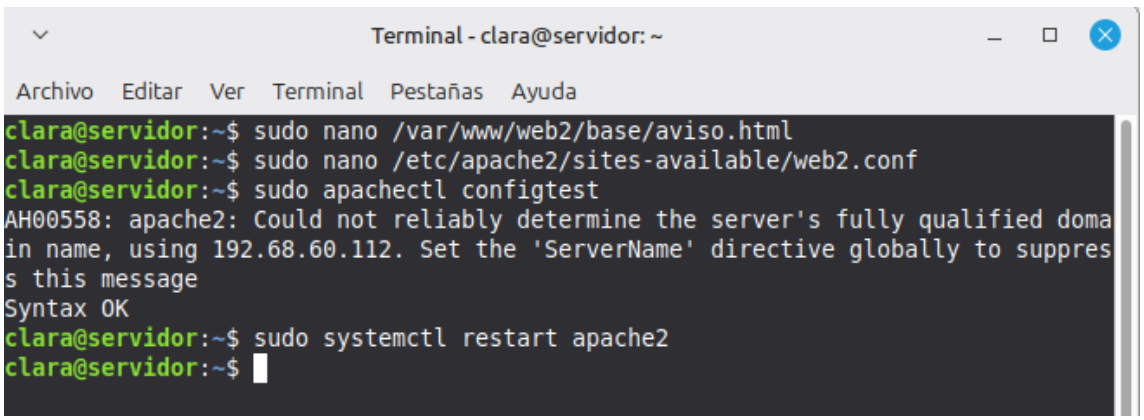
Con esta configuración, al acceder a la raíz del servidor web nº 2 se muestra la página de aviso creada.

Paso 3: Aplicación de los cambios

Comprobar la sintaxis de la configuración y reiniciar el servicio Apache:

```
sudo apachectl configtest
```

```
sudo systemctl restart apache2
```



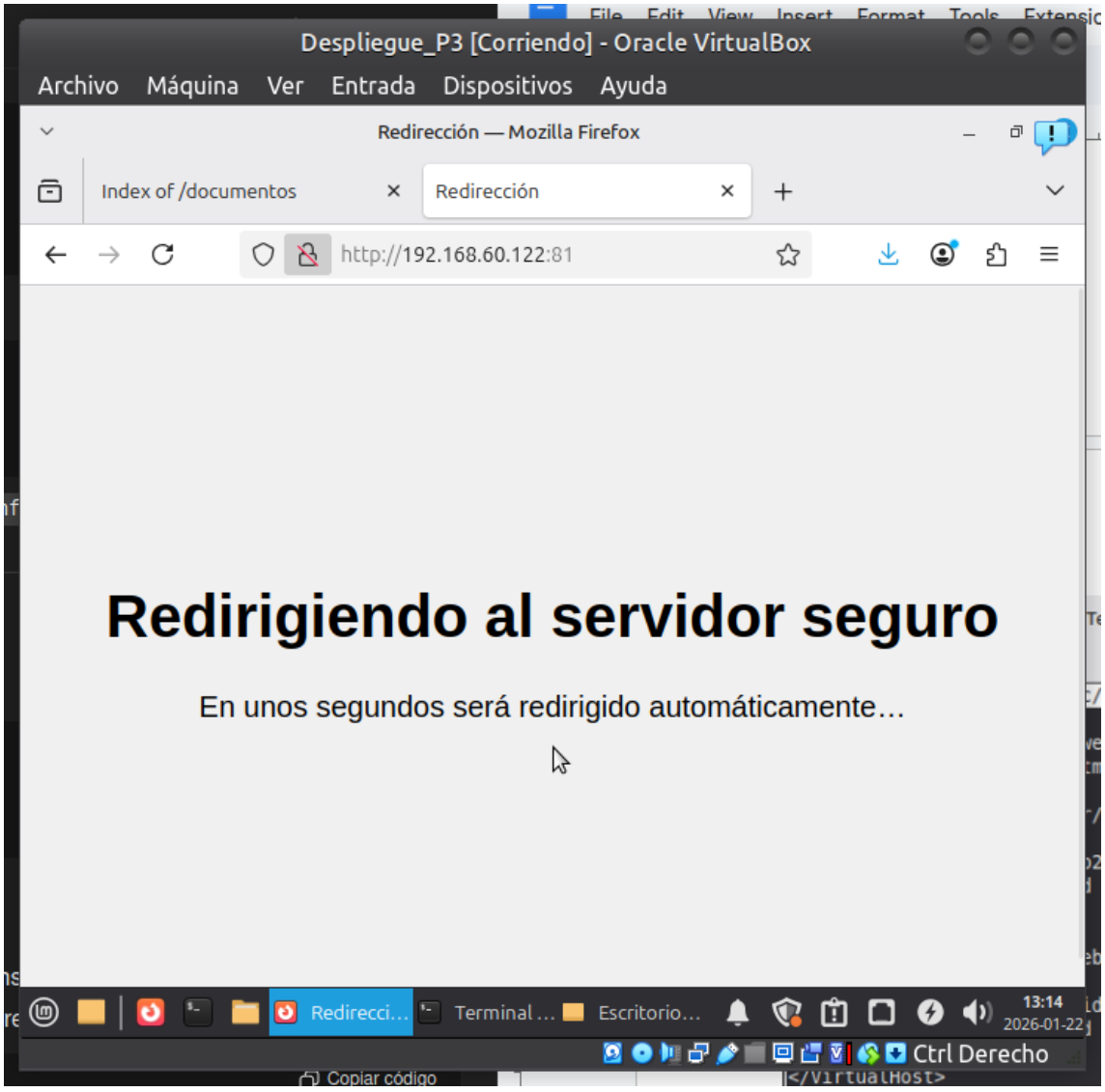
Verificación del funcionamiento

Acceder desde el navegador web al servidor nº 2:

`http://192.168.60.122:81`

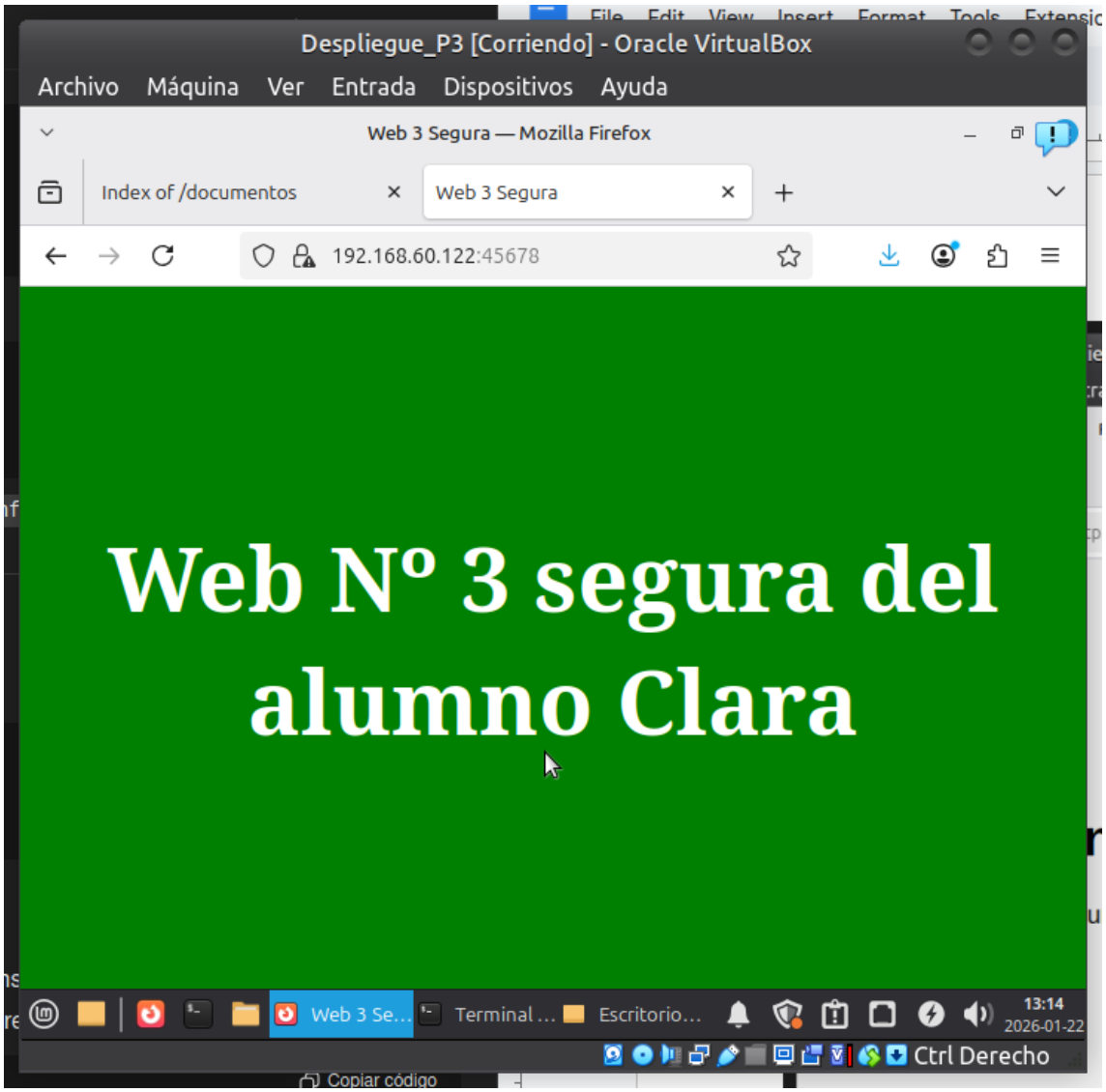
Resultado esperado:

Se muestra la página de aviso.



Tras 5 segundos, el navegador redirige automáticamente a:

`https://192.168.60.122:45678/`



Justificación técnica y de seguridad

Esta configuración informa al usuario de la redirección antes de realizarla y fuerza el acceso al servidor web seguro mediante HTTPS, mejorando la seguridad de las comunicaciones.

El uso de una página intermedia evita redirecciones abruptas, mejora la experiencia de usuario y permite notificar claramente el cambio de contexto, cumpliendo los criterios de administración segura y usabilidad establecidos en la práctica.

PRÁCTICA 12. Ejecución automática de un fichero PHP con extensión personalizada en el servidor nº 1

Objetivo

Configurar el servidor nº 1 para que al iniciarse en su directorio inicial ejecute automáticamente el fichero inicio.hjk, que contendrá código PHP, incluyendo la función phpinfo()

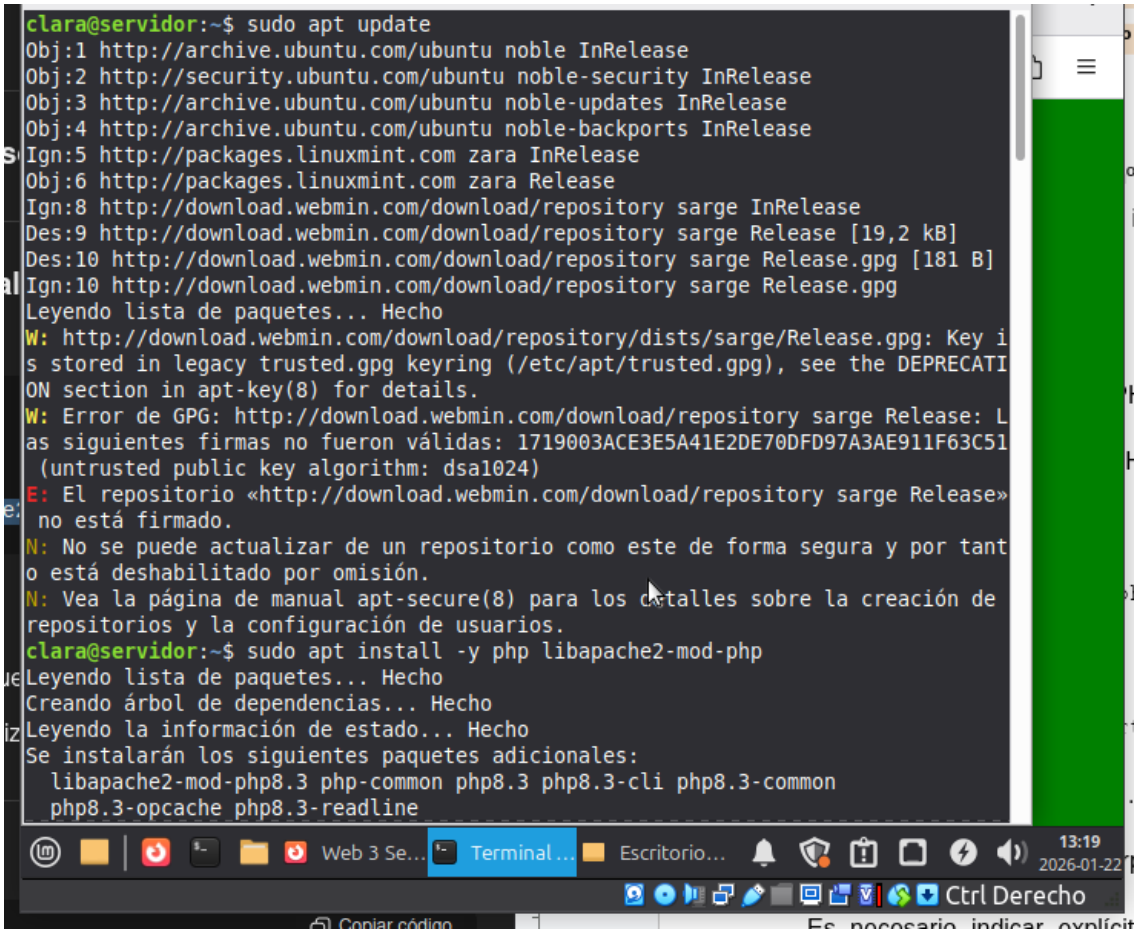
Configuración realizada

Paso 1: Instalación y habilitación del soporte PHP en Apache

Para que Apache pueda interpretar código PHP, se instala el soporte necesario mediante los siguientes comandos:

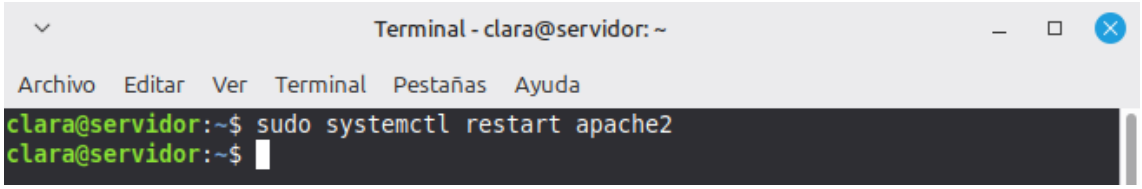
```
sudo apt update
```

```
sudo apt install -y php libapache2-mod-php
```



Una vez finalizada la instalación, se reinicia el servicio Apache:

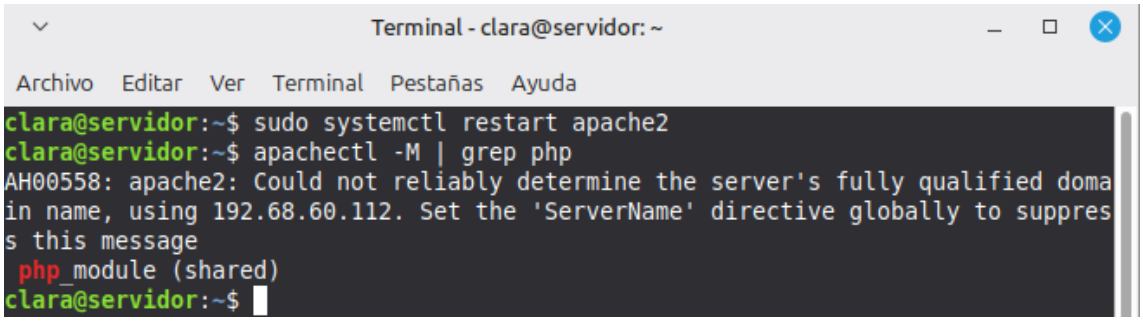
```
sudo systemctl restart apache2
```



Paso 2: Comprobación de que PHP está cargado en Apache

Se verifica que el módulo PHP se encuentra cargado correctamente en Apache mediante el comando:

```
apachectl -M | grep php
```



La aparición del módulo confirma que Apache puede ejecutar código PHP.

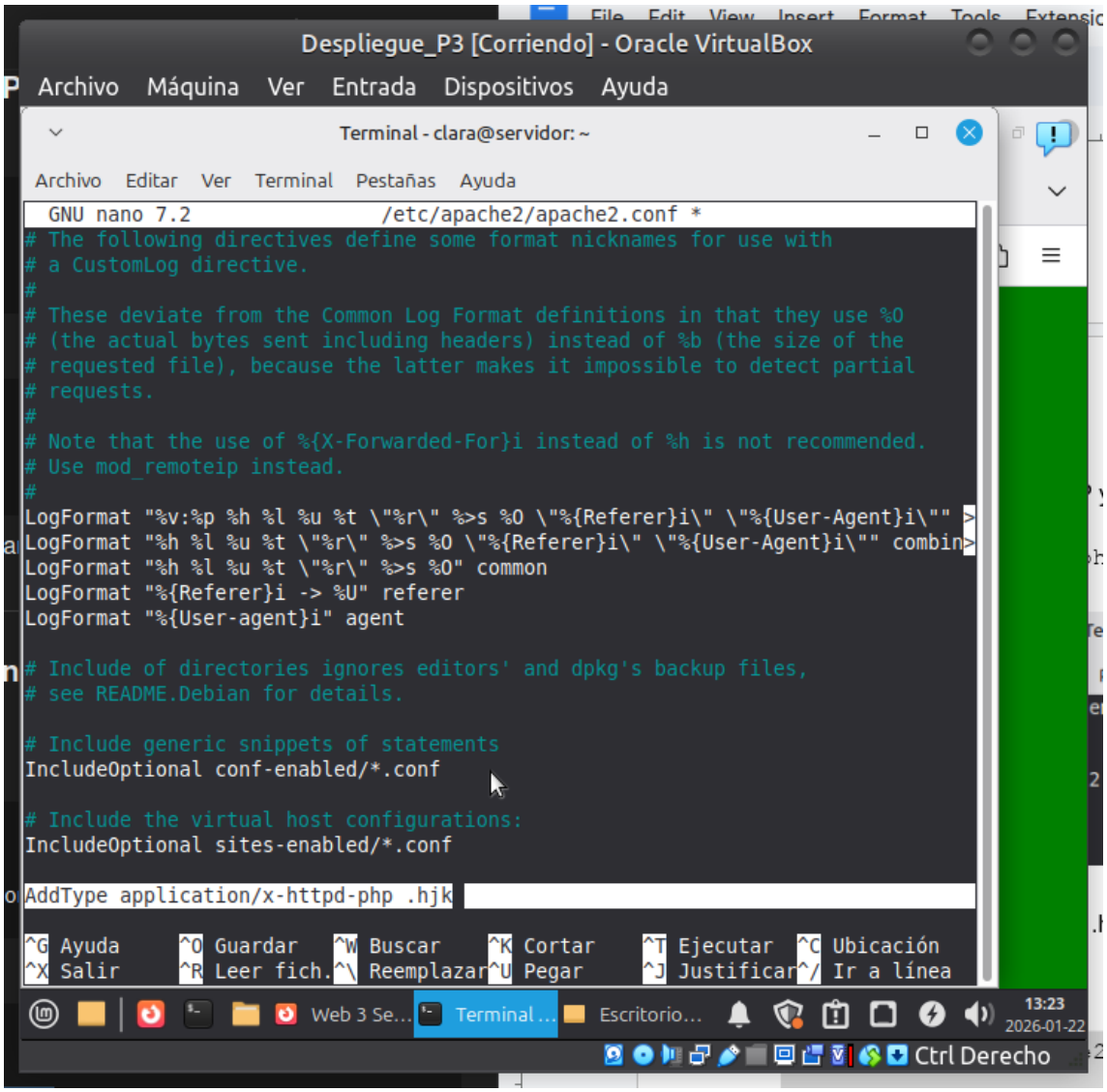
Paso 3: Asociación de la extensión .hjk con PHP

Editar el fichero de configuración global de Apache:

```
sudo nano /etc/apache2/apache2.conf
```

Añadir al final del fichero la siguiente directiva:

```
AddType application/x-httpd-php .hjk
```



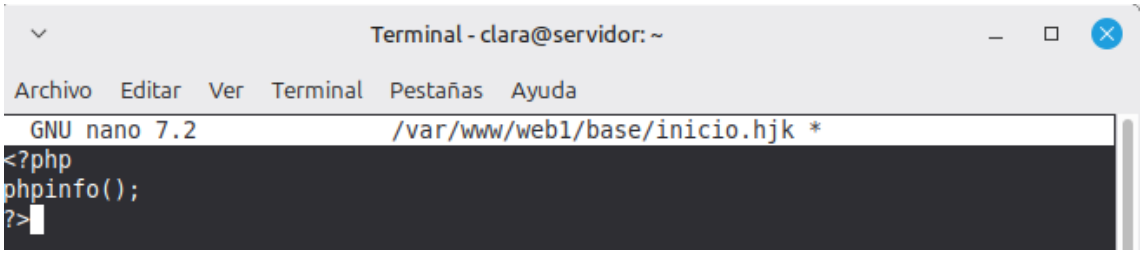
Con esta configuración se indica a Apache que los ficheros con extensión .hjk deben interpretarse como código PHP.

Paso 4: Creación del fichero PHP inicio.hjk

Crear el fichero que se ejecutará automáticamente al acceder al servidor web nº 1:

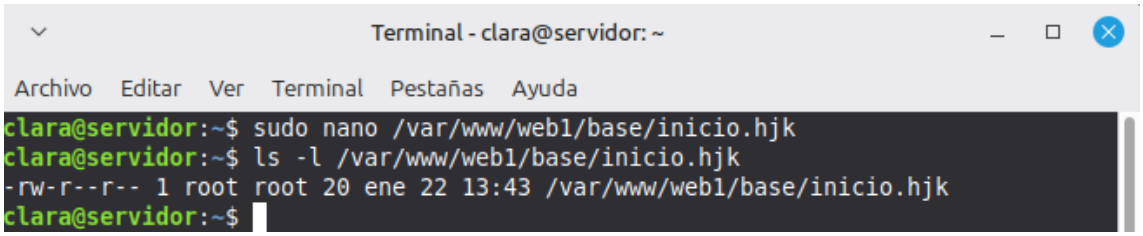
```
sudo nano /var/www/web1/base/inicio.hjk
```

```
<?php
phpinfo();
?>
```



Comprobar que el fichero tiene los permisos correctos:

```
ls -l /var/www/web1/base/inicio.hjk
```



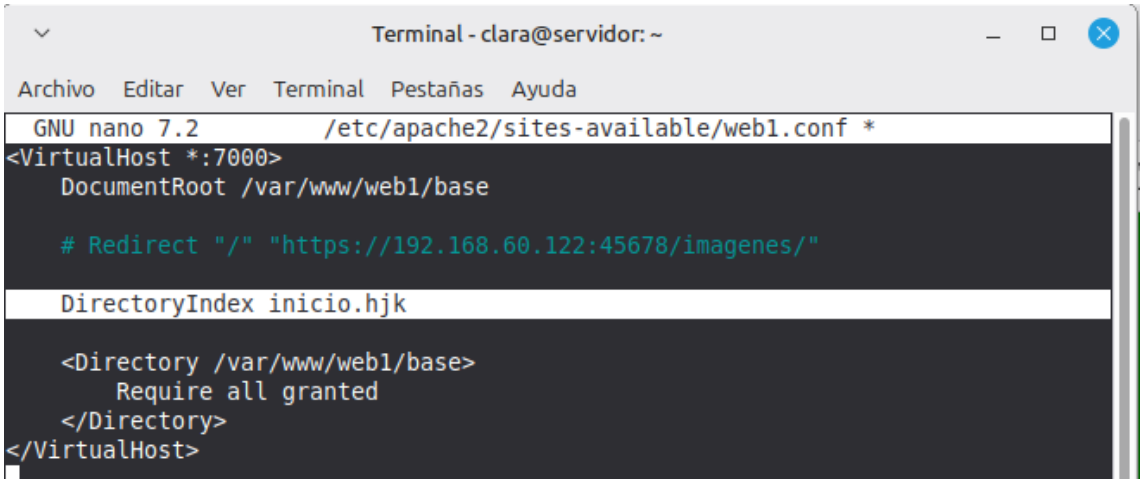
Paso 5: Configuración del fichero índice en el servidor nº 1

Editar el VirtualHost del servidor web nº 1:

```
sudo nano /etc/apache2/sites-available/web1.conf
```

Dentro del bloque <VirtualHost>, asegurarse de que existe la siguiente directiva:

```
DirectoryIndex inicio.hjk
```



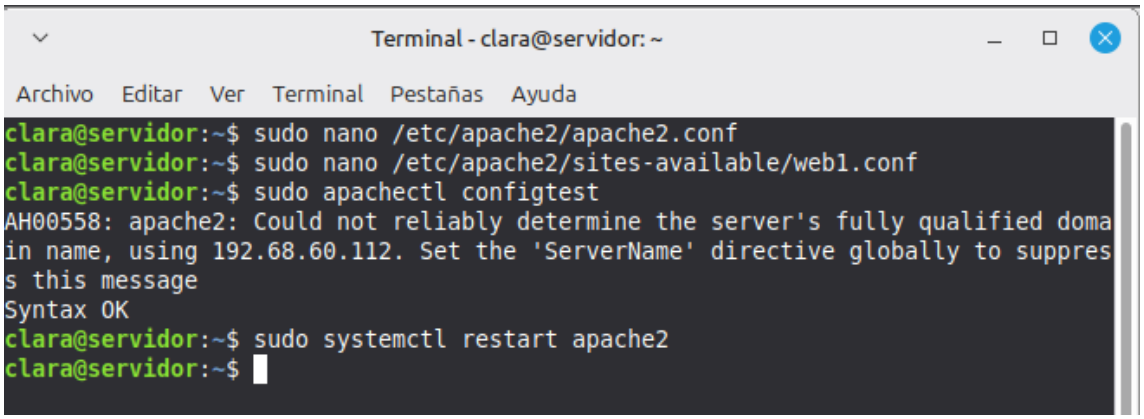
Para que esta práctica funcione correctamente, se debe comentar o eliminar cualquier directiva de redirección configurada previamente, ya que impediría la ejecución automática del fichero inicio.hjk.

Paso 6: Comprobación y reinicio del servicio Apache

Verificar que la configuración no contiene errores y reiniciar Apache:

```
sudo apachectl configtest
```

```
sudo systemctl restart apache2
```



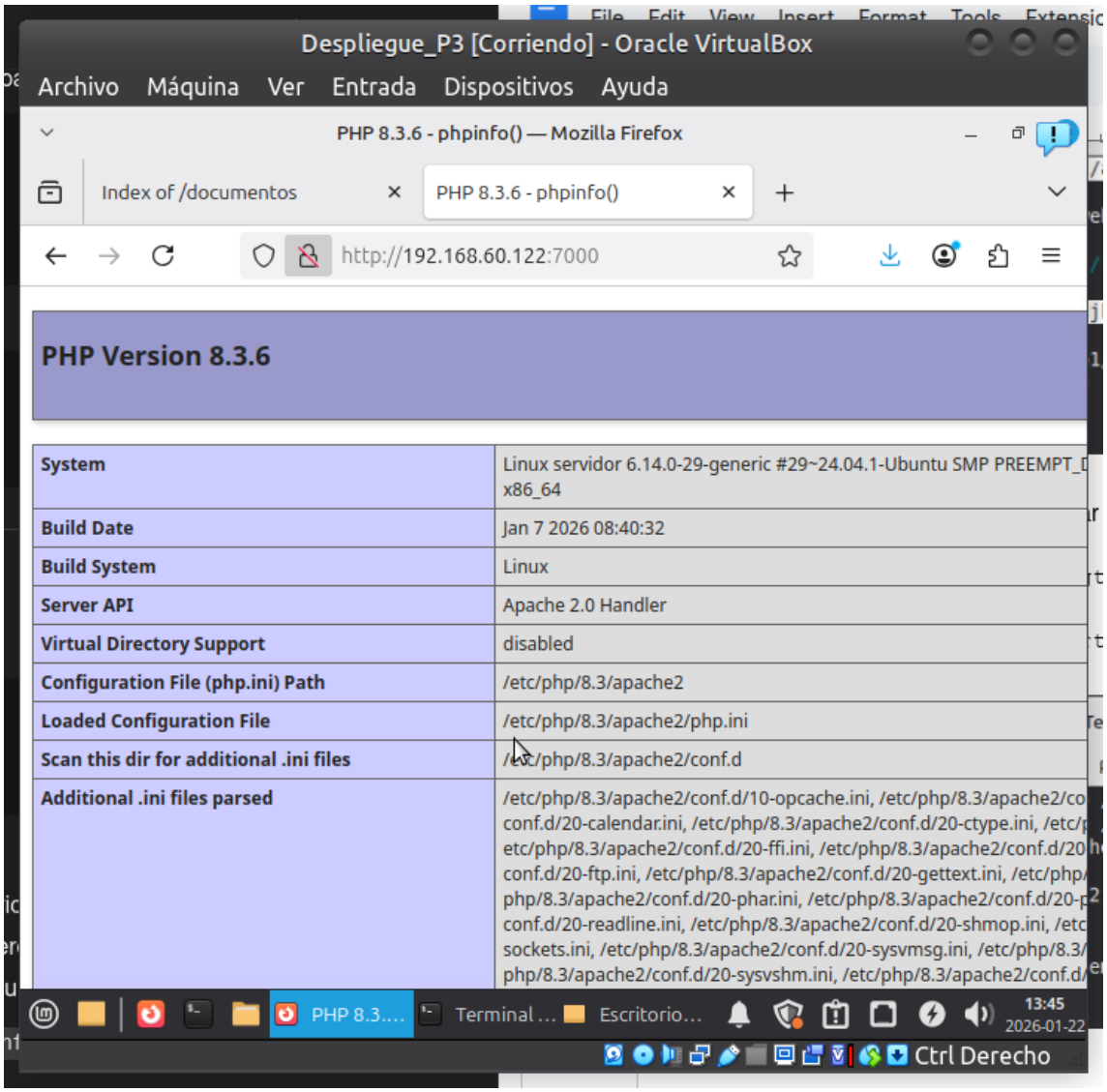
Verificación del funcionamiento

Acceder desde el navegador web al servidor nº 1:

```
http://192.168.60.122:7000
```

Resultado esperado:

- El fichero inicio.hjk se ejecuta automáticamente.
- Se muestra la página generada por la función phpinfo(), confirmando que PHP se interpreta correctamente.



Justificación técnica y de seguridad

La configuración de un índice automático mediante la directiva DirectoryIndex permite controlar qué fichero se ejecuta al acceder al directorio raíz del servidor, evitando la exposición del listado de archivos.

El uso de una extensión personalizada asociada explícitamente a PHP demuestra un control avanzado del servidor Apache y permite ejecutar código dinámico de forma segura y controlada, cumpliendo los criterios de administración avanzada exigidos en la práctica.

PRÁCTICA 13. Registro de los navegadores que acceden al servidor web

Objetivo

Adaptar Apache para que queden registros de los navegadores que visitan la web.

Configuración realizada

Paso 1: Definición de un formato de log personalizado

Apache permite definir formatos de registro personalizados mediante la directiva LogFormat.

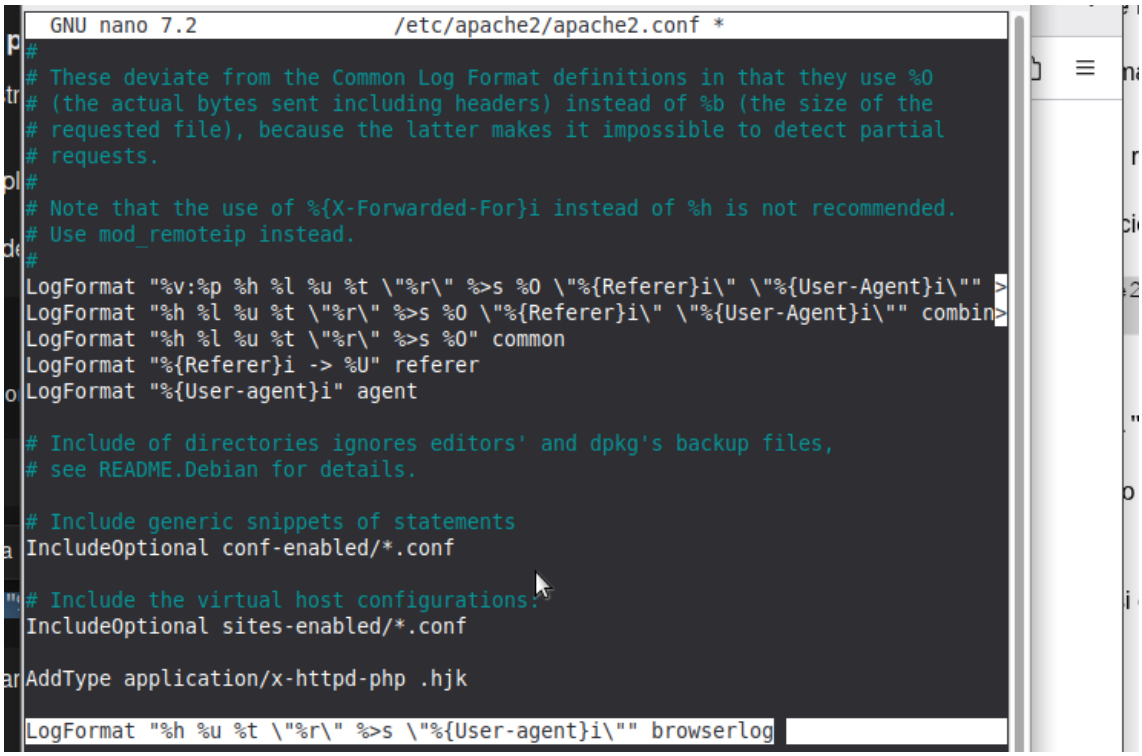
En esta práctica se crea un formato específico que registre explícitamente el User-Agent del navegador.

Editar el fichero de configuración global de Apache:

```
sudo nano /etc/apache2/apache2.conf
```

Añadir al final del fichero la siguiente directiva:

```
LogFormat "%h %u %t \"%r\" %>s \"%{User-agent}i\"" browserlog
```



Este formato de log, denominado browserlog, registra la siguiente información:

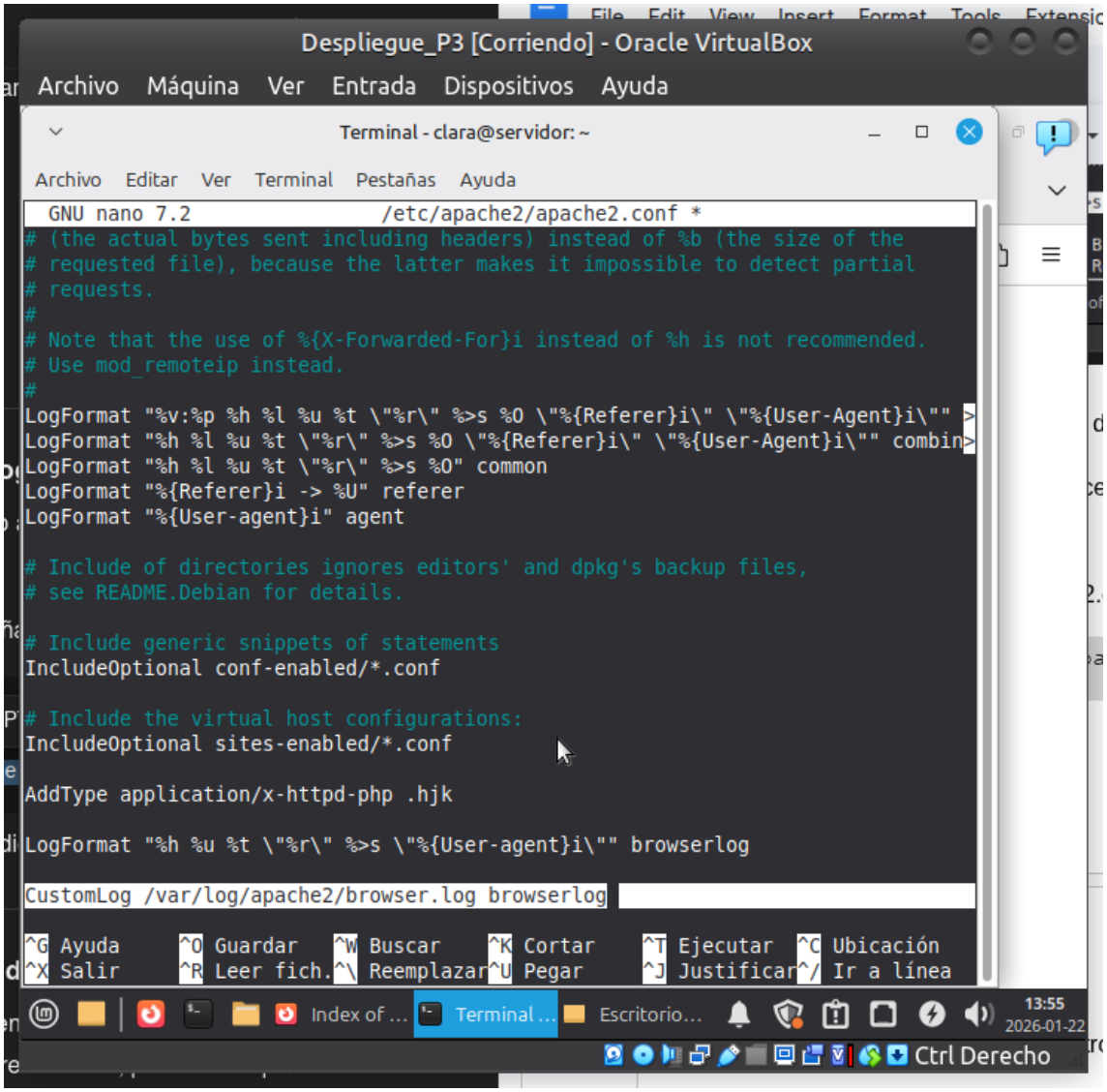
- Dirección IP del cliente.
- Usuario autenticado (si existe).
- Fecha y hora del acceso.
- Petición HTTP realizada.
- Código de estado devuelto por el servidor.
- Navegador utilizado (User-Agent).

Paso 2: Configuración del fichero de registro de navegadores

Indicar a Apache que utilice el formato definido anteriormente y que guarde la información en un fichero de log independiente.

En el mismo fichero (apache2.conf), añadir la siguiente directiva:

```
CustomLog /var/log/apache2/browser.log browserlog
```



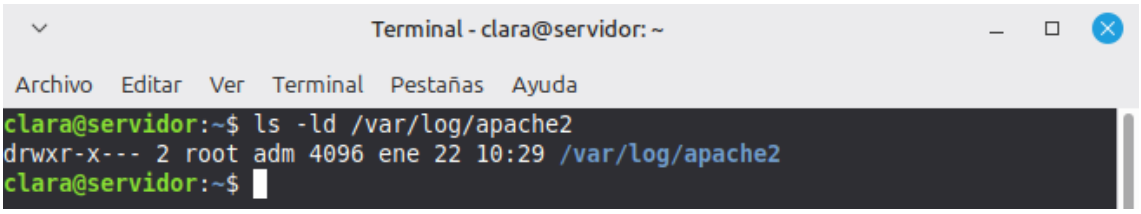
Con esta configuración se crea un fichero de registro específico para almacenar los accesos clasificados por navegador.

Paso 3: Comprobación de los permisos del directorio de logs

Apache necesita permisos de escritura en el directorio de logs para poder registrar la información correctamente.

Comprobar los permisos del directorio:

```
ls -ld /var/log/apache2
```



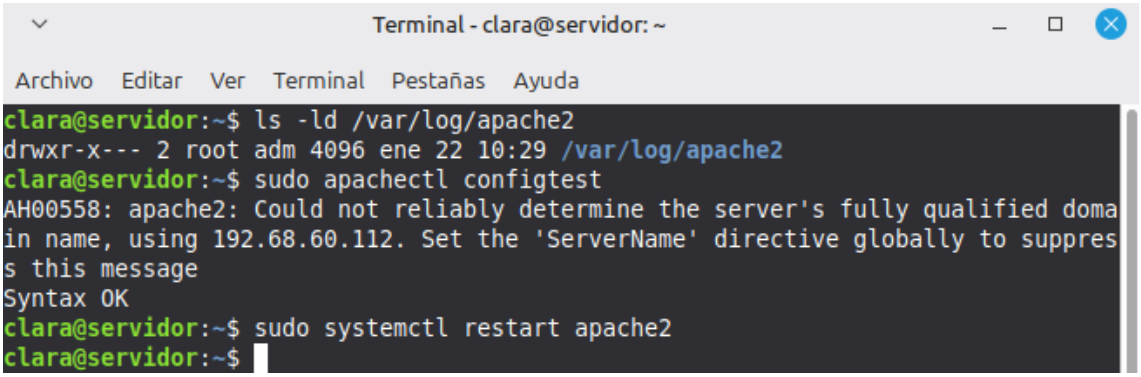
No es necesario modificar los permisos si el directorio pertenece al sistema y Apache ya puede escribir en él.

Paso 4: Aplicación de la configuración

Comprobar la sintaxis de la configuración y reiniciar el servicio Apache para aplicar los cambios:

```
sudo apachectl configtest
```

```
sudo systemctl restart apache2
```



Verificación del funcionamiento

Paso 1: Generación de accesos desde el navegador

Acceder desde el navegador web a cualquiera de los servidores configurados, por ejemplo:

```
http://192.168.60.122:7000
http://192.168.60.122:81
https://192.168.60.122:45678
```

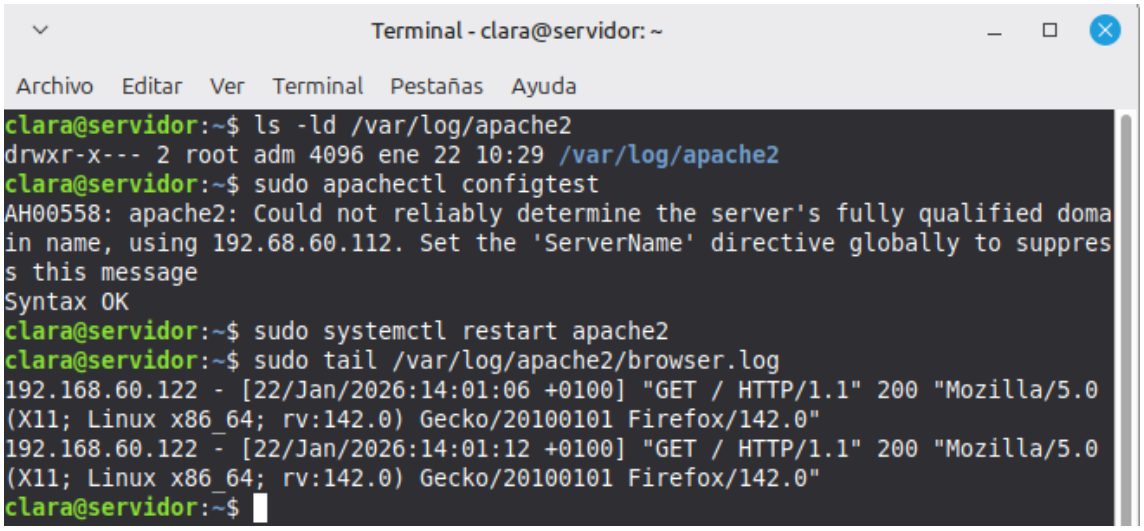
Paso 2: Verificación del fichero de log

Desde la terminal, comprobar el contenido del fichero de registro:

```
sudo tail /var/log/apache2/browser.log
```

Resultado esperado:

- Aparecen nuevas líneas por cada acceso realizado.
- Se identifica claramente el navegador utilizado (Firefox, Chrome, etc.).



Justificación técnica y de seguridad

El registro del navegador utilizado por los clientes permite al administrador del servidor analizar el tipo de navegadores que acceden a la web, detectar accesos desde navegadores obsoletos o potencialmente inseguros y obtener información útil para auditorías y análisis de tráfico.

La separación de este registro en un fichero específico mejora la organización de los logs y facilita su posterior procesamiento automático, cumpliendo criterios de administración avanzada y segura del servidor web.

PRÁCTICA 14. Conteo de visitas por navegador a partir de los registros

Objetivo

Crear un script en Linux o un programa en PHP que cuente cuántas visitas se han producido por cada navegador, leyendo los ficheros de log.

Configuración realizada

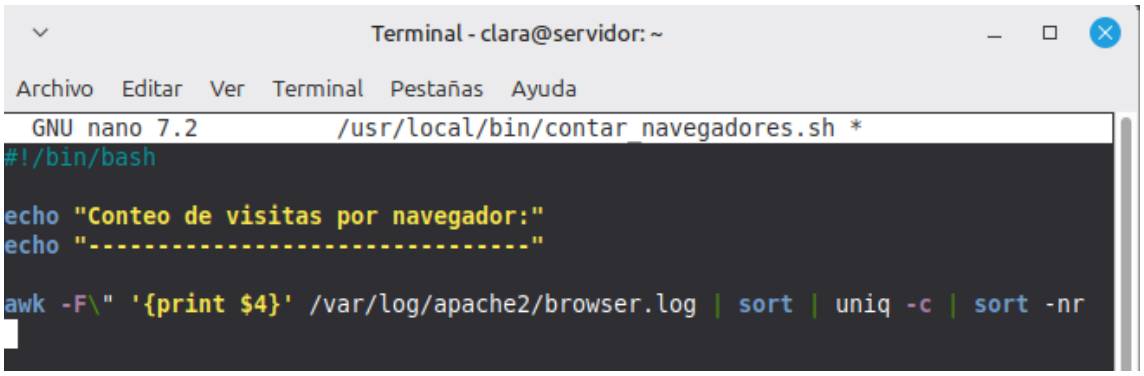
Paso 1: Creación del script de conteo de navegadores

Se crea un script en una ruta estándar del sistema para su ejecución directa como comando:

```
sudo nano /usr/local/bin/contar_navegadores.sh
```

Contenido del script:

```
#!/bin/bash
echo "Conteo de visitas por navegador:"
echo "--------------------------------"
awk -F\" '{print $4}' /var/log/apache2/browser.log | sort | uniq -c |
sort -nr
```



Este script realiza las siguientes acciones:

- Lee el fichero de log /var/log/apache2/browser.log.
- Extrae el campo correspondiente al User-Agent del navegador.
- Ordena los resultados.
- Cuenta cuántas veces aparece cada navegador.
- Muestra el resultado ordenado de mayor a menor número de accesos.

Paso 2: Asignación de permisos de ejecución

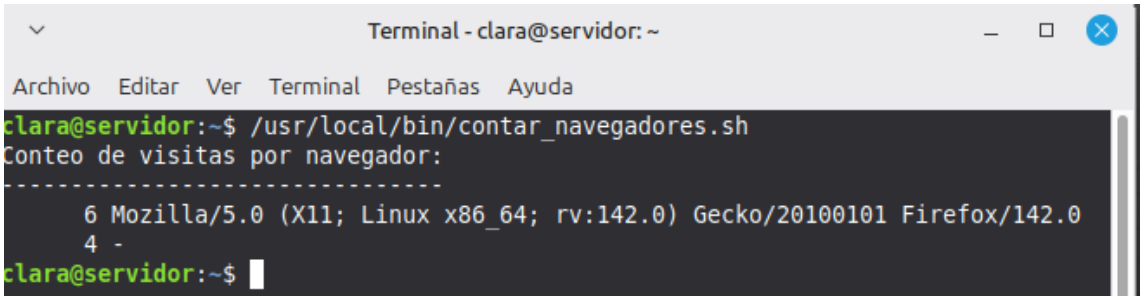
Dar permisos de ejecución al script para poder ejecutarlo como un comando del sistema:

```
sudo chmod +x /usr/local/bin/contar_navegadores.sh
```

Verificación del funcionamiento

Ejecutar el script desde la terminal:

```
/usr/local/bin/contar_navegadores.sh
```



Tras ejecutar el script de conteo de navegadores, se obtiene el siguiente resultado:

- 6 accesos realizados desde el navegador Mozilla Firefox sobre sistema Linux.
- 4 accesos sin identificación de navegador, correspondientes a peticiones que no incluyen la cabecera User-Agent.

Justificación técnica y de seguridad

El análisis automático de los registros del servidor permite conocer los navegadores más utilizados por los usuarios, detectar accesos anómalos o automatizados y facilitar tareas de auditoría y monitorización del servidor web.

El uso de un script en Linux basado en herramientas estándar como awk, sort y uniq proporciona una solución eficiente, reutilizable y fácilmente ampliable, cumpliendo los criterios de administración avanzada y gestión segura de servidores web exigidos en la práctica.

PRÁCTICA 15. Protección completa del servidor web nº 2 mediante autenticación

Objetivo

Proteger todos los directorios del servidor nº 2, sin utilizar .htaccess, con: 2

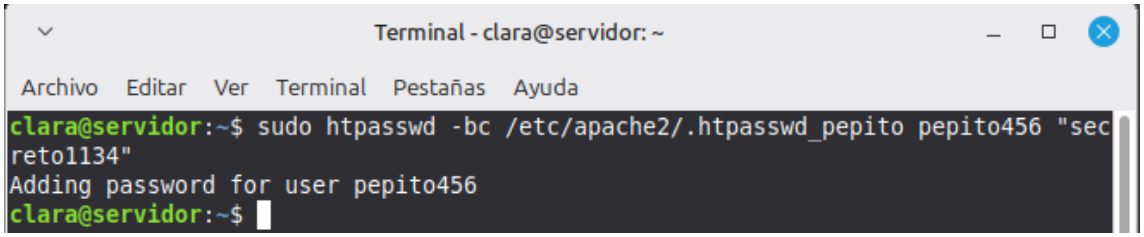
- Usuario: pepito456
- Contraseña: secreto1134

Configuración realizada

Paso 1: Creación del usuario y la contraseña de autenticación

Se crea un fichero de autenticación global y el usuario solicitado mediante el siguiente comando:

```
sudo htpasswd -bc /etc/apache2/.htpasswd_pepito pepito456 "secreto1134"
```



Este fichero almacenará de forma segura las credenciales necesarias para acceder al servidor web nº 2.

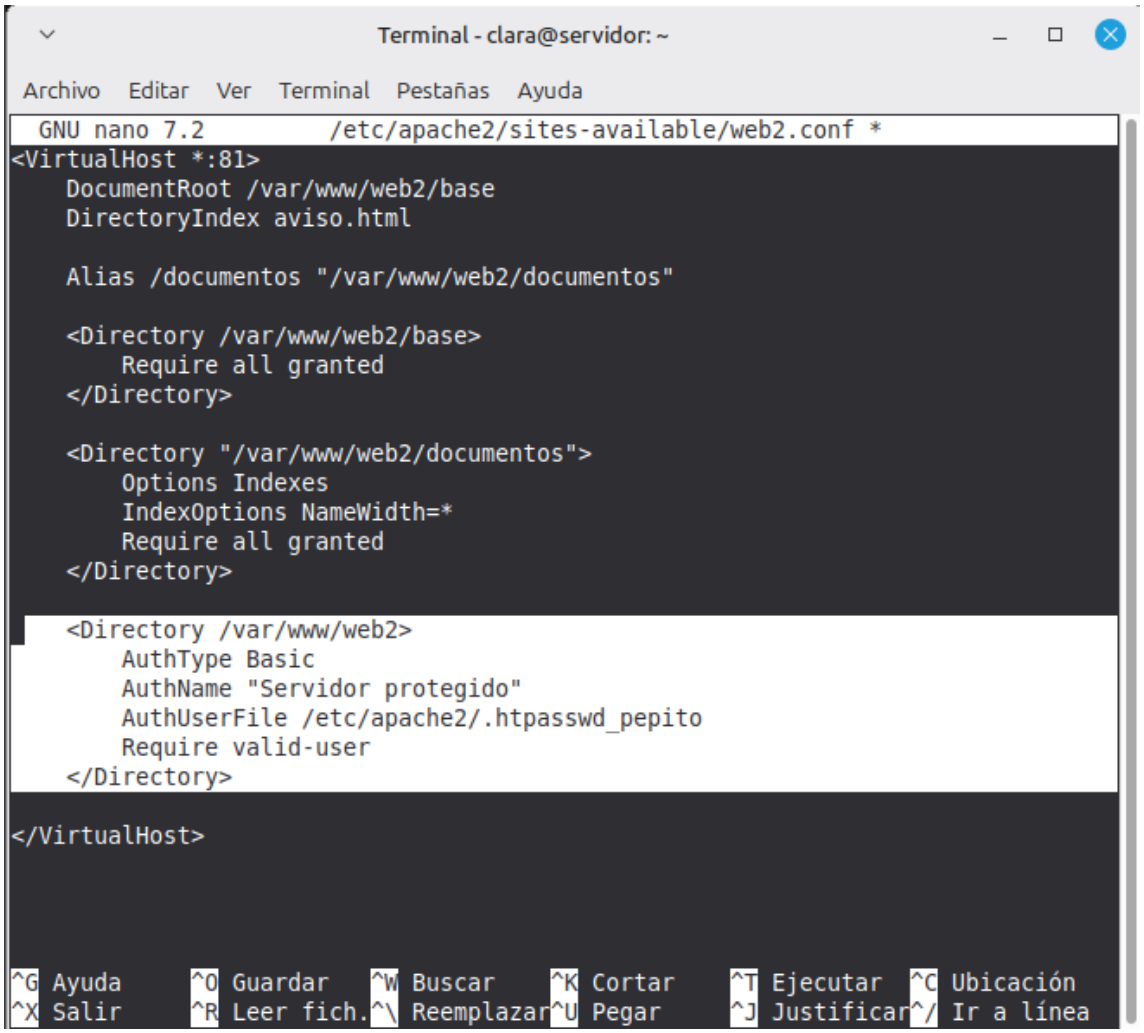
Paso 2: Edición del VirtualHost del servidor web nº 2

Editar el fichero de configuración del servidor web nº 2:

```
sudo nano /etc/apache2/sites-available/web2.conf
```

Dentro del bloque `<VirtualHost *:81>`, **sin eliminar ninguna configuración existente**, añadir el siguiente bloque para proteger todo el contenido del servidor:

```
<Directory /var/www/web2>
    AuthType Basic
    AuthName "Servidor protegido"
    AuthUserFile /etc/apache2/.htpasswd_pepito
    Require valid-user
</Directory>
```



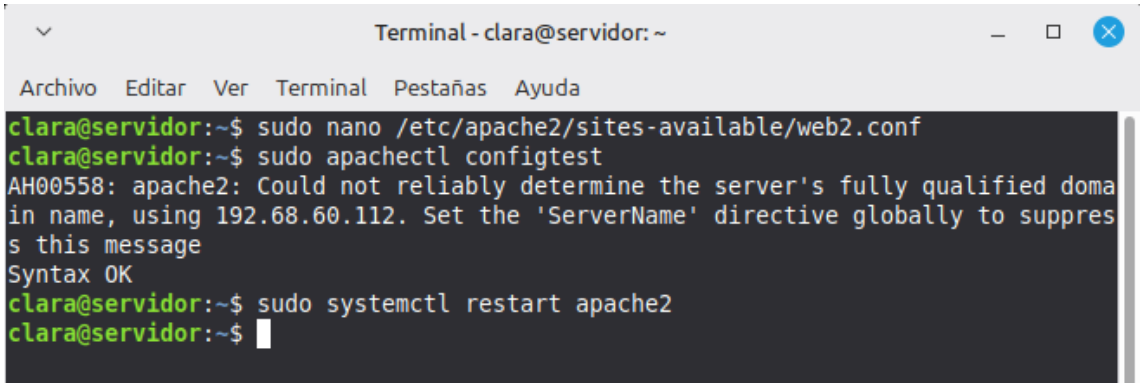
Con esta configuración se fuerza la autenticación para acceder a cualquier recurso alojado en el servidor web nº 2.

Paso 3: Comprobación de la sintaxis y aplicación de los cambios

Verificar que la configuración no contiene errores y reiniciar Apache:

```
sudo apachectl configtest
```

```
sudo systemctl restart apache2
```



Verificación del funcionamiento

Al acceder a la dirección:

`http://192.168.60.122:81/`

no aparece la ventana de autenticación y el navegador redirige directamente al servidor web nº 3.

Esto ocurre porque en la Práctica 11 se configuró una redirección automática desde el servidor web nº 2 al servidor seguro, lo que provoca que el navegador sea enviado fuera del servidor nº 2 antes de que Apache pueda solicitar las credenciales.

Por este motivo:

- No aparece la autenticación.
- Se produce la redirección automática al servidor nº 3.
- No es posible comprobar la protección del servidor nº 2 mientras la redirección esté activa.

La práctica 15 no puede funcionar mientras la práctica 11 esté activa.

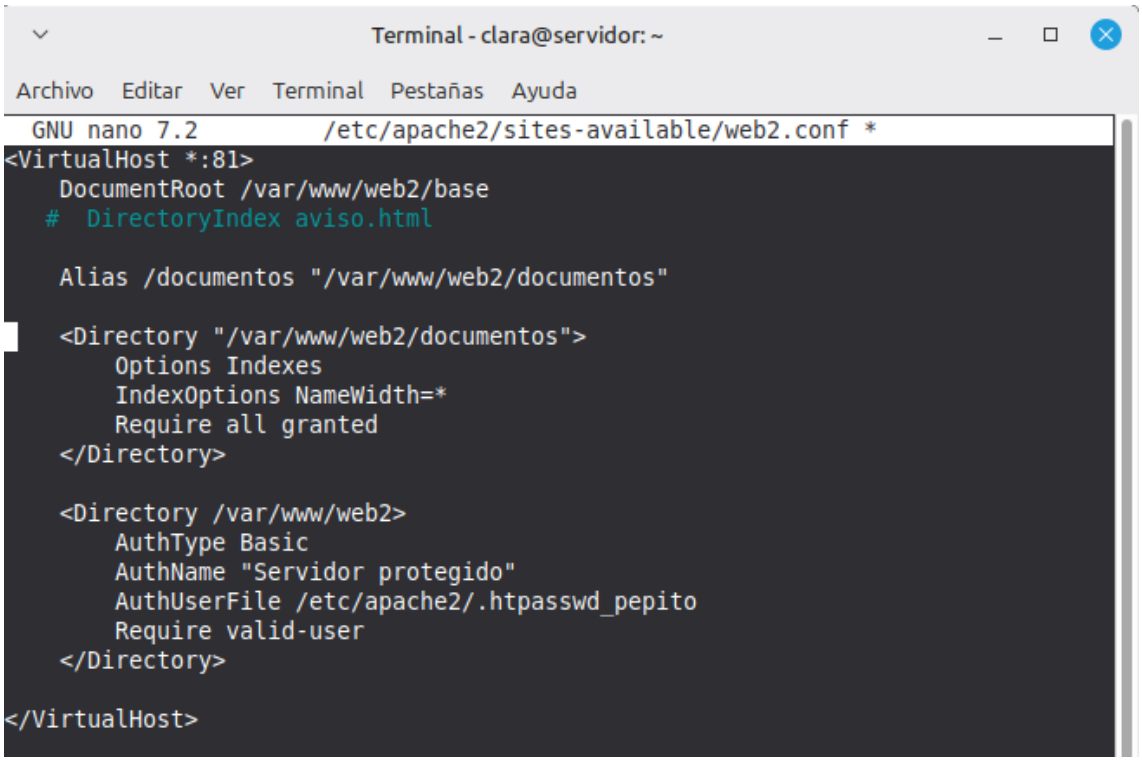
Para poder comprobar correctamente la autenticación del servidor web nº 2, es necesario desactivar temporalmente la redirección configurada en la práctica 11

Paso 4: Edición del VirtualHost del servidor nº 2

```
sudo nano /etc/apache2/sites-available/web2.conf
```

Paso 5: Comentado de la redirección (sin eliminarla)

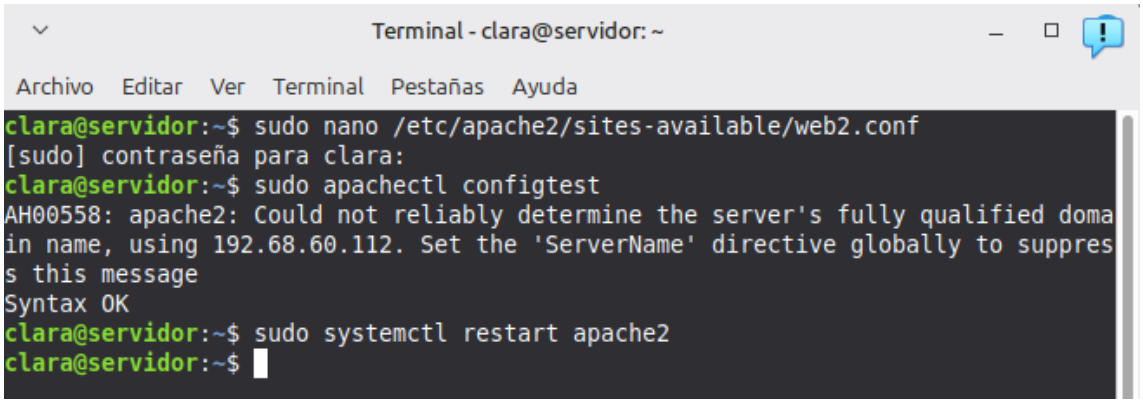
```
# DirectoryIndex aviso.html
```



Paso 6: Aplicación de los cambios

```
sudo apachectl configtest
```

```
sudo systemctl restart apache2
```



Verificación final

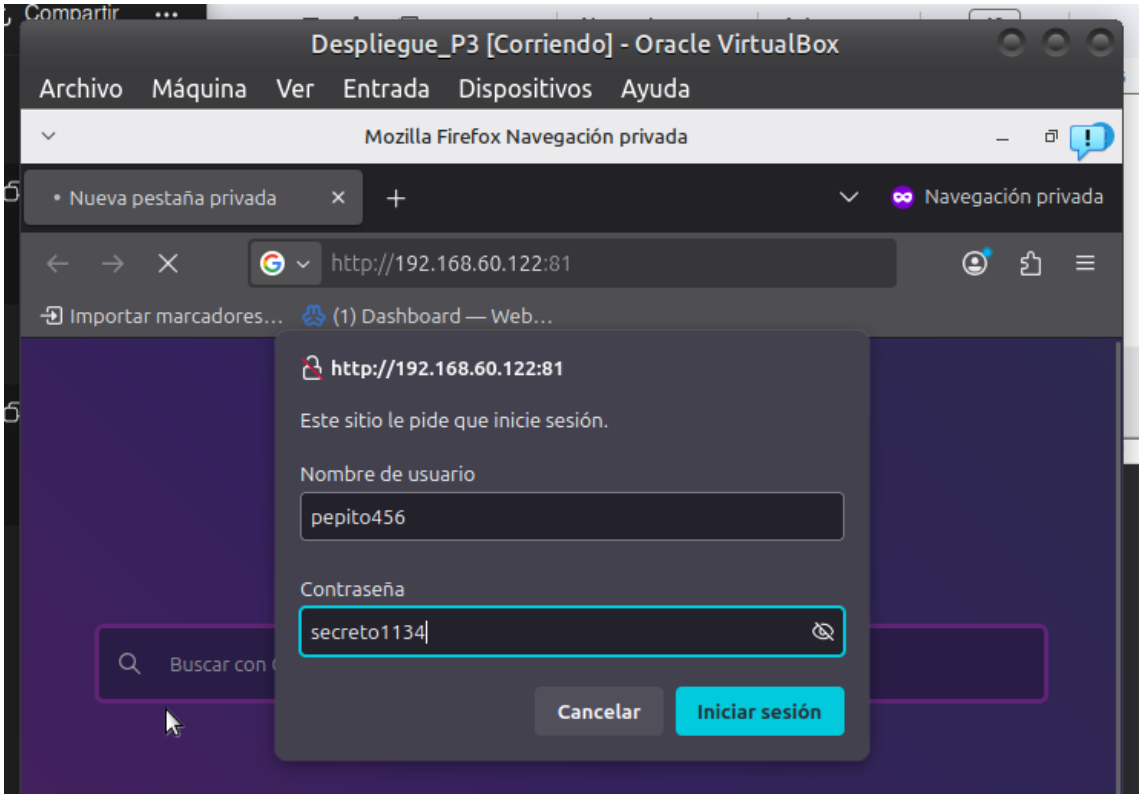
Al acceder a:

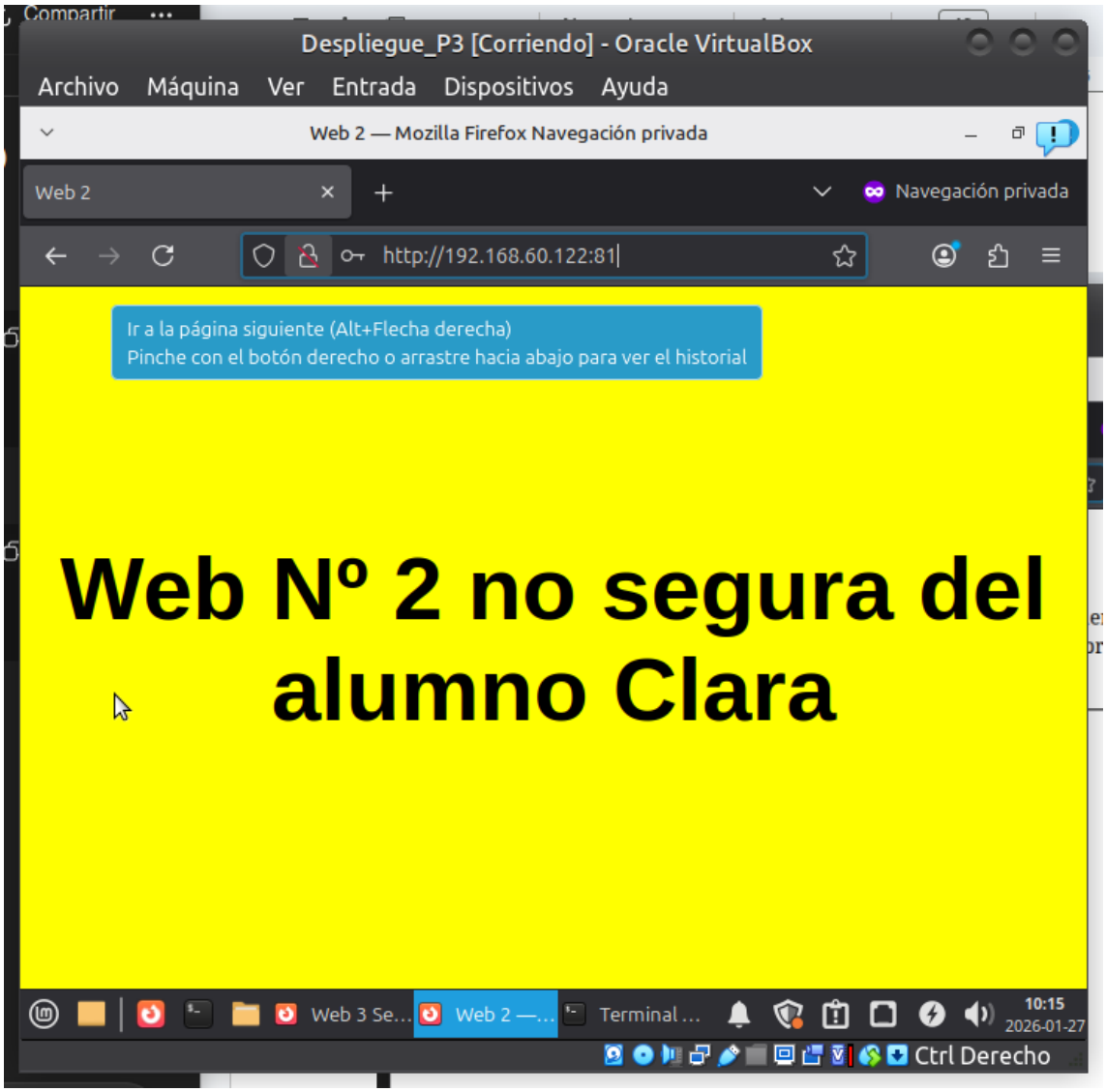
http://192.168.60.122:81/

Aparece la ventana de autenticación.

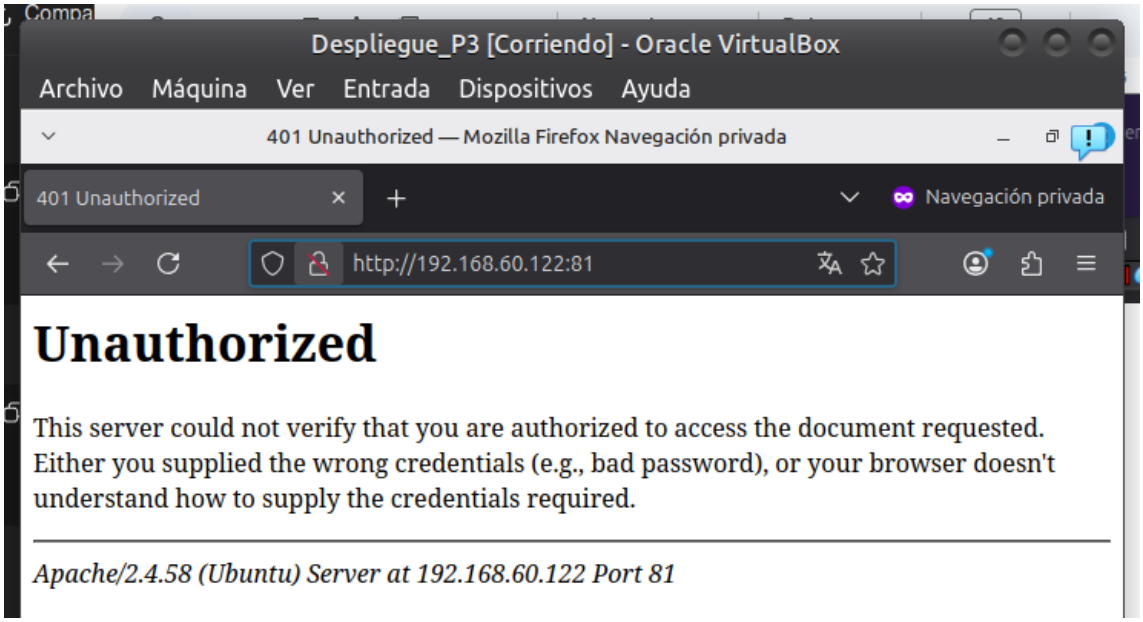
Introduciendo:

- Usuario: pepito456
- Contraseña: secreto1134
    - se permite el acceso al contenido del servidor.





- Sin credenciales válidas:
  - El acceso es denegado.



Justificación técnica y de seguridad

La autenticación básica restringe el acceso al servidor web nº 2 únicamente a usuarios autorizados, evitando accesos no deseados.

Al configurarse directamente en Apache y sin utilizar ficheros .htaccess, se mejora la seguridad mediante una administración centralizada, más controlada y coherente con los criterios de administración segura exigidos en la práctica.

PRÁCTICA 16. Página de cortesía por error 404 (Servidor nº 1)

Objetivo

Crear una página de cortesía que se muestre cuando el recurso solicitado no exista en el servidor nº 1, con el mensaje:

“Lo sentimos, señor usuario, pero la página solicitada no existe”,

y un fondo con la imagen de una flor.

Configuración realizada

Paso 1: Copia de la imagen al servidor web nº 1

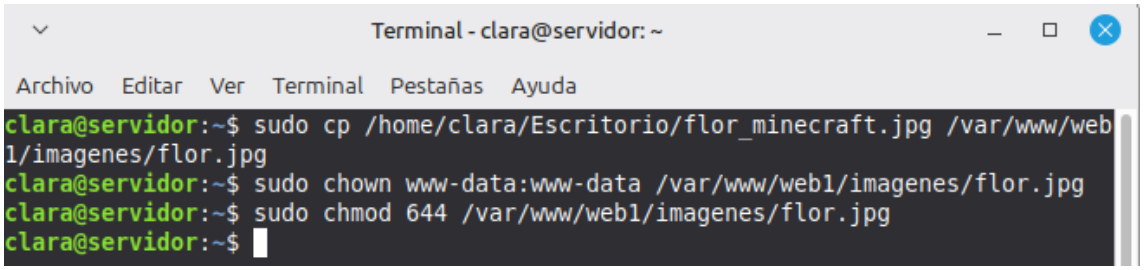
Copiar la imagen desde el Escritorio al directorio de imágenes del servidor web nº 1:

```
sudo cp /home/clara/Escritorio/flor_minecraft.jpg
/var/www/web1/imagenes/flor.jpg
```

Asignar el propietario y los permisos adecuados al fichero:

```
sudo chown www-data:www-data /var/www/web1/imagenes/flor.jpg
```

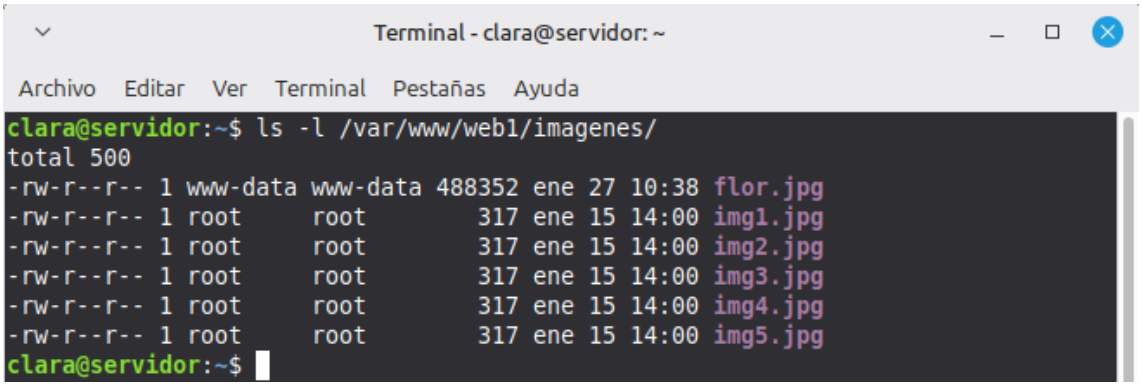
```
sudo chmod 644 /var/www/web1/imagenes/flor.jpg
```



Paso 2: Verificación del acceso a la imagen

Comprobar que Apache puede acceder correctamente a la imagen:

```
ls -l /var/www/web1/imagenes/
```

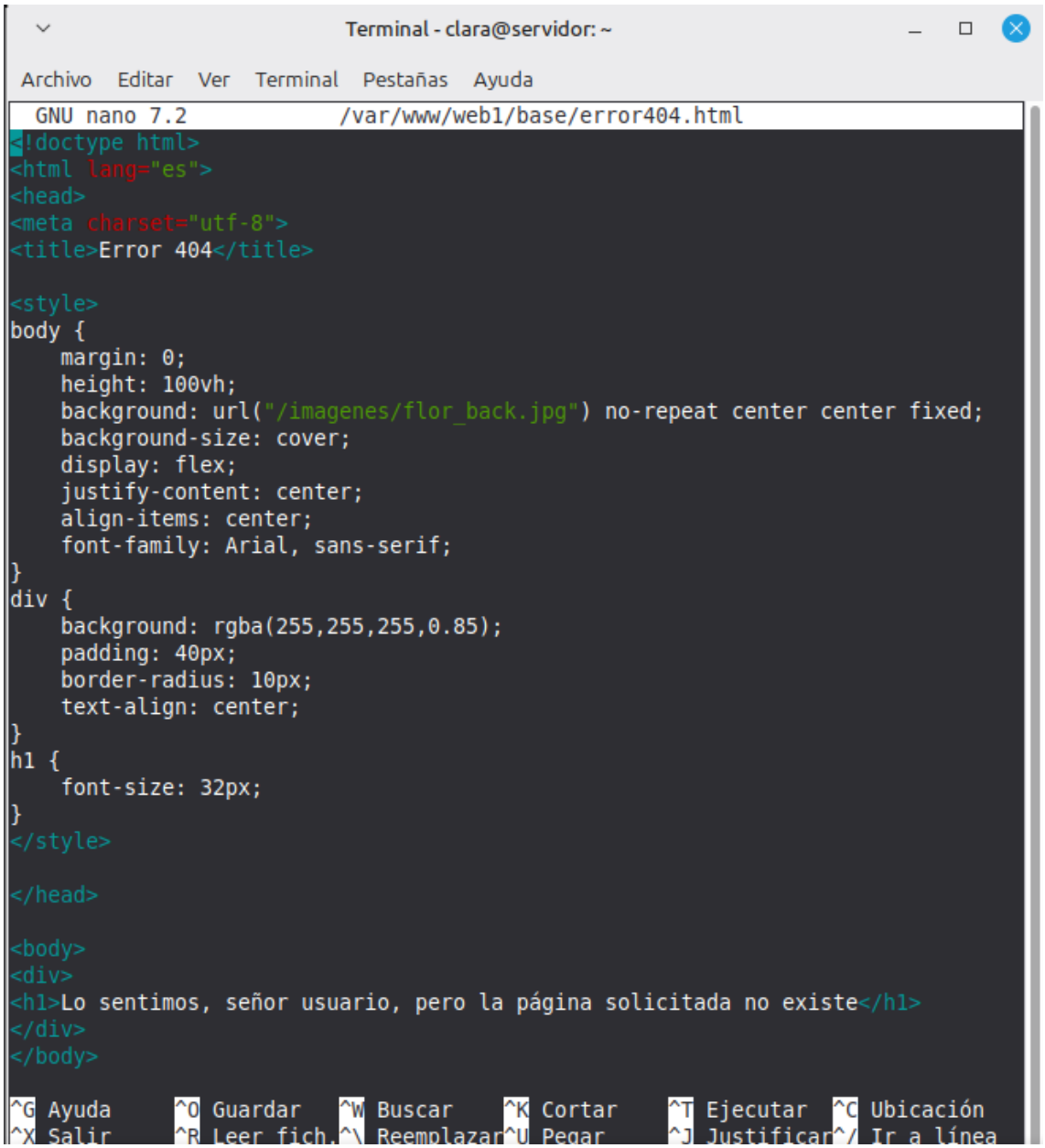


Paso 3: Creación de la página de error personalizada (error404.html)

Crear el fichero de la página de error 404 en el servidor web nº 1:

Contenido del fichero error404.html (ejemplo):

```
<!doctype html>
<html lang="es">
<head>
<meta charset="utf-8">
<title>Error 404</title>

<style>
body {
    background-image: url("/imagenes/flor.jpg");
    background-size: cover;
    background-position: center;
    font-family: Arial, sans-serif;
    color: white;
    text-align: center;
    height: 100vh;
    display: flex;
    justify-content: center;
    align-items: center;
}
div {
    background-color: rgba(0,0,0,0.6);
    padding: 30px;
    border-radius: 10px;
}
</style>
</head>

<body>
<div>
    <h1>Error 404</h1>
    <p>Lo sentimos, señor usuario, pero la página solicitada no existe</p>
</div>
</body>
</html>
```



Paso 4: Configuración de Apache para usar la página de error 404

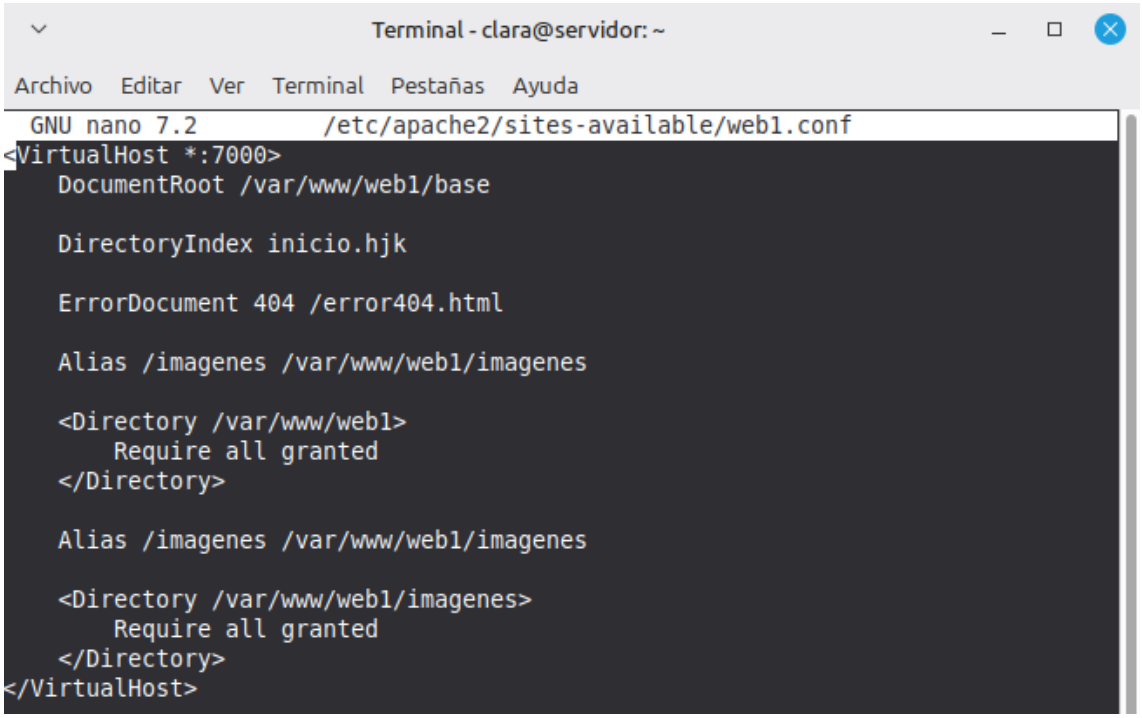
Editar el VirtualHost del servidor web nº 1:

```
sudo nano /etc/apache2/sites-available/web1.conf
```

Dentro del bloque `<VirtualHost *:7000>`, añadir la siguiente directiva **sin eliminar el resto de la configuración**:

```
ErrorDocument 404 /error404.html
```

Con esta directiva se indica a Apache que utilice la página personalizada cuando se solicite un recurso inexistente.

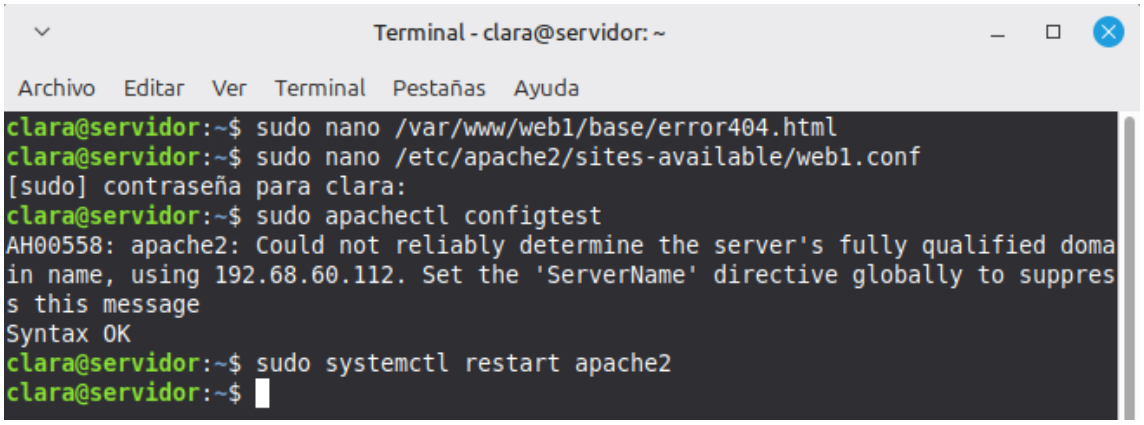


Paso 5: Aplicación de la configuración

Comprobar la sintaxis y reiniciar el servicio Apache:

```
sudo apachectl configtest
```

```
sudo systemctl restart apache2
```



Verificación del funcionamiento

Acceder desde el navegador a un recurso inexistente del servidor web nº 1, por ejemplo:

```
http://192.168.60.122:7000/esto_no_existe.html
```

el servidor muestra correctamente la página de cortesía configurada para el error 404.

En la página se visualiza el mensaje:

“Lo sentimos, señor usuario, pero la página solicitada no existe”

junto con el fondo definido mediante la imagen de una flor, confirmando que la directiva ErrorDocument 404 se ha aplicado correctamente y que Apache utiliza la página personalizada cuando el recurso solicitado no existe.

Justificación técnica y de seguridad

La creación de una página de error 404 personalizada mejora la experiencia del usuario al ofrecer un mensaje claro y controlado cuando se solicita un recurso inexistente.

Además, evita mostrar páginas de error genéricas de Apache que podrían revelar información interna del servidor, aumentando la seguridad y el control sobre el contenido mostrado.

PRÁCTICA 17. Redirección al servidor nº 3 cuando el recurso no exista

Objetivo

Redireccionar automáticamente a la raíz del servidor nº 3 cuando el recurso solicitado no exista.

Configuración realizada

Paso 1: Edición del VirtualHost del servidor nº 1

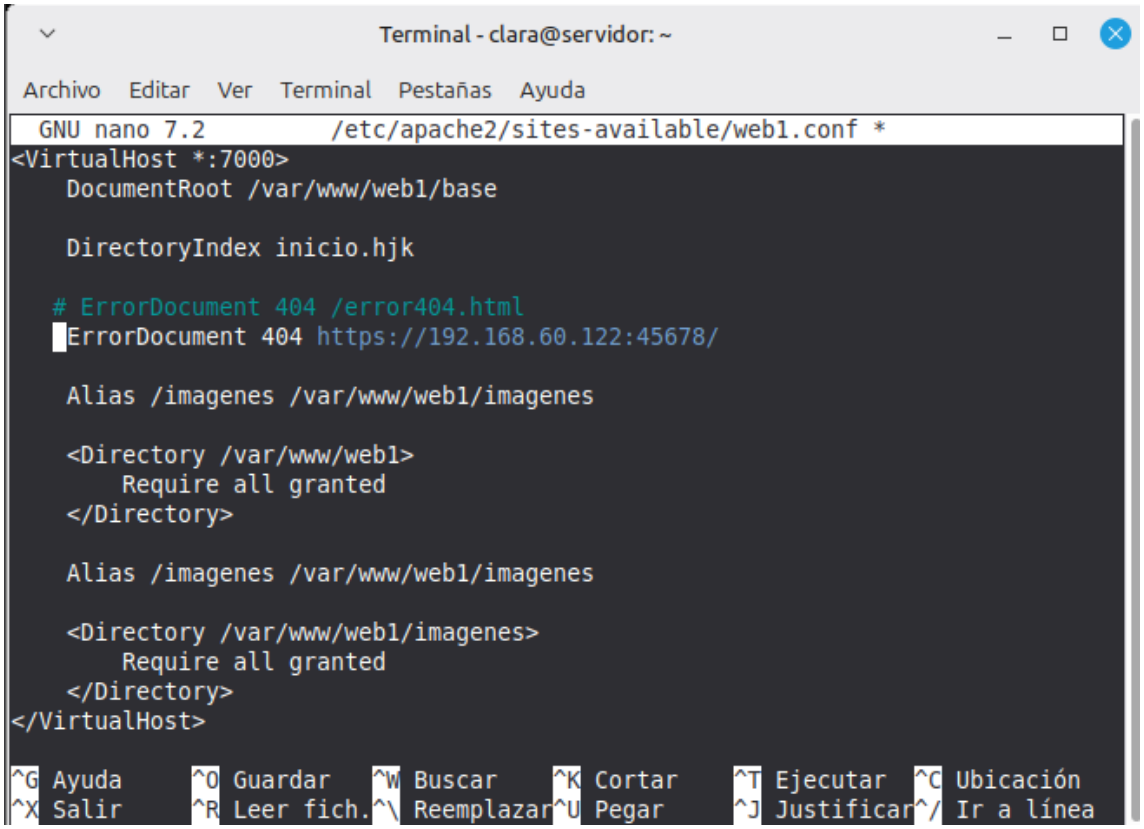
Abrir el fichero de configuración del servidor web nº 1:

```
sudo nano /etc/apache2/sites-available/web1.conf
```

Dentro del bloque `<VirtualHost *:7000>`, añadir o modificar la directiva ErrorDocument 404 para que quede configurada de la siguiente forma:

```
ErrorDocument 404 https://192.168.60.122:45678/
```

Con esta directiva se indica a Apache que, cuando se produzca un error 404 (recurso no encontrado) en el servidor nº 1, se redirija automáticamente al servidor web nº 3.

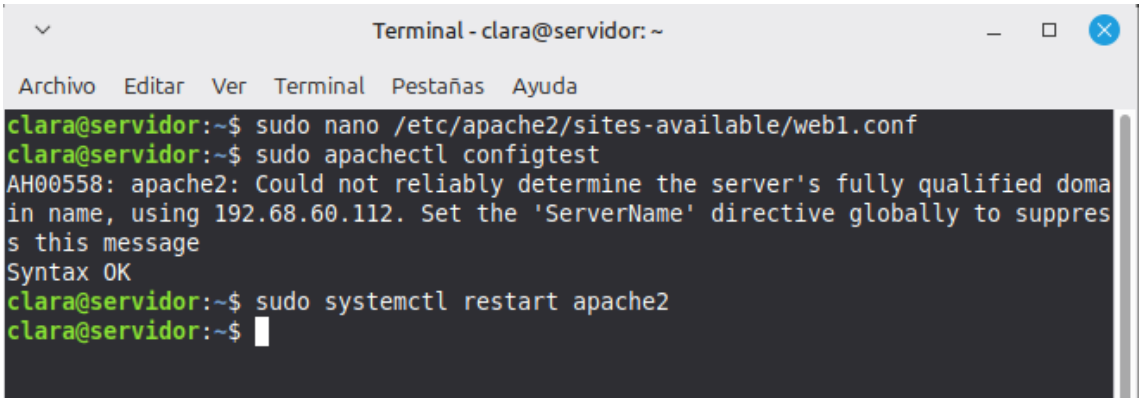


Paso 2: Aplicación de los cambios

Comprobar la sintaxis de la configuración y reiniciar el servicio Apache:

```
sudo apachectl configtest
```

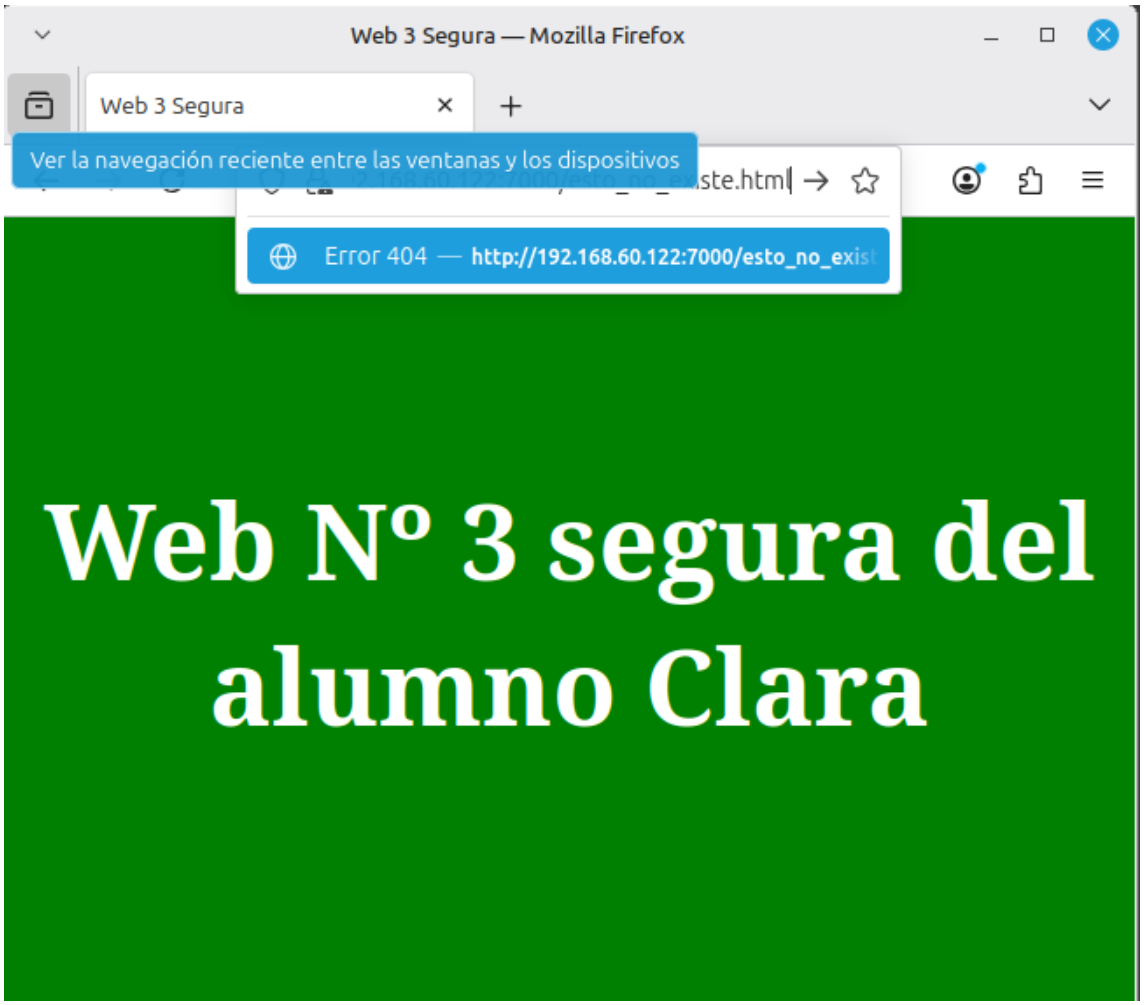
```
sudo systemctl restart apache2
```



Verificación del funcionamiento

Desde el navegador, al acceder a un recurso inexistente del servidor web nº 1, por ejemplo:

```
http://192.168.60.122:7000/esto_no_existe.html
```



El servidor no muestra la página de error local, sino que **redirige automáticamente a la raíz del servidor web nº 3**, cargando correctamente la página principal del servidor seguro.

Este comportamiento confirma que la directiva ErrorDocument 404 está correctamente configurada y que la redirección se realiza de forma automática cuando el recurso solicitado no existe.

Justificación técnica y de seguridad

Esta configuración permite centralizar el tratamiento de errores en un servidor seguro, evitando mostrar información innecesaria sobre el servidor web nº 1.

La redirección automática mejora la experiencia del usuario y refuerza la seguridad al dirigir el tráfico hacia el servidor web nº 3, que ofrece acceso mediante HTTPS.

PRÁCTICA 18. Mostrar documentos del servidor nº 2 mediante alias en la raíz

Objetivo

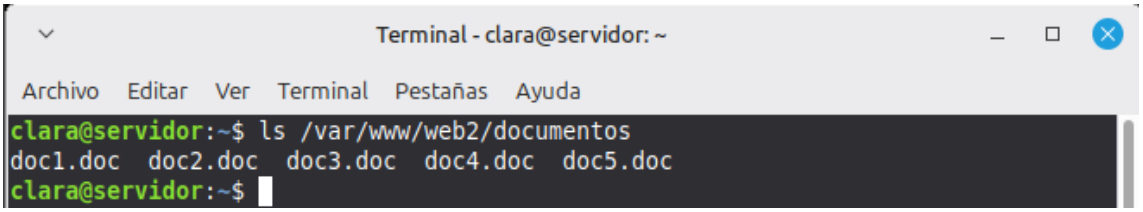
Mostrar mediante alias y sin redirección los contenidos del directorio documentos del servidor nº 2 al acceder a su raíz.

Configuración realizada

Paso 1: Comprobación del directorio documentos

Se verifica que el directorio documentos existe y contiene ficheros:

```
ls /var/www/web2/documentos
```



Paso 2: Edición del VirtualHost del servidor nº 2

Se modifica el VirtualHost para que la raíz del servidor apunte directamente al directorio documentos mediante un alias.

Editar el fichero de configuración del servidor web nº 2:

```
sudo nano /etc/apache2/sites-available/web2.conf
```

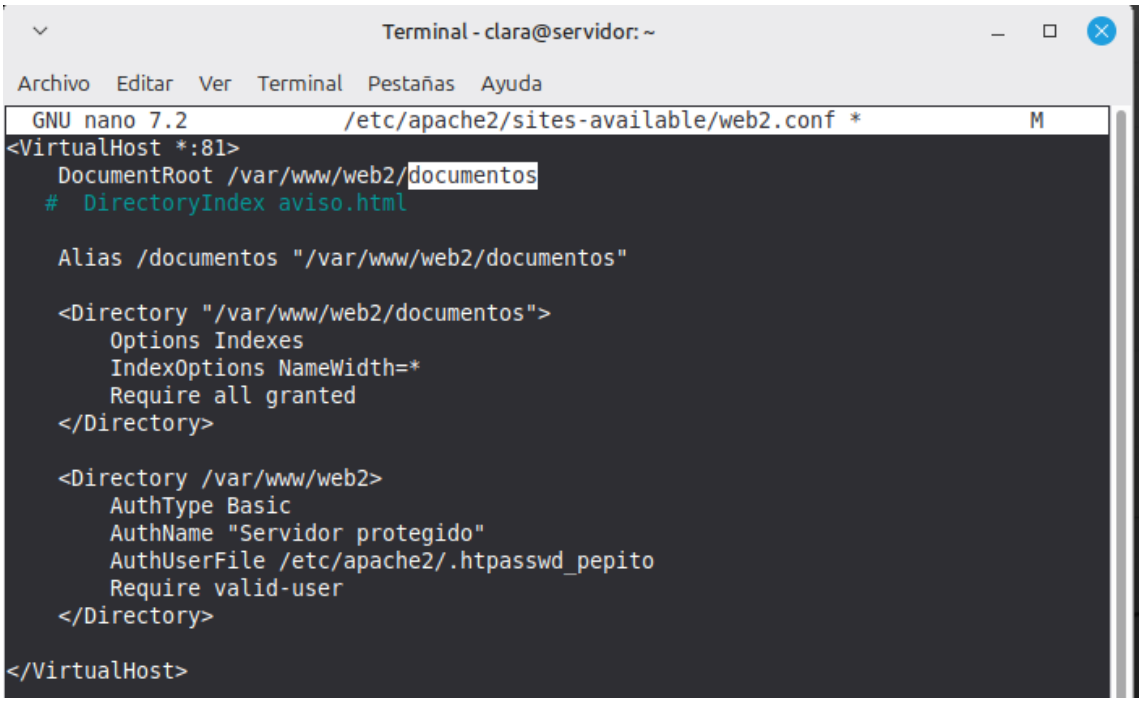
Dentro del bloque `<VirtualHost *:81>`, se añade o ajusta la siguiente directiva:

```
Alias / /var/www/web2/documentos/
```

De este modo, al acceder a la raíz del servidor, Apache sirve directamente el contenido del directorio documentos sin realizar redirecciones externas.

Se mantiene la configuración del listado de directorios:

```
<Directory /var/www/web2/documentos>
    Options Indexes
    IndexOptions NameWidth=*
    Require all granted
</Directory>
```

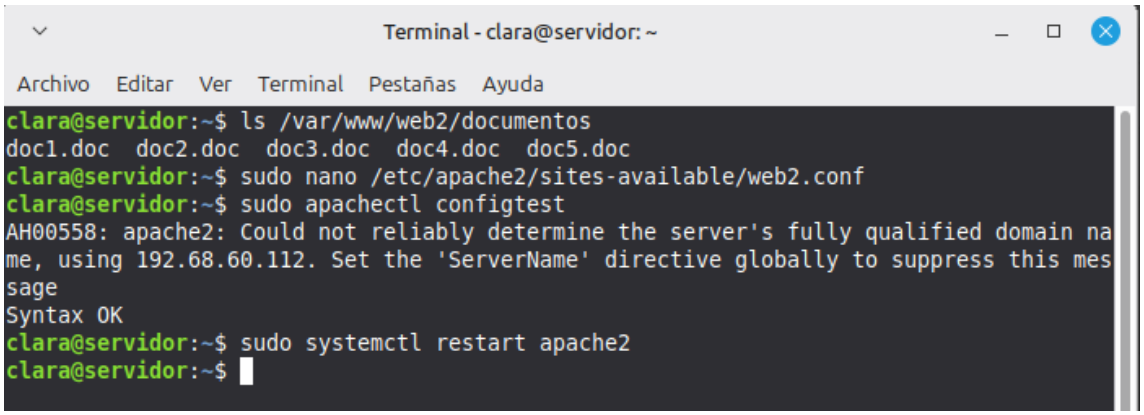


Paso 3: Aplicación de los cambios

Comprobar la sintaxis de la configuración y reiniciar el servicio Apache:

```
sudo apachectl configtest
```

```
sudo systemctl restart apache2
```

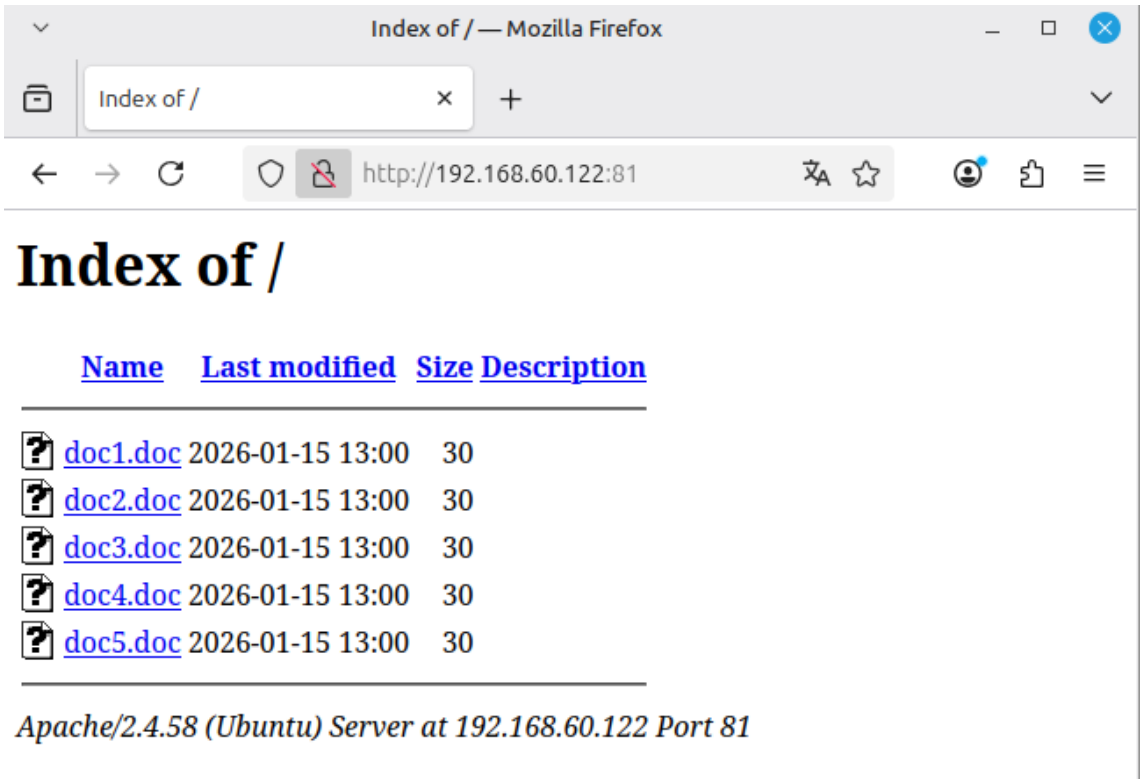


Verificación del funcionamiento

Desde un navegador web, al acceder a:

```
http://192.168.60.122:81/
```

se muestra directamente el listado del contenido del directorio documentos, apareciendo la página **Index of /** generada por Apache.



En el listado se visualizan correctamente los ficheros (doc1.doc, doc2.doc, etc.), con los nombres completos y sin redirección de la URL, confirmando que el acceso se realiza mediante un alias y no mediante una redirección.

Este comportamiento verifica que la directiva Alias / /var/www/web2/documentos/ está correctamente configurada y que Apache sirve el contenido del directorio documentos desde la raíz del servidor web nº 2.

Justificación técnica y de seguridad

Esta configuración simplifica el acceso a los contenidos del servidor web nº 2 al exponer directamente el directorio documentos desde la raíz del sitio.

El uso de un alias permite una administración clara y controlada del servidor web, evitando redirecciones innecesarias y manteniendo una estructura organizada y fácil de mantener.

PRÁCTICA 19. Acceso al directorio documentos mediante alias docs (Servidor nº 1)

Objetivo

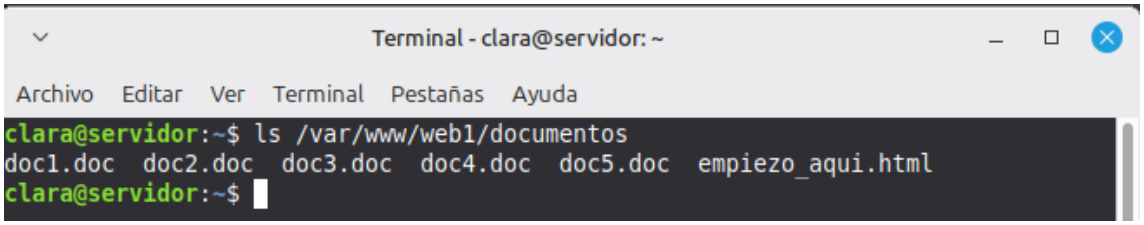
Acceder mediante alias al directorio documentos tecleando docs en la URL del servidor nº 1.

Configuración realizada

Paso 1: Comprobación del directorio documentos

Se verifica que el directorio documentos existe y contiene los ficheros correspondientes:

```
ls /var/www/web1/documentos
```



Paso 2: Edición del VirtualHost del servidor nº 1

Se configura un alias que permita acceder al directorio documentos utilizando la ruta /docs.

Editar el fichero de configuración del servidor web nº 1:

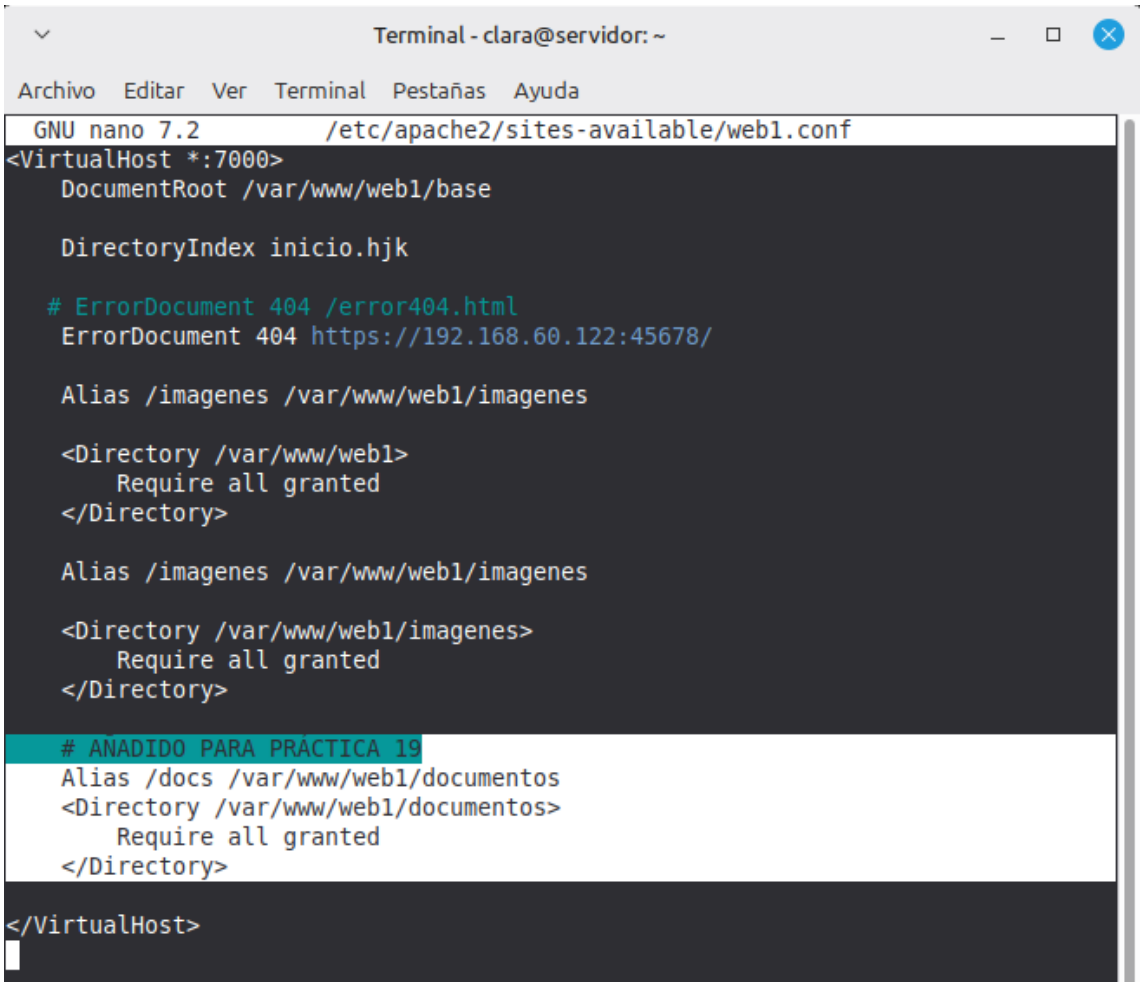
```
sudo nano /etc/apache2/sites-available/web1.conf
```

Añadir la siguiente directiva:

```
Alias /docs /var/www/web1/documentos
```

Y definir el acceso al directorio asociado al alias:

```
<Directory /var/www/web1/documentos>
    Require all granted
</Directory>
```



Esta configuración permite que Apache mapee internamente la URL /docs con el directorio real del sistema de ficheros, sin modificar la URL mostrada al usuario.

Paso 3: Aplicación de los cambios

Comprobar la sintaxis de la configuración y reiniciar el servicio Apache:

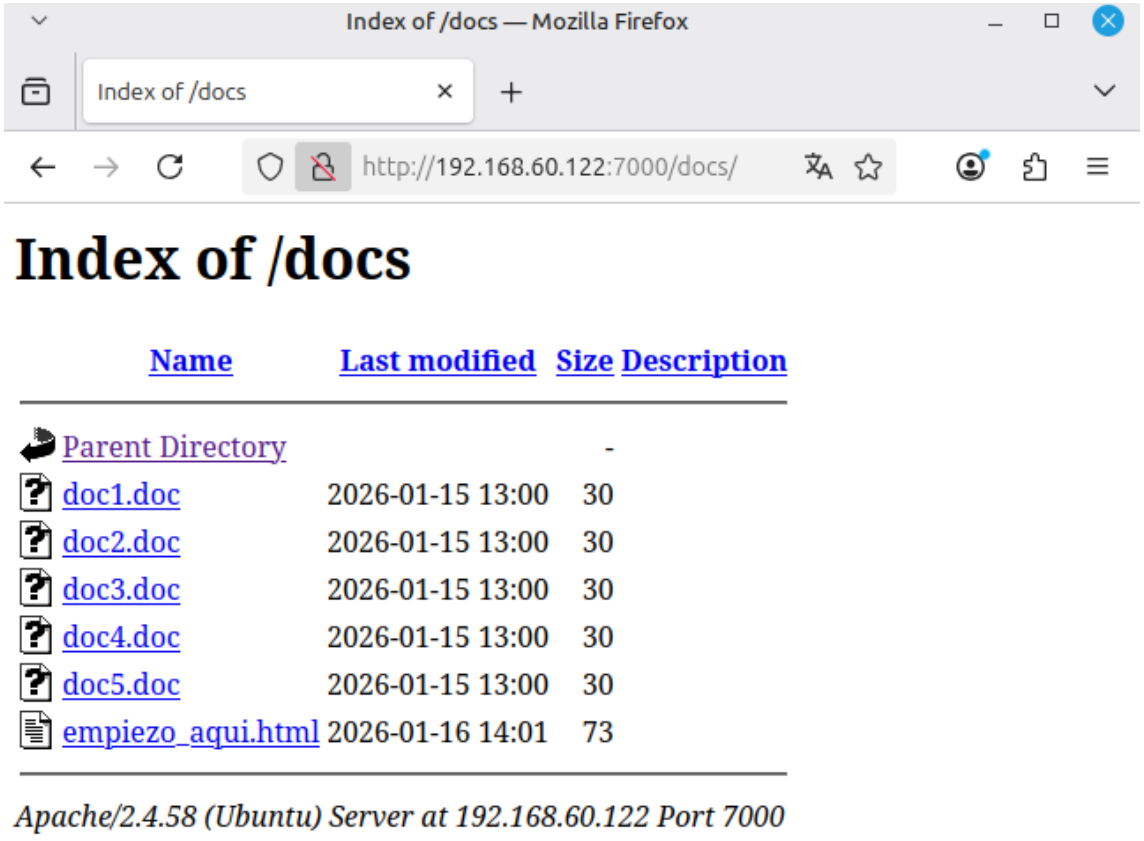
```
sudo apachectl configtest
```

```
sudo systemctl restart apache2
```

Verificación del funcionamiento

Desde un navegador web se accede a:

`http://192.168.60.122:7000/docs`



http://192.168.60.122:7000/docs/doc1.doc

se muestra correctamente el contenido del directorio documentos del servidor web nº 1.

La URL permanece como /docs y no se realiza ninguna redirección, confirmando que el acceso se efectúa mediante un alias configurado en Apache.

Justificación técnica y de seguridad

El uso de alias permite ofrecer rutas de acceso más claras y controladas sin modificar la estructura real del sistema de archivos.

Esta técnica mejora la organización del servidor web y facilita su administración, manteniendo una configuración limpia, segura y fácilmente mantenible.

PRÁCTICA 20. Acceso a URLs sin distinción entre mayúsculas y minúsculas (Servidor nº 3)

Objetivo

Permitir que las URLs del servidor nº 3 sean accesibles independientemente del uso de mayúsculas o minúsculas.

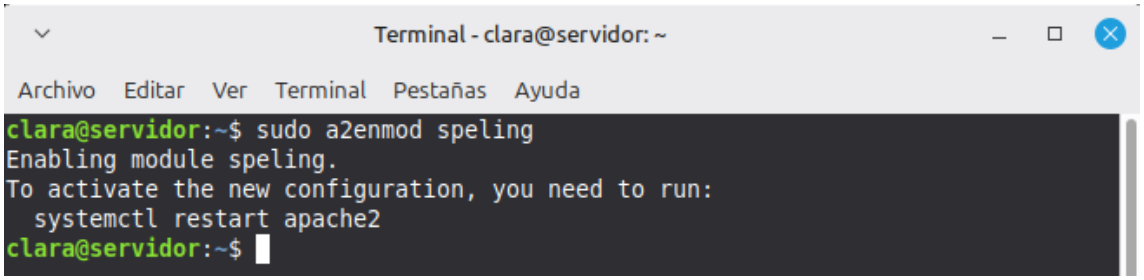
Configuración realizada

Paso 1: Habilitación del módulo necesario de Apache

Apache utiliza el módulo speling (nombre intencionadamente mal escrito) para corregir automáticamente diferencias en el uso de mayúsculas y minúsculas en las URLs.

Activar el módulo:

```
sudo a2enmod speling
```



Paso 2: Reinicio de Apache para aplicar el módulo

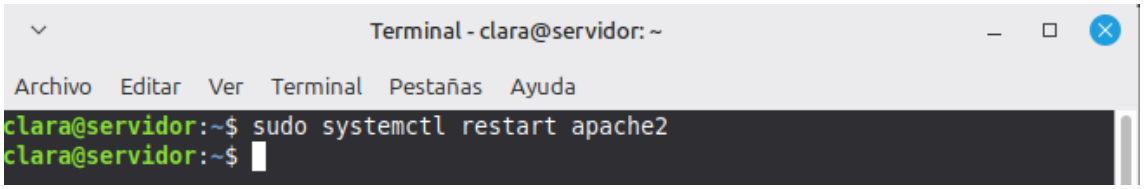
Reiniciar el servicio Apache para cargar el módulo activado:

```
sudo systemctl restart apache2
```

Paso 3: Activación de la corrección de mayúsculas y minúsculas

Editar el VirtualHost seguro del servidor web nº 3:

```
sudo nano /etc/apache2/sites-available/web3-ssl.conf
```

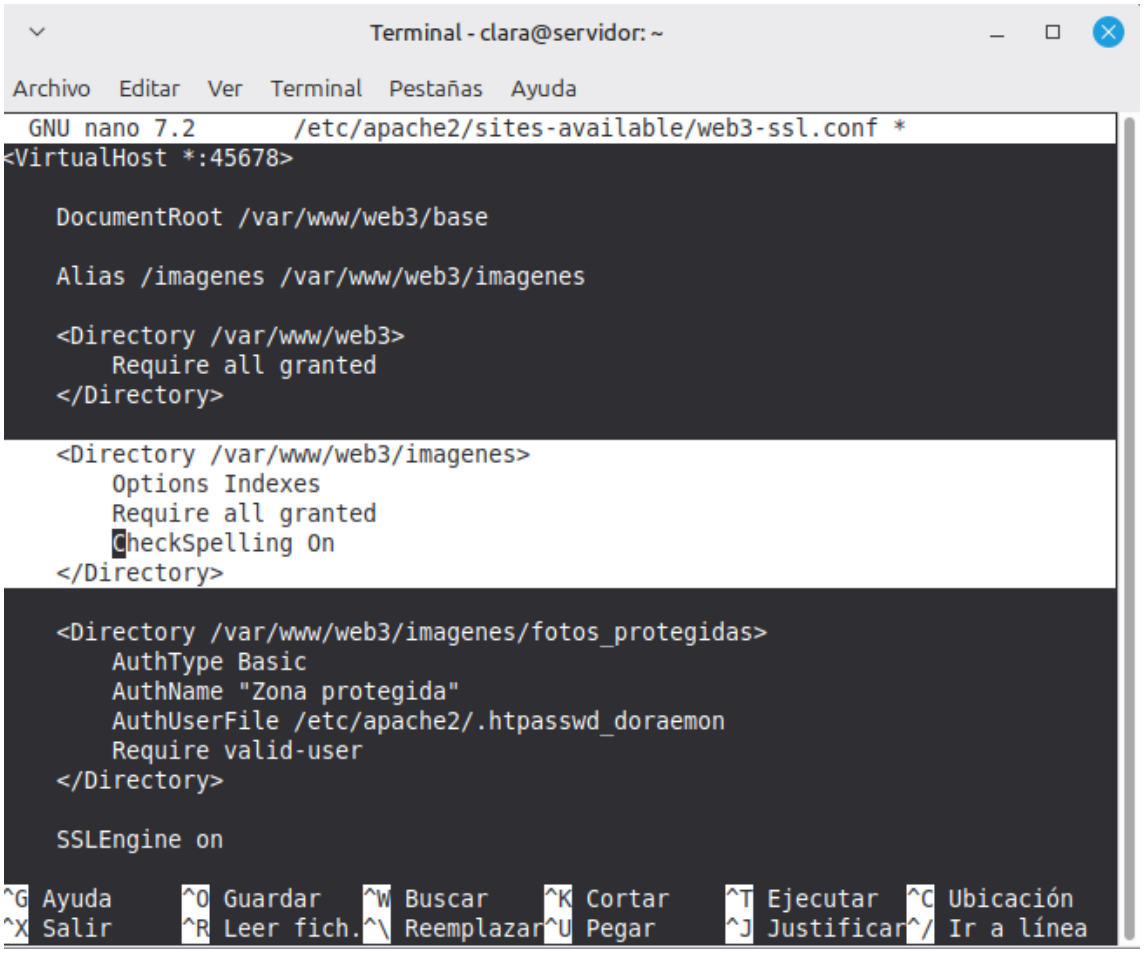


Dentro del bloque <VirtualHost *:45678>, **sin eliminar ninguna configuración existente**, añadir la siguiente directiva:

```
CheckSpelling On
```

Además, dentro del bloque correspondiente al directorio de imágenes, habilitar también la corrección:

```
<Directory /var/www/web3/imagenes>
    Options Indexes
    Require all granted
    CheckSpelling On
</Directory>
```



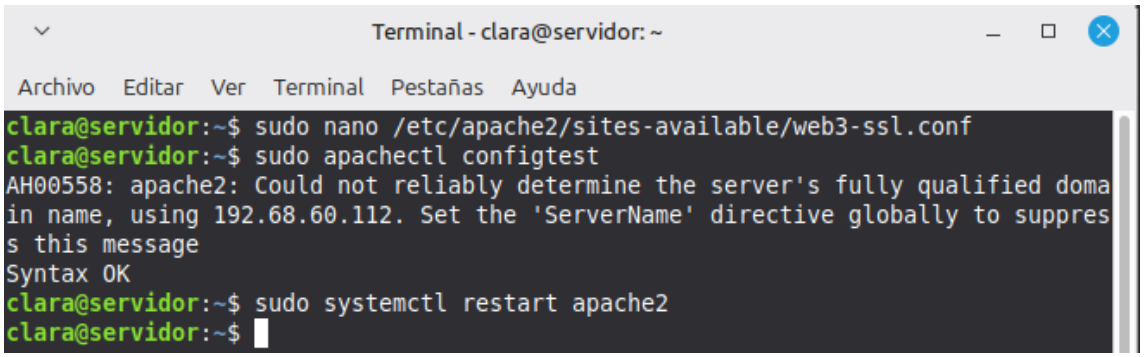
Con esta configuración, Apache corrige automáticamente las diferencias de mayúsculas y minúsculas al acceder a los recursos.

Paso 4: Comprobación de la configuración y reinicio

Verificar la sintaxis de la configuración y reiniciar Apache:

```
sudo apachectl configtest
```

```
sudo systemctl restart apache2
```



Verificación del funcionamiento

Desde el navegador web, se accede al mismo recurso utilizando distintas combinaciones de mayúsculas y minúsculas, por ejemplo:

`https://192.168.60.122:45678/imagenes/mayusminus.jpg`

y

https://192.168.60.122:45678/imagenes/MAYUSMINUS.jpg

En ambos casos, el servidor devuelve correctamente el recurso solicitado, confirmando que la corrección automática de mayúsculas y minúsculas funciona de forma adecuada.

Justificación técnica y de seguridad

El uso del módulo mod_speling mejora la usabilidad del servidor web al evitar errores 404 provocados por diferencias en el uso de mayúsculas y minúsculas en las URLs.

Esta configuración facilita el acceso a los recursos sin comprometer la seguridad del servidor y contribuye a una experiencia de usuario más tolerante y controlada.

PRÁCTICA 21. Sustitución automática de docs por documentos (Servidor nº 1)

Objetivo

Hacer que al teclear la cadena docs en una URL se sustituya automáticamente por documentos.

Configuración realizada

Paso 1: Edición del VirtualHost del servidor nº 1

Editar el fichero de configuración del servidor web nº 1:

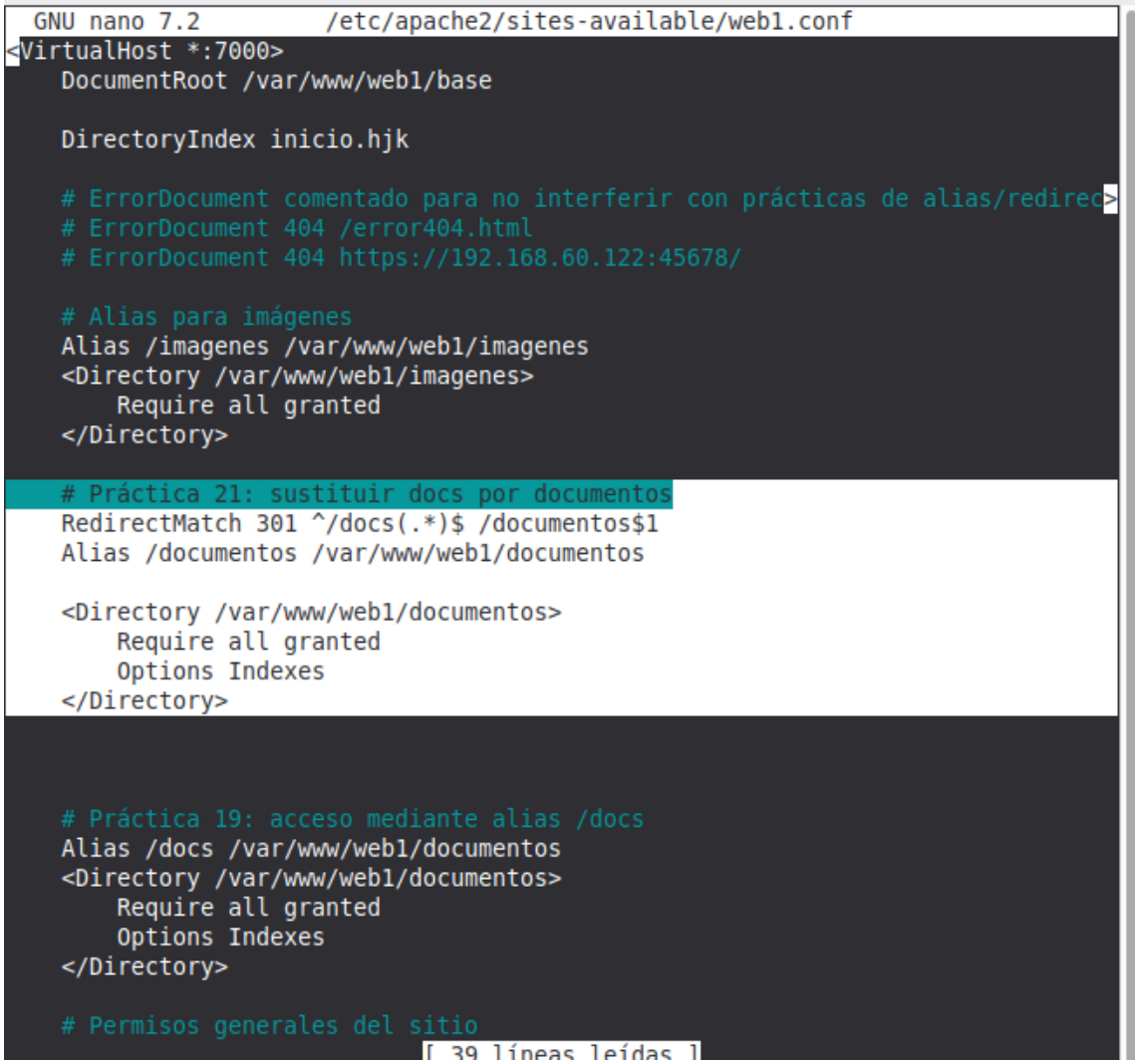
```
sudo nano /etc/apache2/sites-available/web1.conf
```

Dentro del bloque `<VirtualHost *:7000>`, añadir las siguientes directivas sin eliminar ninguna configuración existente:

```
RedirectMatch 301 ^/docs(.*)$ /documentos$1

Alias /documentos /var/www/web1/documentos
```

Además, definir el acceso al directorio asociado:

```
<Directory /var/www/web1/documentos>
    Require all granted
    Options Indexes
</Directory>
```



Con esta configuración:

- Cualquier acceso a una URL que contenga /docs se redirige automáticamente a /documentos.
- El alias /documentos apunta internamente al directorio real del servidor.
- El acceso al contenido se realiza de forma controlada y transparente.

Paso 2: Aplicación de los cambios

Comprobar la sintaxis de la configuración y reiniciar el servicio Apache:

```
sudo apachectl configtest
```

```
sudo systemctl restart apache2
```

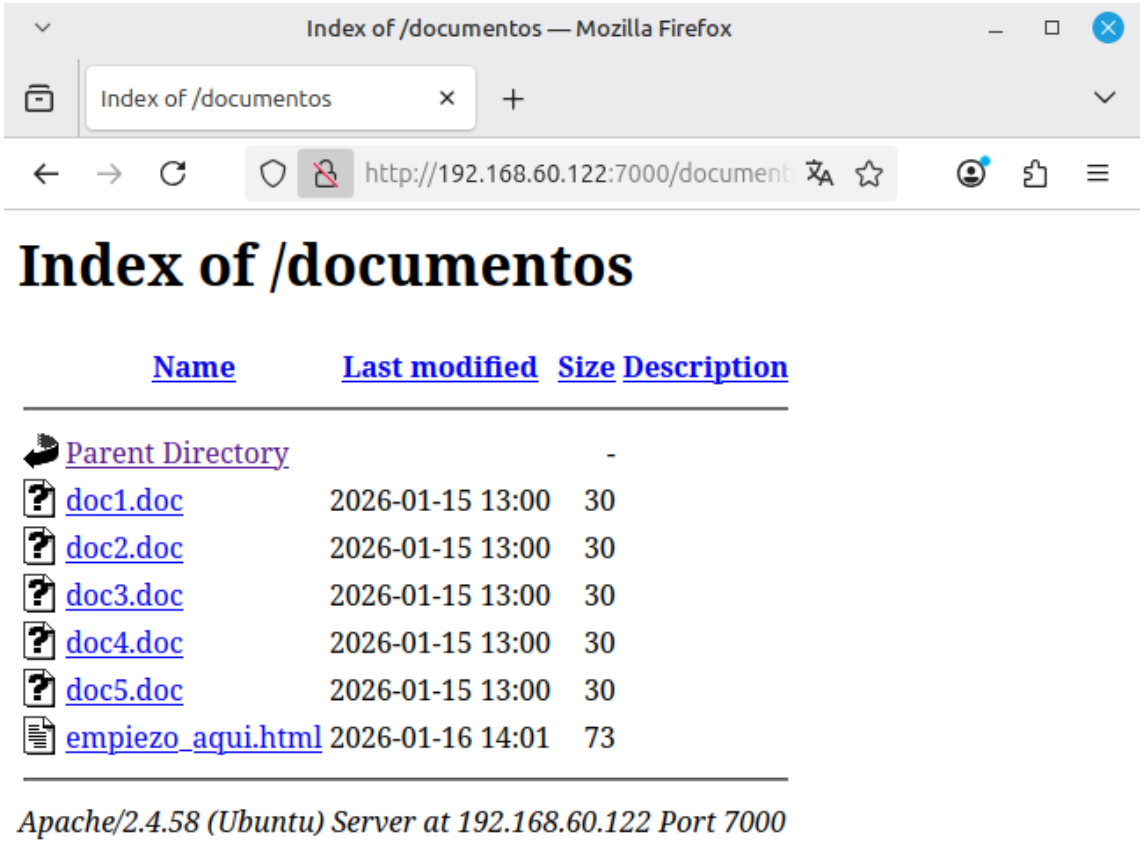
Verificación del funcionamiento

Desde el navegador web, al acceder a:

```
http://192.168.60.122:7000/docs
```

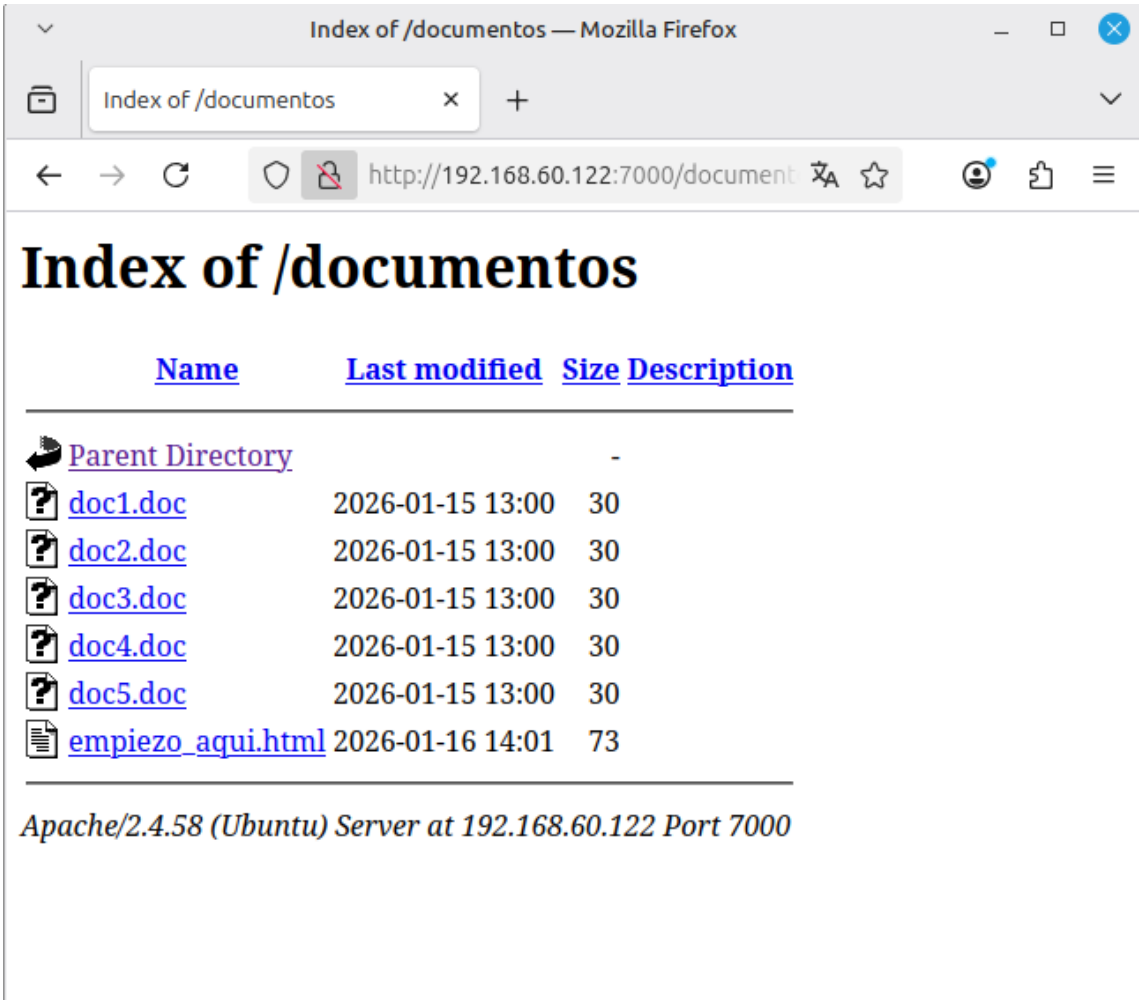
la URL se sustituye automáticamente por:

```
http://192.168.60.122:7000/documentos
```



Asimismo, el acceso directo a:

`http://192.168.60.122:7000/documentos`



muestra correctamente el contenido del directorio documentos.

Este comportamiento confirma que la sustitución automática de la cadena docs por documentos funciona correctamente.

Justificación técnica y de seguridad

La directiva RedirectMatch permite normalizar las URLs y mantener coherencia en las rutas de acceso al contenido del servidor.

La combinación de redirección y alias facilita la administración del servidor web, mejora la claridad de las URLs y permite reorganizar el acceso a los recursos sin modificar la estructura física del sistema de archivos.

PRÁCTICA 22. Restricción de acceso por IP al directorio fotos_protegidas (Servidor nº 3)

Objetivo

Restringir el acceso al directorio fotos_protegidas para que solo sea accesible desde las dos IP activas anterior y posterior a la IP propia.

Configuración realizada

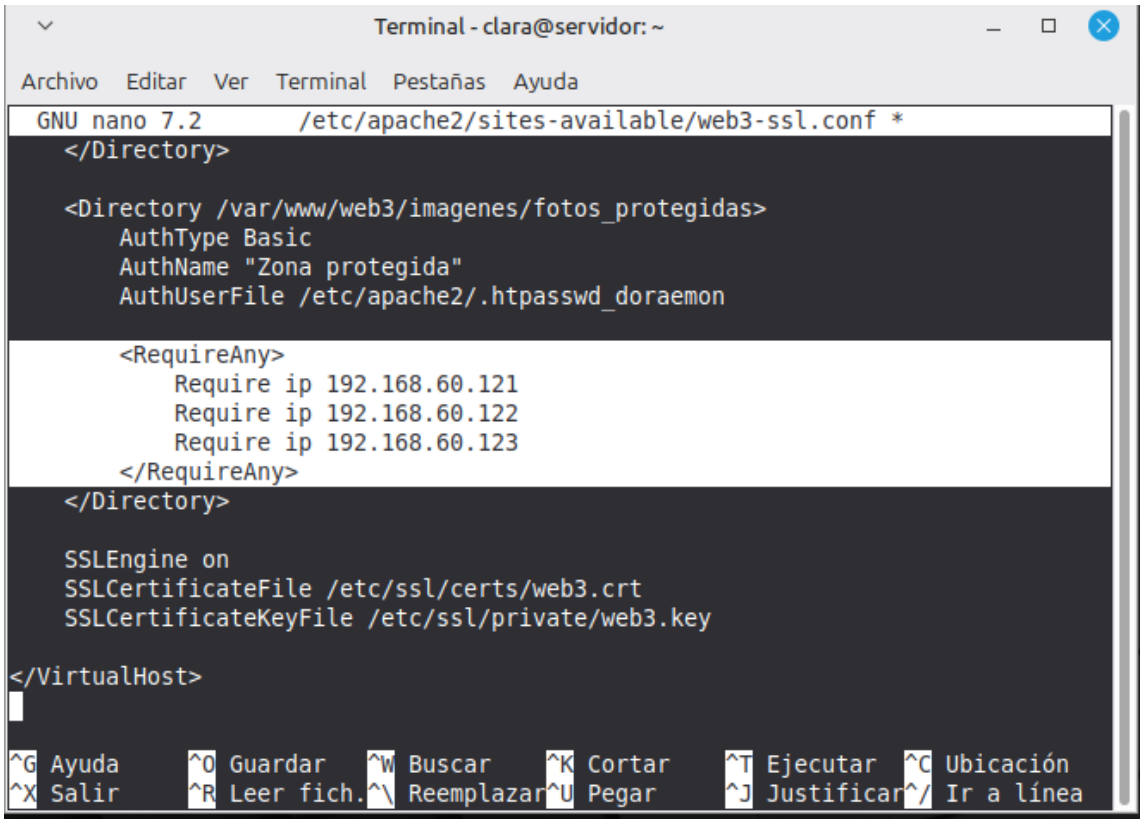
Paso 1: Edición del VirtualHost HTTPS del servidor nº 3

Se edita el fichero de configuración del servidor web seguro:

```
sudo nano /etc/apache2/sites-available/web3-ssl.conf
```

Dentro del fichero, se mantiene la configuración de autenticación básica y se añade la restricción por direcciones IP en el bloque correspondiente al directorio protegido:

```
<Directory /var/www/web3/imagenes/fotos_protegidas>
    AuthType Basic
    AuthName "Zona protegida"
    AuthUserFile /etc/apache2/.htpasswd_doraemon

    Require ip 192.168.60.121
    Require ip 192.168.60.122
    Require ip 192.168.60.123
</Directory>
```



Con esta configuración:

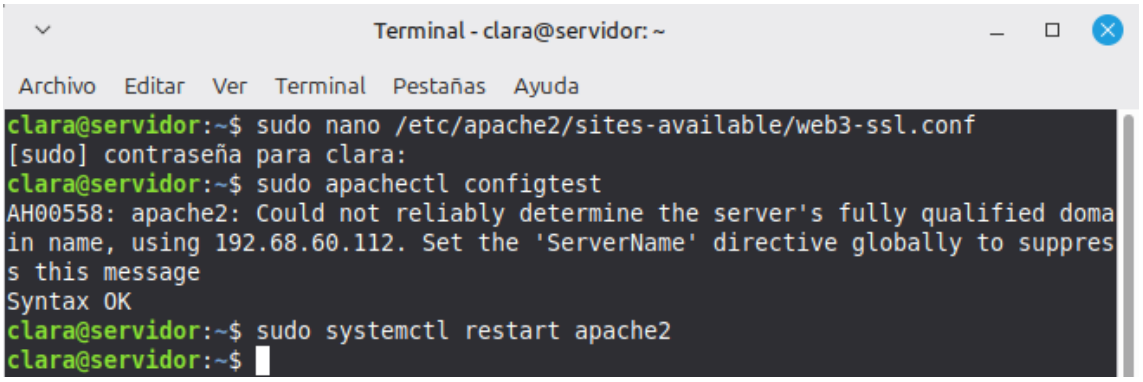
- Se mantiene la autenticación mediante usuario y contraseña.
- El acceso solo se permite desde las IP indicadas.
- Apache exige que se cumpla al menos una de las direcciones IP autorizadas.

Paso 2: Aplicación de los cambios

Se comprueba la sintaxis de la configuración y se reinicia el servicio Apache:

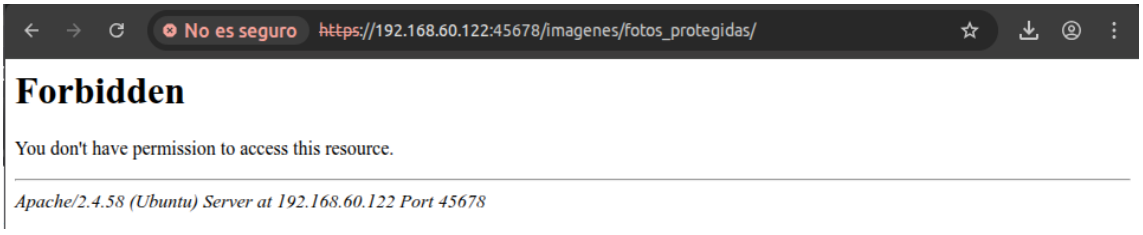
```
sudo apachectl configtest
```

```
sudo systemctl restart apache2
```



Verificación del funcionamiento

Al intentar acceder desde la `IP 192.168.60.22`, el acceso queda bloqueado porque esa dirección no está incluida en las IP permitidas (192.168.60.121, 192.168.60.122 y 192.168.60.123).



Al acceder desde una IP autorizada (por ejemplo desde el propio servidor con `IP 192.168.60.122`), el servidor solicita usuario y contraseña, y tras introducir credenciales válidas se permite el acceso al contenido del directorio.

Este comportamiento confirma que la restricción por IP se aplica correctamente y que el directorio solo es accesible desde las direcciones autorizadas.

Justificación técnica y de seguridad

La restricción de acceso por dirección IP, combinada con la autenticación básica, permite limitar el acceso al directorio fotos_protegidas únicamente a usuarios y equipos autorizados dentro de la red.

Esta configuración añade una capa adicional de seguridad al servidor web nº 3, reduciendo el riesgo de accesos no autorizados y permitiendo una administración centralizada del control de acceso sin utilizar ficheros .htaccess.

PRÁCTICA 23. Creación de un subdirectorio accesible sin restricciones manteniendo la protección del directorio padre (Servidor nº 3)

Objetivo

Crear un directorio fotos_protegidas2, dependiente de fotos_protegidas, accesible sin restricciones, manteniendo la protección del directorio padre.

Configuración realizada

Paso 1: Creación del directorio fotos_protegidas2

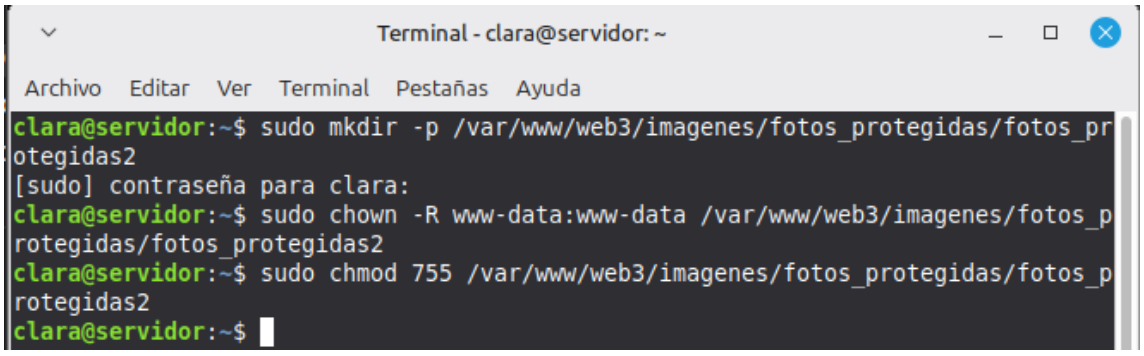
Se crea el nuevo directorio dentro del directorio protegido:

```
sudo mkdir -p /var/www/web3/imagenes/fotos_protegidas/fotos_protegidas2
```

Se asignan los permisos adecuados para que Apache pueda acceder al contenido:

```
sudo chown -R www-data:www-data
/var/www/web3/imagenes/fotos_protegidas/fotos_protegidas2
```

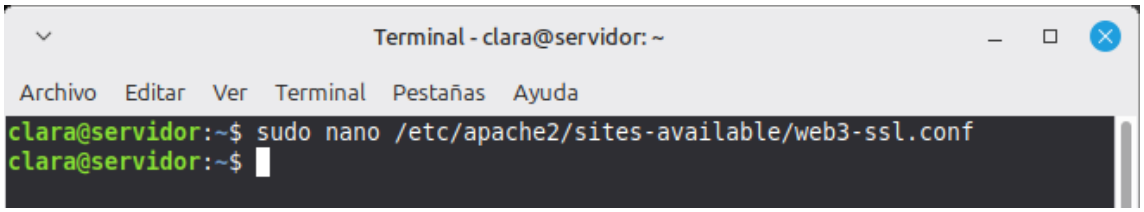
```
sudo chmod 755
/var/www/web3/imagenes/fotos_protegidas/fotos_protegidas2
```



Paso 2: Edición del VirtualHost HTTPS del servidor nº 3

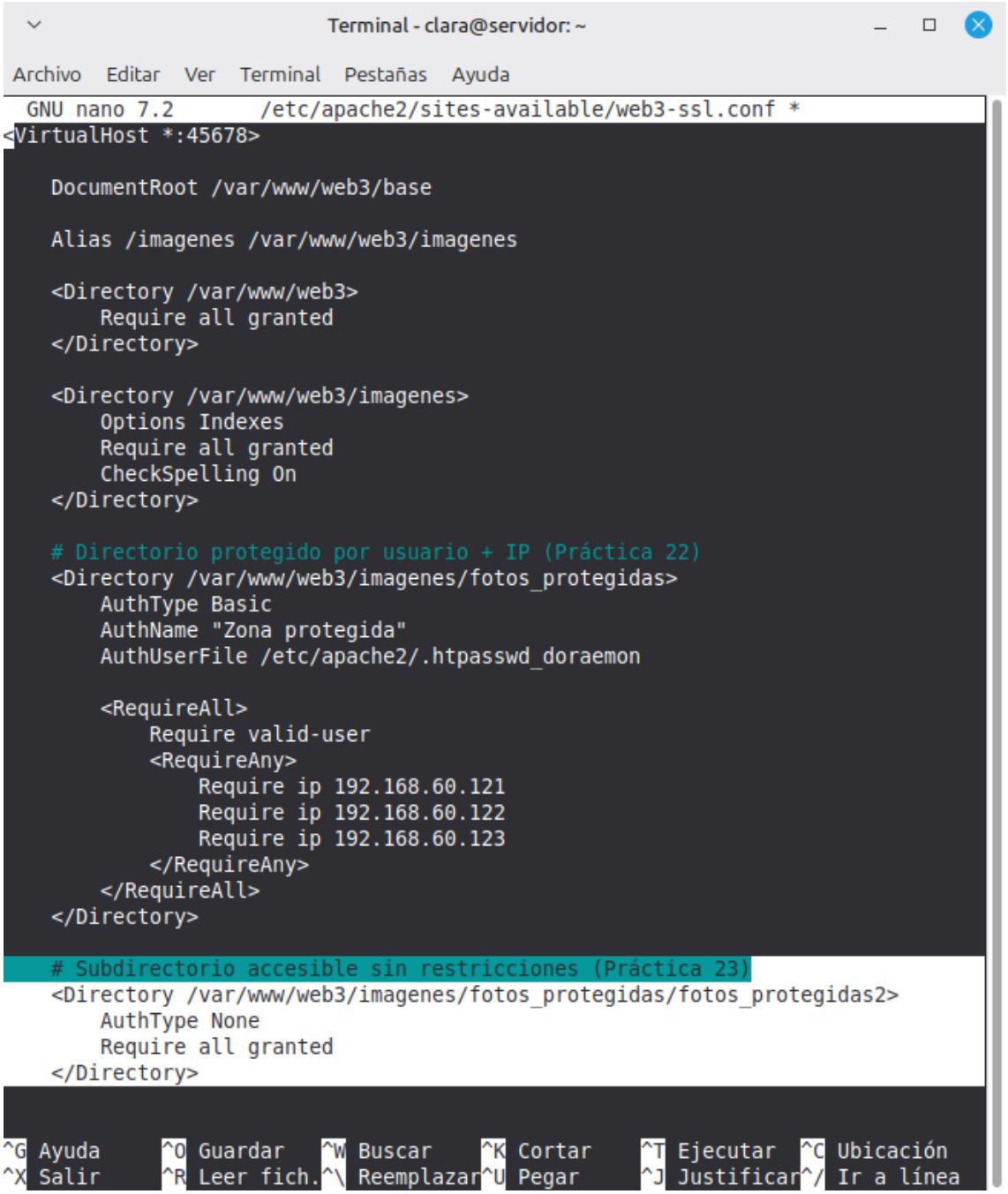
Se edita el fichero de configuración del servidor web seguro:

```
sudo nano /etc/apache2/sites-available/web3-ssl.conf
```



Dentro del fichero, se mantiene la protección del directorio padre fotos_protegidas y se añade un bloque específico para el subdirectorio fotos_protegidas2 que anula la autenticación:

```
<Directory /var/www/web3/imagenes/fotos_protegidas/fotos_protegidas2>
    AuthType None
    Require all granted
</Directory>
```



Con esta configuración:

El directorio fotos_protegidas continúa protegido por autenticación e IP.

El subdirectorio fotos_protegidas2 es accesible sin usuario ni contraseña.

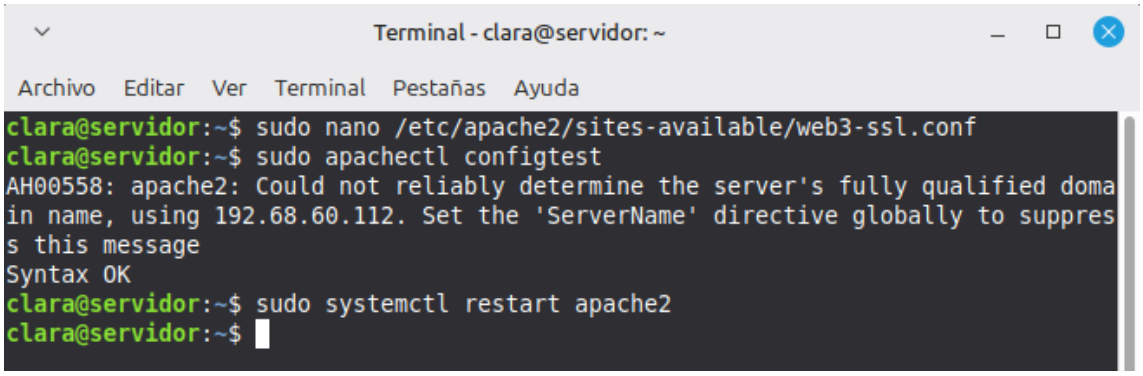
Se sobrescriben las restricciones heredadas del directorio padre únicamente para este subdirecto.

Paso 3: Aplicación de los cambios

Se comprueba la sintaxis de la configuración y se reinicia el servicio Apache:

```
sudo apachectl configtest
```

```
sudo systemctl restart apache2
```



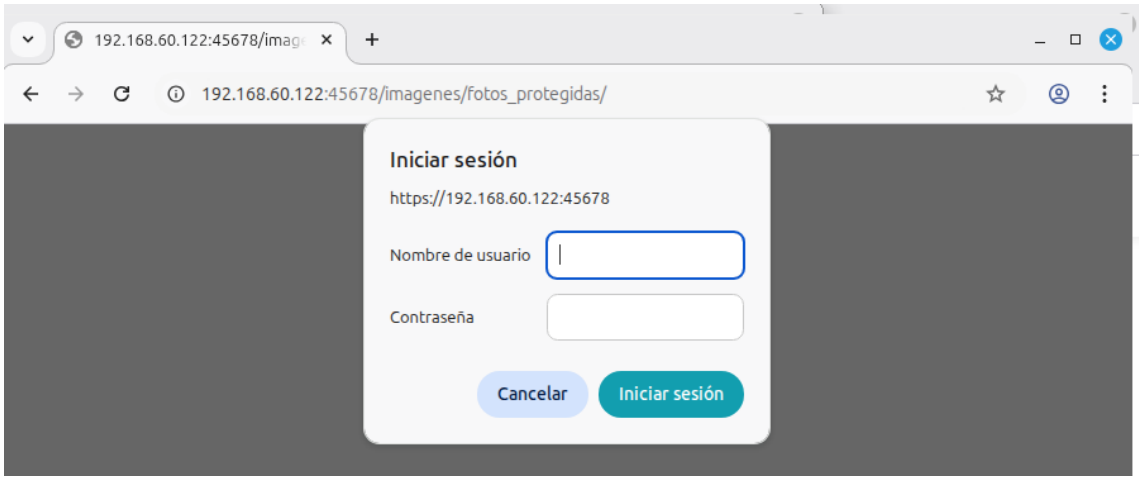
Verificación del funcionamiento

Se accede desde el navegador a los siguientes recursos:

Acceso al directorio padre:

```
https://192.168.60.122:45678/imagenes/fotos_protegidas/
```

El servidor solicita autenticación y aplica la restricción por IP.



Acceso al subdirectorio, Entra sin pedir autenticación

Justificación técnica y de seguridad

Esta configuración permite proteger el acceso al directorio fotos_protegidas del servidor web nº 3 mediante autenticación básica y restricción por direcciones IP, evitando accesos no autorizados.

El subdirectorio fotos_protegidas2 es accesible sin restricciones, manteniendo la protección del directorio padre y demostrando el control de acceso a nivel de servidor sin utilizar ficheros .htaccess.

Recomendaciones de uso seguro del servidor

Para garantizar un **funcionamiento seguro y estable del servidor web Apache**, se recomienda aplicar las siguientes **medidas de uso y administración**:

- Mantener el servidor Apache y el sistema operativo **actualizados**, aplicando periódicamente las **actualizaciones** de **seguridad** disponibles.
- Utilizar siempre **conexiones seguras** mediante **HTTPS** para **proteger la información transmitida entre el servidor y los clientes**.
- **Limitar el acceso a recursos sensibles** mediante **autenticación** y **restricciones por dirección IP** siempre que sea posible.
- **Revisar de forma periódica los ficheros de log** del servidor para detectar accesos anómalos, errores o intentos de uso indebido.
- **Gestionar el control de acceso** directamente desde los ficheros de configuración de Apache, evitando el uso de ficheros .htaccess en entornos controlados.
- **Proteger adecuadamente los directorios** que contengan información sensible, aplicando permisos y directivas de acceso acordes a su nivel de seguridad.

Conclusiones

Durante el desarrollo de estas prácticas se ha **trabajado de forma progresiva** con la configuración avanzada del servidor web Apache, aplicando **redirecciones**, **alias**, **control de errores**, **autenticación**, **restricciones** por **IP** y **gestión** de **permisos** en directorios. Esto ha permitido **comprender** mejor cómo funciona Apache en un entorno real y cómo afectan las distintas directivas al comportamiento del servidor.

A lo largo del proceso han surgido **diversas incidencias prácticas**, especialmente **relacionadas con la interacción entre distintas directivas como Alias**, **Redirect**, **ErrorDocument**, **AuthType** o **CheckSpelling**. En varios casos, configuraciones **aparentemente correctas no funcionaban debido a redirecciones previas**, **reglas solapadas** o a la **herencia de configuraciones en los directorios**, lo que ha **obligado a revisar y ajustar cuidadosamente los ficheros de configuración**.

Uno de los aspectos más complejos ha sido entender cómo Apache hereda las restricciones de seguridad en los subdirectorios. Proteger un directorio padre implica automáticamente proteger sus subdirectorios, y **permitir excepciones requiere una configuración explícita**. Este comportamiento se ha podido comprobar en la práctica de creación de un subdirectorio accesible sin restricciones dentro de un directorio protegido, donde fue necesario redefinir el contexto de autenticación para evitar conflictos.

En conjunto, estas prácticas han permitido **afianzar conocimientos prácticos sobre la administración segura de servidores web**, aprendiendo a **diagnosticar y resolver problemas reales de configuración**. El resultado final es una **infraestructura funcional**, **coherente** y **segura**, que refleja un **aprendizaje progresivo** y una **mejor comprensión del funcionamiento interno de Apache**.