

PRÁCTICA 3

INSTALACIÓN DE SERVIDORES WEB



CLARA MONTAÑO RODRÍGUEZ

ÍNDICE

ADECUACIÓN DE LA PRÁCTICA AL ENUNCIADO Y CRITERIOS DE EVALUACIÓN.....	4
1. Correspondencia con el enunciado	4
2. Correspondencia con los criterios de evaluación.....	5
ESCENARIO	6
FASE 0: IP Y RED	6
FASE 1: COMPROBAR SI APACHE ESTÁ INSTALADO Y QUÉ VERSIÓN TIENE.....	7
1.1 Instalación del servidor web Apache.....	7
1.2 Habilitar e iniciar Apache	8
1.3 Verificación de la instalación	8
FASE 2: CONFIGURACIÓN DE PUERTOS Y ARQUITECTURA APACHE.....	9
2.1 Copia de seguridad del fichero de puerto	9
2.2 Copia de seguridad del fichero de puerto	9
2.3 Desactivar el sitio por defecto de Apache	11
2.4 Comprobar sintaxis y reiniciar Apache.....	12
2.5 Verificar que Apache escucha en los puertos requeridos	12
FASE 3: ESTRUCTURA DE DIRECTORIOS Y CONTENIDOS	13
3.1 Creación de la estructura	13
3.2 Asignación de propietario y permisos	14
3.3 Creación del contenido mínimo en documentos e imágenes	15
3.4 Comprobación del contenido creado	16
FASE 4: SERVIDOR WEB Nº 1 (PUERTO 80 - HTTP)	18
4.1 Crear el VirtualHost de la Web nº1	18
4.2 Activar el sitio web nº1.....	19
4.3 Crear el contenido HTML (fondo rojo)	20

4.4 Comprobación del acceso a la Web nº1	22
FASE 5: SERVIDOR WEB Nº 2 (PUERTO 81 - HTTP)	24
5.1 Crear el VirtualHost de la Web nº 2	24
5.2 Activar el sitio web nº 2.....	25
5.3 Crear el contenido HTML de la Web nº 2 (fondo amarillo)	26
5.4 Crear los otros 4 HTML obligatorios (web 2)	27
5.5 Verificación del servidor web nº 2.....	28
5.6 Comprobación del acceso a la Web nº 2 desde el navegador	29
FASE 6: SERVIDOR WEB Nº 3 (HTTPS - PUERTO 443).....	30
6.1 Activar el módulo SSL de Apache	30
6.2 Crear el certificado SSL autofirmado	31
6.3 Crear el VirtualHost HTTPS (web3)	34
6.4 Activar el sitio web seguro	35
6.5 Crear la página HTML de la Web nº 3 (verde).....	36
6.6 Creación de los ficheros HTML adicionales de la Web nº 3	37
6.7 Verificación del servidor web seguro desde terminal	38
6.8 Verificación del servidor web seguro desde el navegador	39
6.9 Verificación final de puertos en escucha	43
INFORMACIÓN Y CONTENIDOS IMPORTANTES	44
1. JUSTIFICACIÓN DE LA ARQUITECTURA UTILIZADA.....	44
2. DIFERENCIA ENTRE SERVIDORES, PUERTOS Y NIVELES DE SEGURIDAD	44
3. DESCRIPCIÓN DE HTTP Y HTTPS	45
4. RELACIÓN ENTRE PROTOCOLOS, PUERTOS Y FUNCIONAMIENTO DEL SERVIDOR APACHE	45
CONCLUSIÓN	46

PRÁCTICA 3

ADECUACIÓN DE LA PRÁCTICA AL ENUNCIADO Y CRITERIOS DE EVALUACIÓN

1. Correspondencia con el enunciado

Requisito del enunciado	Dónde se encuentra en el trabajo
IP asignada a la máquina virtual	FASE 0: IP Y RED
Instalación y verificación de Apache	FASE 1 (1.1, 1.2 y 1.3)
Uso exclusivo de dirección IP (sin DNS)	ESCENARIO y FASE 0
Tres webs en la misma máquina virtual	FASE 2 y justificación teórica (Información, punto 1)
Estructura base / documentos / imagenes	FASE 3 (3.1 a 3.4)
5 HTML, 5 DOC y 5 JPG por web	FASE 3.3 y fases 4, 5 y 6
Web nº 1 – HTTP, puerto 80, fondo rojo	FASE 4
Web nº 2 – HTTP, puerto 81, fondo amarillo	FASE 5
Web nº 3 – HTTPS, puerto 443, certificado	FASE 6
Certificado autofirmado	FASE 6.2
Acceso correcto a las tres webs	FASES 4.4, 5.6 y 6.8
Servicio Apache activo	FASE 1.3
Puertos 80, 81 y 443 en escucha	FASE 6.9

2. Correspondencia con los criterios de evaluación

Requisito del enunciado	Dónde se encuentra en el trabajo
Justificación de la arquitectura web	Información y contenidos importantes – Punto 1
Diferenciación entre servidores, puertos y seguridad	Información – Punto 2 y Fases 4, 5 y 6
Descripción de HTTP y HTTPS	Información – Punto 3
Relación entre protocolos, puertos y Apache	Información – Punto 4
Documentación de instalación y configuración	Fases 1 a 6
Evidencias gráficas y verificaciones	Fases 4.4, 5.6, 6.8 y 6.9
Claridad, orden y presentación	Estructura general del documento

El documento se ha estructurado siguiendo un índice propio con el objetivo de facilitar el seguimiento del proceso completo de configuración, permitiendo que la práctica pueda utilizarse como guía o manual de referencia, sin que ello afecte al cumplimiento de los requisitos ni de los criterios de evaluación establecidos en el enunciado.

ESCENARIO

La práctica se desarrollará en una máquina virtual con Linux Mint XFCE, dentro del entorno Mx 12. El servidor web Apache puede encontrarse instalado o no. En caso de no estarlo, deberá realizarse su instalación previa, así como la comprobación de su correcto funcionamiento. Todas las tareas de instalación y configuración deberán ser debidamente documentadas, formando parte esencial de la evaluación.

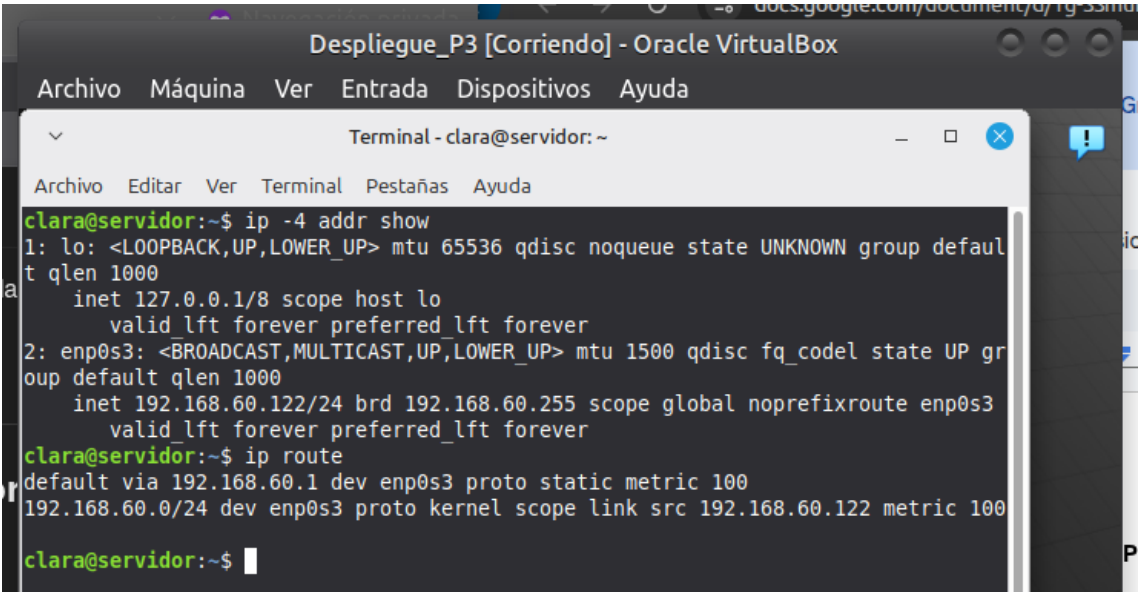
FASE 0: IP Y RED

IP asignada a la máquina virtual:

La máquina virtual tiene asignada la dirección IP **192.168.60.122/24** en la interfaz `enp0s3`. La puerta de enlace por defecto es **192.168.60.1**, lo que indica conectividad correcta dentro de la red **192.168.60.0/24**. El acceso a los servidores se realiza exclusivamente mediante dirección IP, sin uso de DNS.

```
ip -4 addr show
```

```
ip route
```



The screenshot shows a terminal window titled "Terminal - clara@servidor: ~" within an Oracle VM VirtualBox environment. The window has a menu bar with "Archivo", "Editar", "Ver", "Terminal", "Pestañas", and "Ayuda". The terminal output shows the following commands and results:

```
clara@servidor:~$ ip -4 addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    inet 192.168.60.122/24 brd 192.168.60.255 scope global noprefixroute enp0s3
        valid_lft forever preferred_lft forever
clara@servidor:~$ ip route
default via 192.168.60.1 dev enp0s3 proto static metric 100
192.168.60.0/24 dev enp0s3 proto kernel scope link src 192.168.60.122 metric 100
clara@servidor:~$
```

FASE 1: COMPROBAR SI APACHE ESTÁ INSTALADO Y QUÉ VERSIÓN TIENE

Primero verificamos si tenemos Apache y en qué versión lo tenemos.

```
apache2 -v
```

```
clara@servidor:~$ apache2 -v
No se ha encontrado la orden «apache2», pero se puede instalar con:
sudo apt install apache2-bin
clara@servidor:~$
```

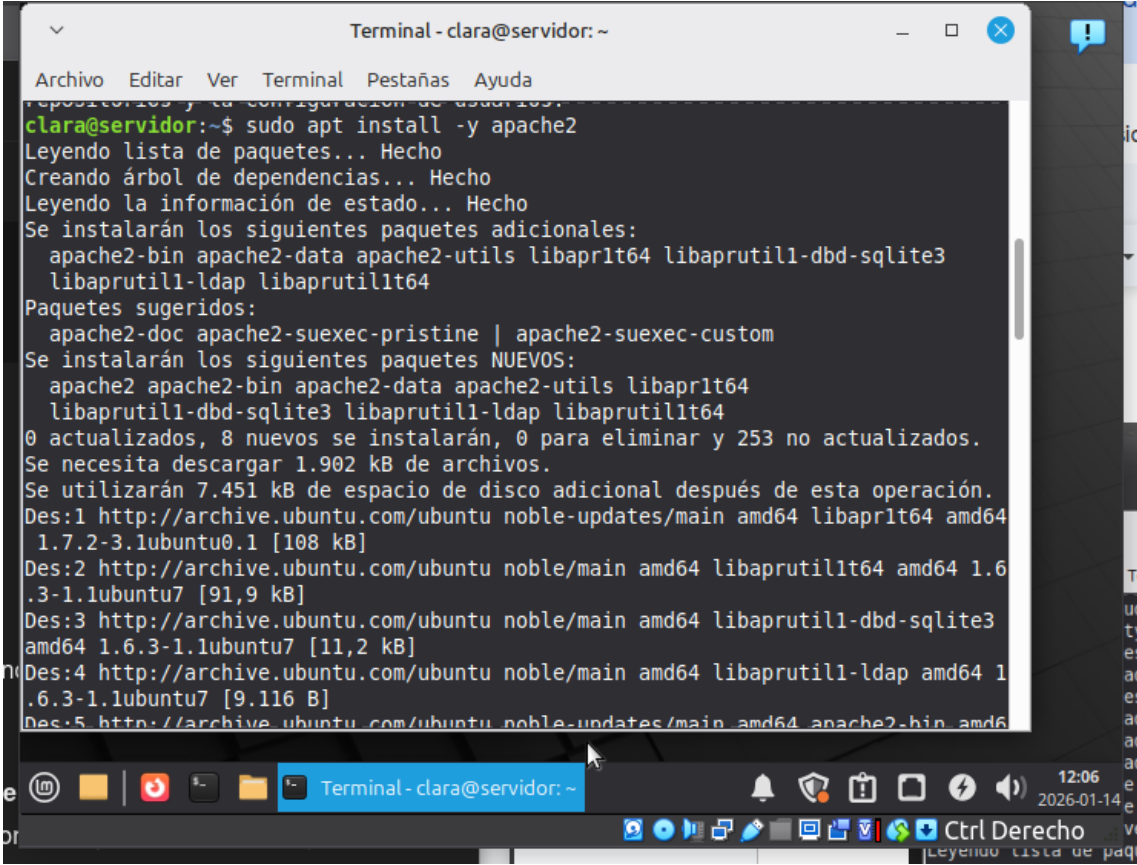
Aquí vemos que no tenemos instalado Apache por lo que vamos a instalarlo.

```
sudo apt update
```

```
sudo apt install -y apache2
```

1.1 Instalación del servidor web Apache

Se procede a la instalación del servidor web Apache mediante el gestor de paquetes APT (apt install apache2). Durante este proceso se instalan los binarios necesarios, los ficheros de configuración y el servicio asociado al servidor web.



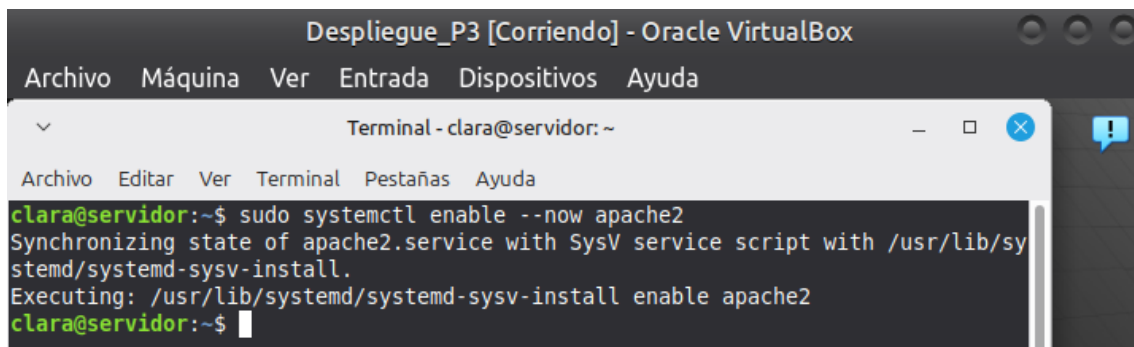
```
Terminal - clara@servidor: ~
Archivo Editar Ver Terminal Pestañas Ayuda
clara@servidor:~$ sudo apt install -y apache2
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  apache2-bin apache2-data apache2-utils libapr1t64 libaprutil1-dbd-sqlite3
  libaprutil1-ldap libaprutil1t64
Paquetes sugeridos:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom
Se instalarán los siguientes paquetes NUEVOS:
  apache2 apache2-bin apache2-data apache2-utils libapr1t64
  libaprutil1-dbd-sqlite3 libaprutil1-ldap libaprutil1t64
0 actualizados, 8 nuevos se instalarán, 0 para eliminar y 253 no actualizados.
Se necesita descargar 1.902 kB de archivos.
Se utilizarán 7.451 kB de espacio de disco después de esta operación.
Des:1 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 libapr1t64 amd64
  1.7.2-3.1ubuntu0.1 [108 kB]
Des:2 http://archive.ubuntu.com/ubuntu noble/main amd64 libaprutil1t64 amd64 1.6
  .3-1.1ubuntu7 [91,9 kB]
Des:3 http://archive.ubuntu.com/ubuntu noble/main amd64 libaprutil1-dbd-sqlite3
  amd64 1.6.3-1.1ubuntu7 [11,2 kB]
Des:4 http://archive.ubuntu.com/ubuntu noble/main amd64 libaprutil1-ldap amd64 1
  .6.3-1.1ubuntu7 [9.116 B]
Des:5 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 apache2-bin amd64
```

1.2 Habilitar e iniciar Apache

Una vez que comprobemos que tenemos instalado Apache vamos a habilitarlo e iniciarlo.

```
sudo systemctl enable --now apache2
```

El servicio Apache se habilita y se inicia mediante `systemctl enable --now apache2`, asegurando su ejecución inmediata y su arranque automático en futuros reinicios del sistema.



The screenshot shows a terminal window titled "Despliegue_P3 [Corriendo] - Oracle VirtualBox". Inside the terminal, the user runs the command `sudo systemctl enable --now apache2`. The output shows the system synchronizing the state of the `apache2.service` with the SysV service script and then executing the command to enable the service. The prompt returns to `clara@servidor:~$`.

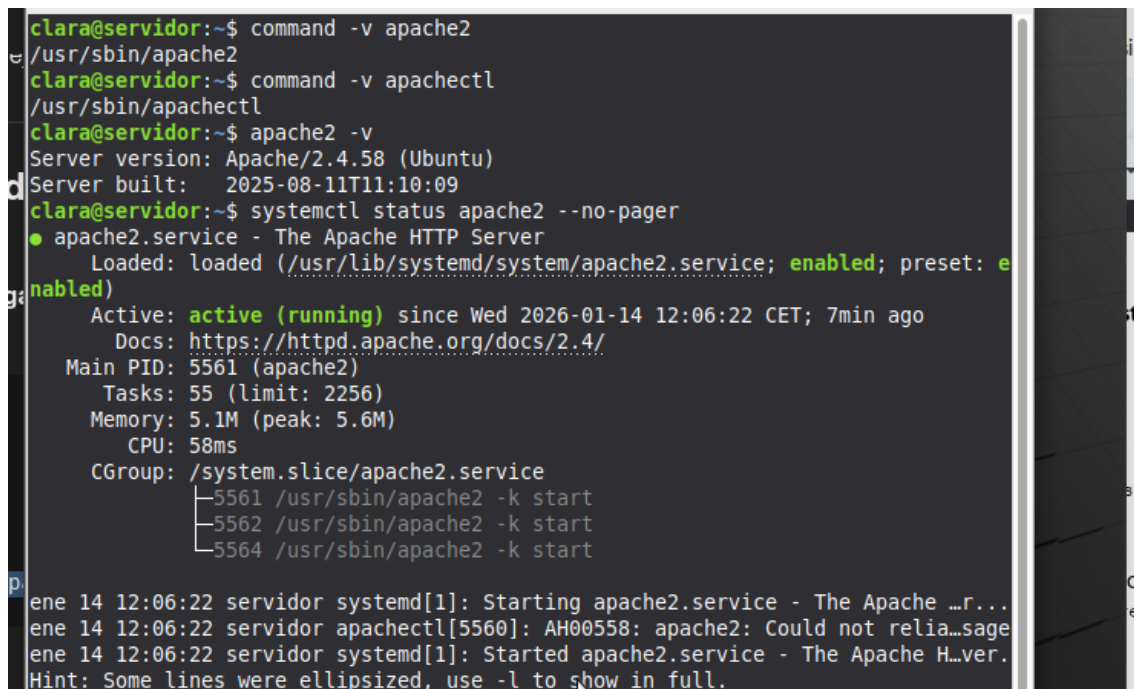
1.3 Verificación de la instalación

```
command -v apache2
```

```
command -v apachectl
```

```
apache2 -v
```

```
systemctl status apache2 --no-pager
```



The screenshot shows a terminal window with the following commands and output:

```
clara@servidor:~$ command -v apache2
/usr/sbin/apache2
clara@servidor:~$ command -v apachectl
/usr/sbin/apachectl
clara@servidor:~$ apache2 -v
Server version: Apache/2.4.58 (Ubuntu)
Server built:   2025-08-11T11:10:09
clara@servidor:~$ systemctl status apache2 --no-pager
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
   Active: active (running) since Wed 2026-01-14 12:06:22 CET; 7min ago
     Docs: https://httpd.apache.org/docs/2.4/
    Main PID: 5561 (apache2)
      Tasks: 55 (limit: 2256)
     Memory: 5.1M (peak: 5.6M)
        CPU: 58ms
    CGroup: /system.slice/apache2.service
            └─5561 /usr/sbin/apache2 -k start
              5562 /usr/sbin/apache2 -k start
              5564 /usr/sbin/apache2 -k start

ene 14 12:06:22 servidor systemd[1]: Starting apache2.service - The Apache ...r...
ene 14 12:06:22 servidor apachectl[5560]: AH00558: apache2: Could not relia...sage
ene 14 12:06:22 servidor systemd[1]: Started apache2.service - The Apache H...ver.
Hint: Some lines were ellipsized, use -l to show in full.
```

Con esto comprobamos que Apache existe en el sistema, la versión que tiene y que el servicio está activo (`active (running)`) .

Se verifica la correcta instalación de Apache mediante la localización de sus binarios (`command -v apache2`), la consulta de la versión instalada (`apache2 -v`) y la comprobación del estado del servicio con `systemctl status apache2`, confirmando que el servidor se encuentra activo y en funcionamiento.

FASE 2: CONFIGURACIÓN DE PUERTOS Y ARQUITECTURA APACHE

A continuación, vamos a preparar Apache para alojar **tres servidores web diferenciados en la misma máquina**, utilizando **puertos distintos y niveles de seguridad diferentes**.

2.1 Copia de seguridad del fichero de puerto

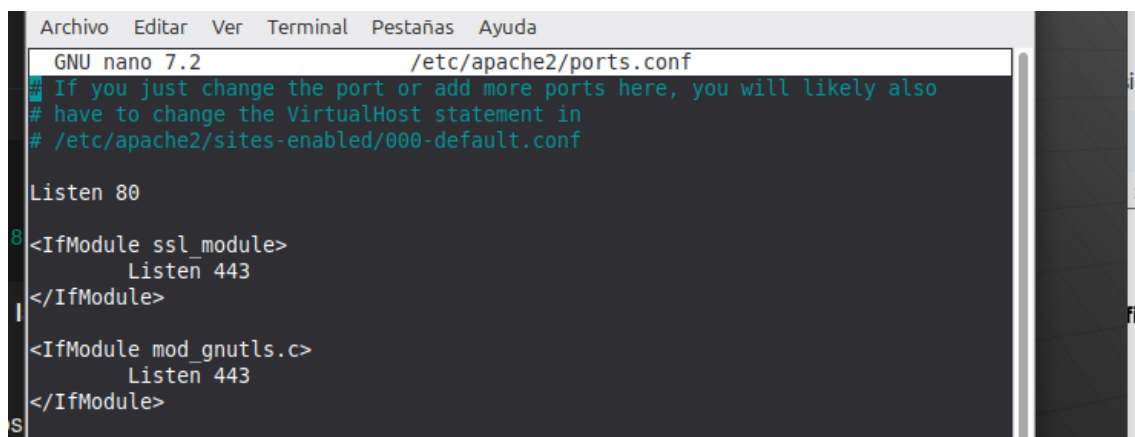
```
sudo cp /etc/apache2/ports.conf /etc/apache2/ports.conf.bak
```

Antes de modificar la configuración del servidor, realiza una copia de seguridad del fichero `/etc/apache2/ports.conf`, permitiendo restaurar la configuración original en caso de error.

2.2 Copia de seguridad del fichero de puerto

```
sudo cp /etc/apache2/ports.conf /etc/apache2/ports.conf.bak
```

Comprueba que en el fichero aparezcan estas líneas (no importa el orden, pero debe de existir, si falta alguna añádela al fichero)



```
Archivo  Editar  Ver  Terminal  Pestañas  Ayuda
GNU nano 7.2 /etc/apache2/ports.conf
# If you just change the port or add more ports here, you will likely also
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf

Listen 80

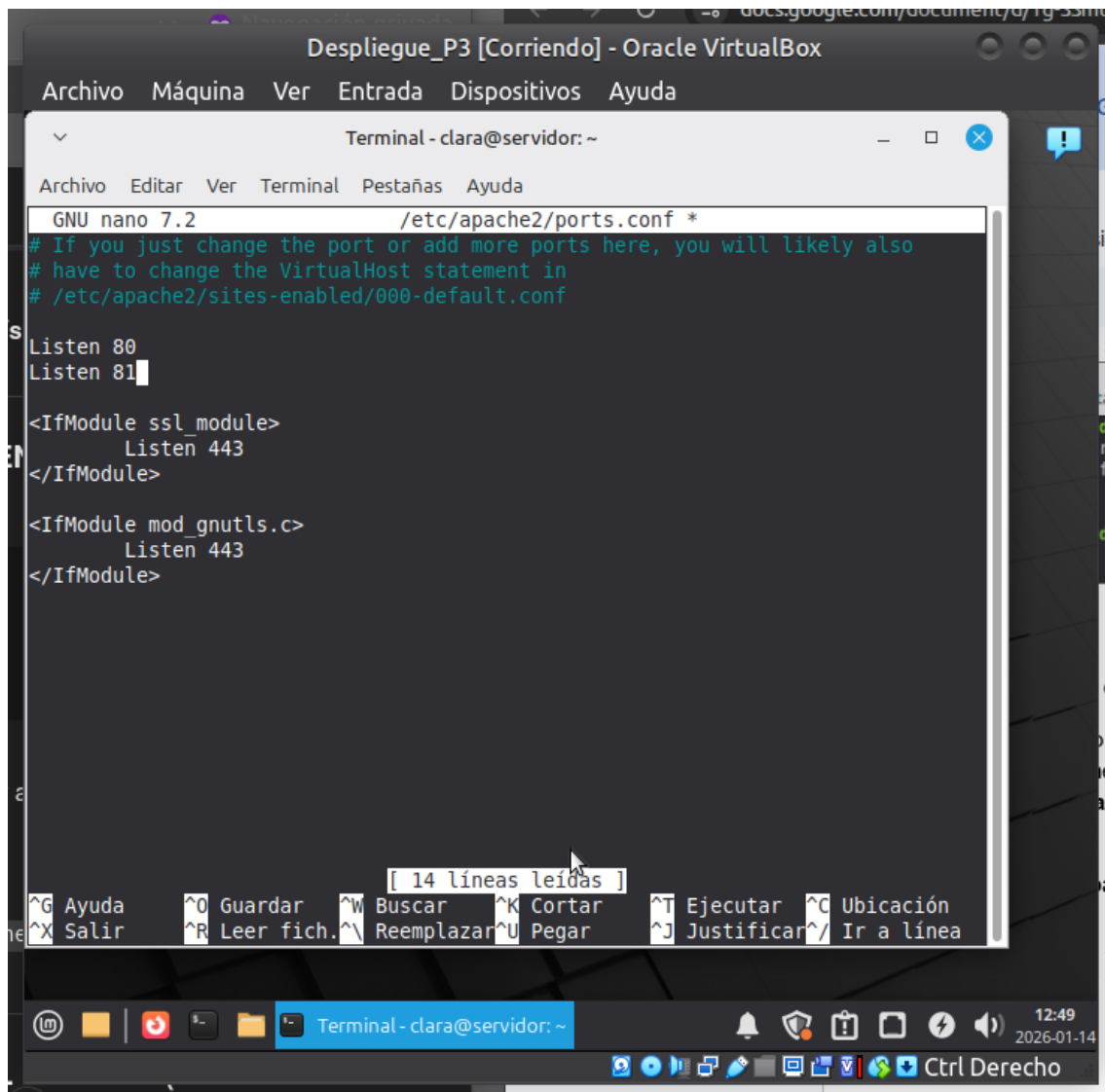
<IfModule ssl_module>
    Listen 443
</IfModule>

<IfModule mod_gnutls.c>
    Listen 443
</IfModule>
```

Ahora mismo nuestro archivo nos dice:

- Apache escucha en **80**
- Apache escuchará en **443** cuando SSL esté activo
- **NO escucha en el puerto 81**, que es obligatorio para la Web nº2

Tenemos que añadir por los tanto `Listen 81` justo debajo de `Listen 80`



```
Despliegue_P3 [Corriendo] - Oracle VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda

Terminal - clara@servidor: ~
Archivo  Editar  Ver  Terminal  Pestañas  Ayuda

GNU nano 7.2 /etc/apache2/ports.conf *
# If you just change the port or add more ports here, you will likely also
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf

Listen 80
Listen 81

<IfModule ssl_module>
    Listen 443
</IfModule>

<IfModule mod_gnutls.c>
    Listen 443
</IfModule>
```

Guardar y salir (nano)

Ctrl + O

Enter

Ctrl + X

2.3 Desactivar el sitio por defecto de Apache

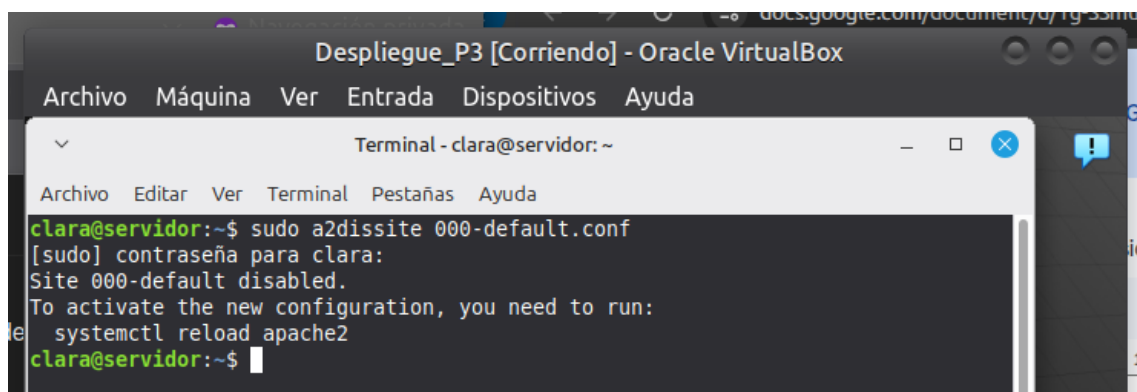
Apache incluye un VirtualHost por defecto que **no debe usarse** en esta práctica.

Se deshabilita el sitio virtual por defecto (`000-default.conf`) incluido en la instalación inicial de Apache, ya que este se encuentra configurado para escuchar en el puerto 80.

Dado que el servidor web nº 1 de la práctica también utiliza el puerto 80, mantener ambos sitios activos podría provocar conflictos o servir contenido no deseado.

De este modo, se garantiza que el tráfico HTTP en el puerto 80 sea atendido exclusivamente por el servidor web definido específicamente para esta práctica.

```
sudo a2dissite 000-default.conf
```



```
Despliegue_P3 [Corriendo] - Oracle VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda

Terminal - clara@servidor: ~
Archivo  Editar  Ver  Terminal  Pestañas  Ayuda

clara@servidor:~$ sudo a2dissite 000-default.conf
[sudo] contraseña para clara:
Site 000-default disabled.
To activate the new configuration, you need to run:
# systemctl reload apache2
clara@servidor:~$
```

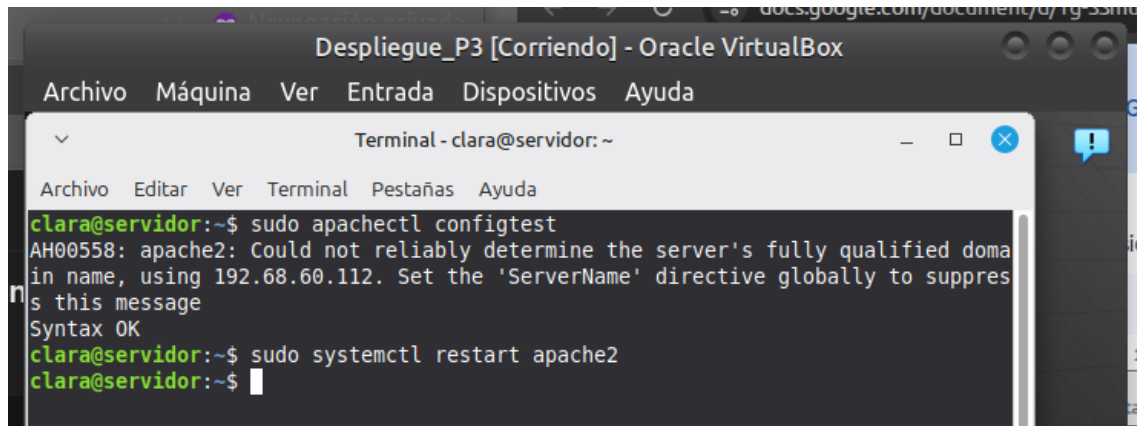
Esto significa que:

- El sitio por defecto **ha sido desactivado**
- Apache **ya no usará** `000-default.conf`
- Falta **aplicar el cambio** (reload o restart)

2.4 Comprobar sintaxis y reiniciar Apache

```
sudo apachectl configtest
```

```
sudo systemctl restart apache2
```



```
Despliegue_P3 [Corriendo] - Oracle VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda

Terminal - clara@servidor: ~
Archivo  Editar  Ver  Terminal  Pestañas  Ayuda

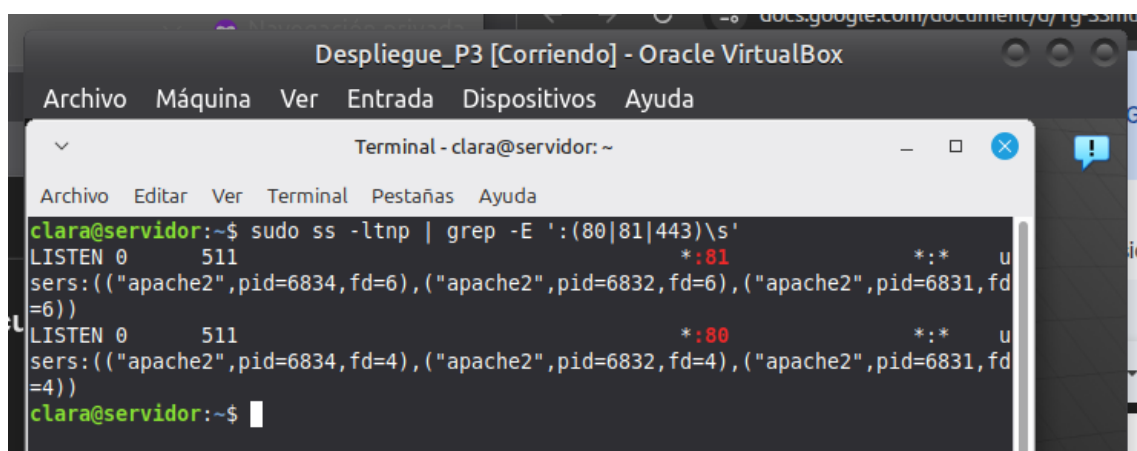
clara@servidor:~$ sudo apachectl configtest
AH00558: apache2: Could not reliably determine the server's fully qualified domain
name, using 192.68.60.112. Set the 'ServerName' directive globally to suppress t
his message
Syntax OK
clara@servidor:~$ sudo systemctl restart apache2
clara@servidor:~$
```

Se valida la sintaxis de los ficheros de configuración de Apache mediante `apachectl configtest`, obteniendo como resultado **Syntax OK**. Posteriormente, se reinicia el servicio Apache para aplicar los cambios realizados.

2.5 Verificar que Apache escucha en los puertos requeridos

```
sudo ss -ltnp | grep -E ':(80|81|443)\s'
```

Se verifica que el servidor Apache se encuentra escuchando correctamente en los puertos 80, 81 y 443, confirmando la correcta configuración de la arquitectura web basada en múltiples servidores virtuales por puerto.



```
Despliegue_P3 [Corriendo] - Oracle VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda

Terminal - clara@servidor: ~
Archivo  Editar  Ver  Terminal  Pestañas  Ayuda

clara@servidor:~$ sudo ss -ltnp | grep -E ':(80|81|443)\s'
LISTEN 0      511          *:81          *:~          u
sers:(("apache2",pid=6834,fd=6),("apache2",pid=6832,fd=6),("apache2",pid=6831,fd
=6))
LISTEN 0      511          *:80          *:~          u
sers:(("apache2",pid=6834,fd=4),("apache2",pid=6832,fd=4),("apache2",pid=6831,fd
=4))
clara@servidor:~$
```

Se verifica mediante el comando `ss -ltnp` que el servidor Apache se encuentra escuchando correctamente en los puertos 80 y 81, correspondientes a los servidores web no seguros definidos en la práctica.

En este punto, el puerto 443 aún no se encuentra en escucha, ya que la configuración del servidor web seguro (HTTPS) y la activación del módulo SSL se realizarán en fases posteriores.

FASE 3: ESTRUCTURA DE DIRECTORIOS Y CONTENIDOS

En este punto vamos a crear la estructura de directorios para cada uno de los tres servidores web.

3.1 Creación de la estructura

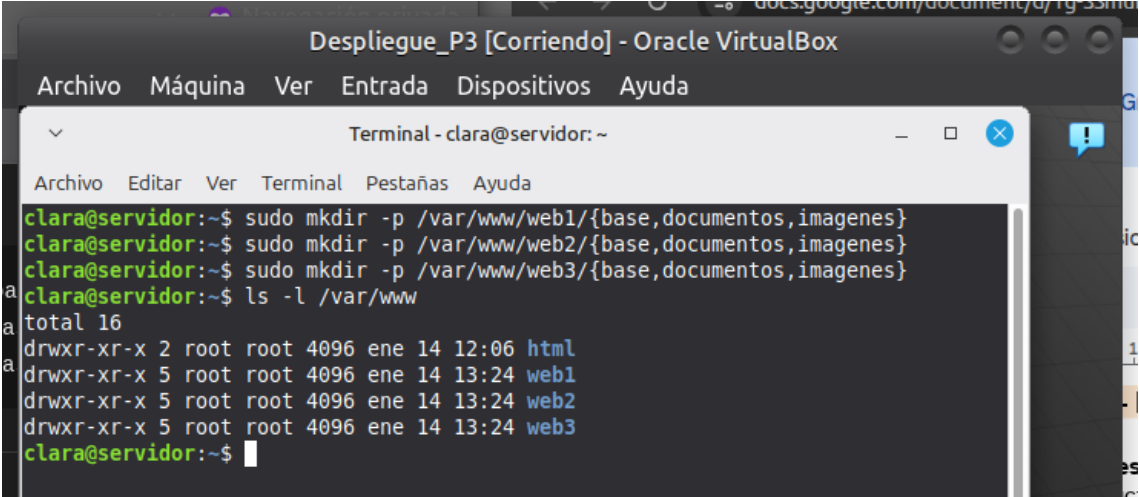
```
sudo mkdir -p /var/www/web1/{base,documentos,imagenes}
```

```
sudo mkdir -p /var/www/web2/{base,documentos,imagenes}
```

```
sudo mkdir -p /var/www/web3/{base,documentos,imagenes}
```

Comprobamos que se han creado los directorios correctamente:

```
ls -l /var/www
```



The screenshot shows a terminal window titled "Despliegue_P3 [Corriendo] - Oracle VirtualBox". Inside the terminal, the user 'clara' has executed three 'mkdir' commands to create the directory structure for three web servers. The final command executed is 'ls -l /var/www', which displays the following output:

```
total 16
drwxr-xr-x 2 root root 4096 ene 14 12:06 html
drwxr-xr-x 5 root root 4096 ene 14 13:24 web1
drwxr-xr-x 5 root root 4096 ene 14 13:24 web2
drwxr-xr-x 5 root root 4096 ene 14 13:24 web3
```

3.2 Asignación de propietario y permisos

Debemos configurar los permisos y propietarios con los siguientes comandos:

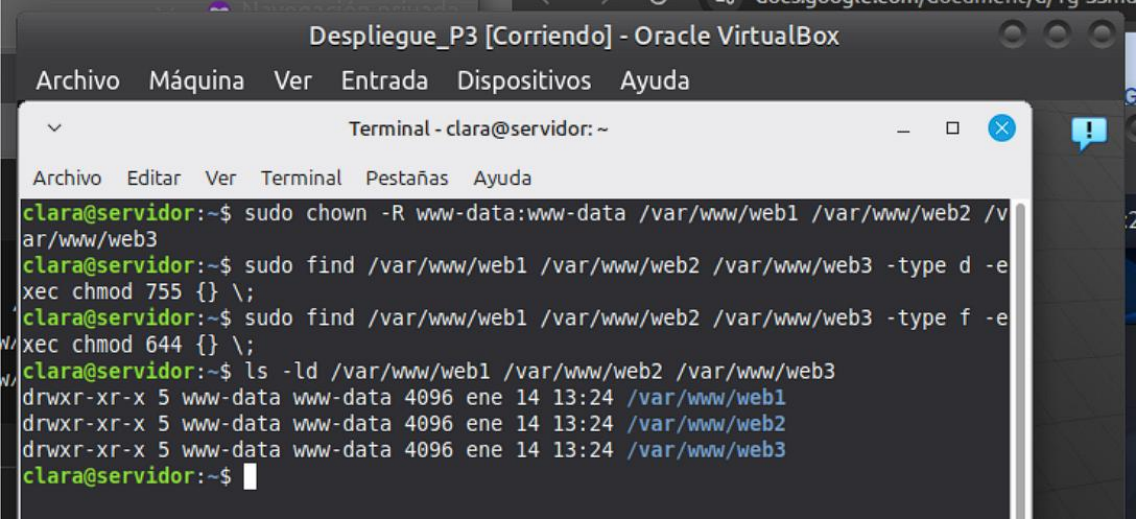
```
sudo chown -R www-data:www-data /var/www/web1 /var/www/web2  
/var/www/web3
```

```
sudo find /var/www/web1 /var/www/web2 /var/www/web3 -type d -exec  
chmod 755 {} \;
```

```
sudo find /var/www/web1 /var/www/web2 /var/www/web3 -type f -exec  
chmod 644 {} \;
```

Y comprobamos que se han dado bien los permisos con:

```
ls -ld /var/www/web1 /var/www/web2 /var/www/web3
```



The screenshot shows a terminal window titled "Despliegue_P3 [Corriendo] - Oracle VirtualBox". The terminal output shows the following commands and their results:

```
clara@servidor:~$ sudo chown -R www-data:www-data /var/www/web1 /var/www/web2 /var/www/web3  
clara@servidor:~$ sudo find /var/www/web1 /var/www/web2 /var/www/web3 -type d -exec chmod 755 {} \;  
clara@servidor:~$ sudo find /var/www/web1 /var/www/web2 /var/www/web3 -type f -exec chmod 644 {} \;  
clara@servidor:~$ ls -ld /var/www/web1 /var/www/web2 /var/www/web3  
drwxr-xr-x 5 www-data www-data 4096 ene 14 13:24 /var/www/web1  
drwxr-xr-x 5 www-data www-data 4096 ene 14 13:24 /var/www/web2  
drwxr-xr-x 5 www-data www-data 4096 ene 14 13:24 /var/www/web3  
clara@servidor:~$
```

Se comprueba que los directorios raíz de los servidores web (`/var/www/web1`, `/var/www/web2` y `/var/www/web3`) existen y tienen asignado como propietario y grupo al usuario `www-data`.

Los permisos configurados (**755** para directorios y **644** para ficheros) **garantizan el acceso de lectura al contenido web por parte del servidor Apache**, cumpliendo las buenas prácticas de **seguridad**.

3.3 Creación del contenido mínimo en documentos e imágenes

Aunque los ficheros HTML se crean y documentan de forma individual en las fases correspondientes a cada servidor web, en esta fase se completa la estructura mínima exigida mediante la creación de ficheros **.doc** y **.jpg** en los directorios **documentos** e **imagenes** de cada web.

Creación de ficheros .doc

Se generan cinco documentos .doc en el directorio documentos de cada servidor web mediante los siguientes comandos:

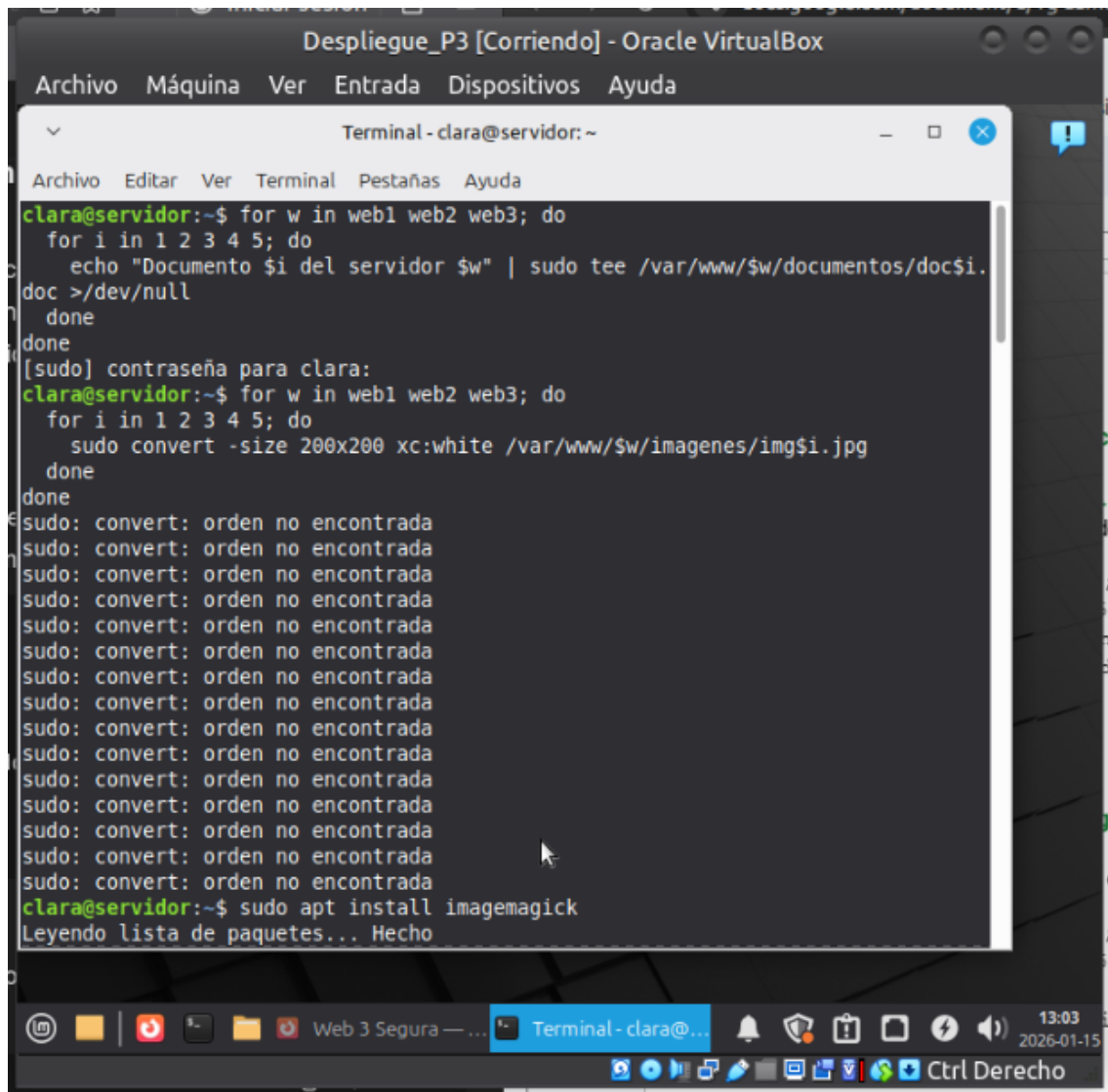
```
for w in web1 web2 web3; do
    for i in 1 2 3 4 5; do
        echo "Documento $i del servidor $w" | sudo tee
        /var/www/$w/documentos/doc$i.doc >/dev/null
    done
done
```

Creación de ficheros .jpg

Se crean cinco imágenes .jpg de ejemplo en el directorio imagenes de cada servidor web:

```
for w in web1 web2 web3; do
    for i in 1 2 3 4 5; do
        sudo convert -size 200x200 xc:black
        /var/www/$w/imagenes/img$i.jpg
    done
done
```

(En caso de no disponer de ImageMagick, se instala previamente mediante `sudo apt install imagemagick`)



3.4 Comprobación del contenido creado

Se verifica que cada servidor web dispone del contenido mínimo exigido:

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

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

```
ls /var/www/web3/documentos /var/www/web3/imagenes
```

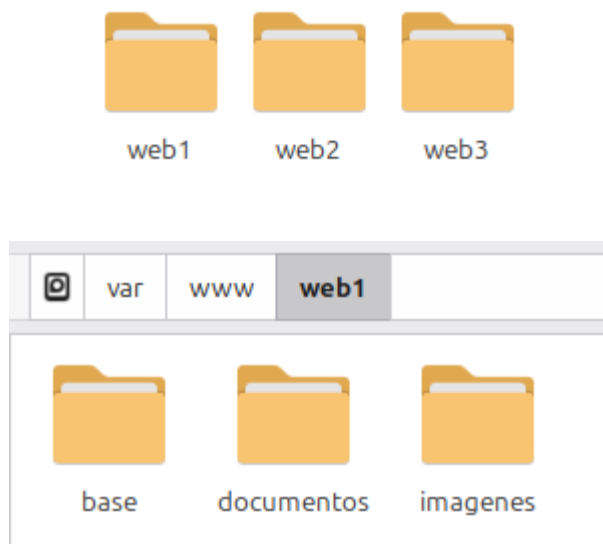
```
Despliegue_P3 [Corriendo] - Oracle VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda

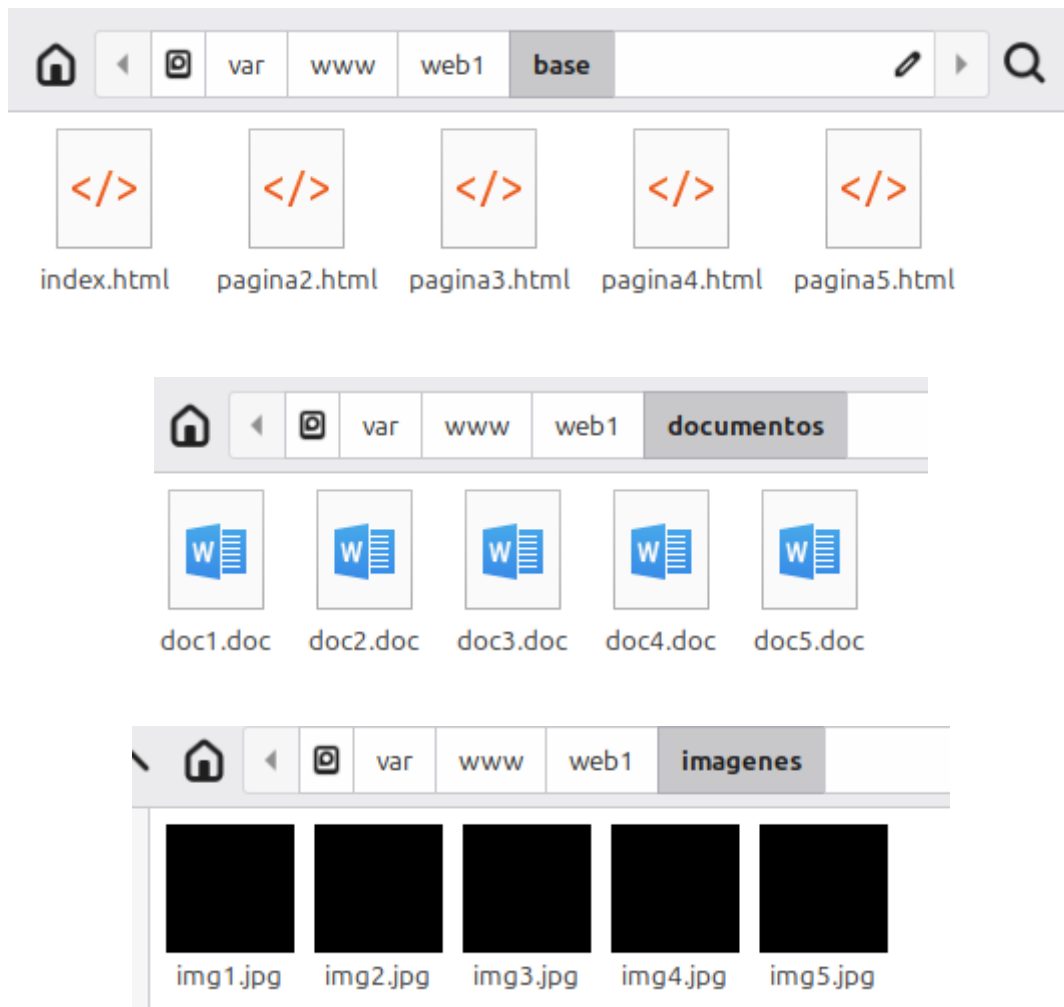
Terminal - clara@servidor: ~
Archivo  Editar  Ver  Terminal  Pestañas  Ayuda

clara@servidor:~$ ls /var/www/web1/documentos /var/www/web1/imagenes
/var/www/web1/documentos:
doc1.doc doc2.doc doc3.doc doc4.doc doc5.doc
/var/www/web1/imagenes:
img1.jpg img2.jpg img3.jpg img4.jpg img5.jpg
clara@servidor:~$ ls /var/www/web2/documentos /var/www/web2/imagenes
/var/www/web2/documentos:
doc1.doc doc2.doc doc3.doc doc4.doc doc5.doc
/var/www/web2/imagenes:
img1.jpg img2.jpg img3.jpg img4.jpg img5.jpg
clara@servidor:~$ ls /var/www/web3/documentos /var/www/web3/imagenes
/var/www/web3/documentos:
doc1.doc doc2.doc doc3.doc doc4.doc doc5.doc
/var/www/web3/imagenes:
img1.jpg img2.jpg img3.jpg img4.jpg img5.jpg
```

Cada uno de los tres servidores web dispone de:

- **Cinco ficheros HTML** en el directorio **base** (creados en las fases específicas de cada web que se verán en las fases 4, 5 y 6).
- **Cinco ficheros .doc** en el directorio **documentos**.
- **Cinco ficheros .jpg** en el directorio **imagenes**.





FASE 4: SERVIDOR WEB N° 1 (PUERTO 80 - HTTP)

4.1 Crear el VirtualHost de la Web n°1

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

Contenido del fichero:

```
<VirtualHost *:80>

DocumentRoot /var/www/web1/base

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

Guardar y salir:

Ctrl + O

Enter

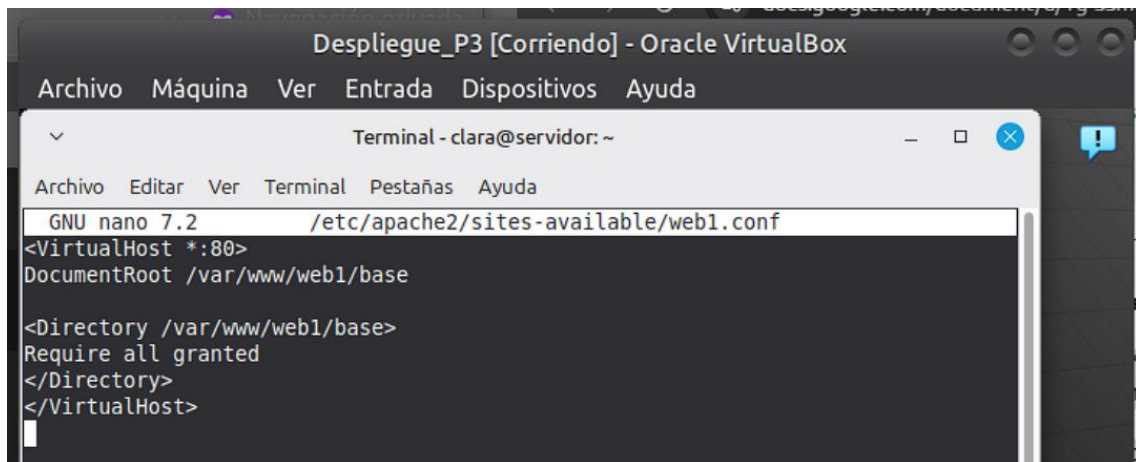
Ctrl + X

Con el contenido de este fichero definimos un servidor virtual en el puerto 80 mediante la directiva `<VirtualHost *:80>`.

El contenido web se sirve desde el directorio `/var/www/web1/base`, configurado como `DocumentRoot`.

Mediante la directiva `Require all granted` se permite el acceso al contenido desde cualquier cliente.

Se utiliza una configuración sencilla y funcional.



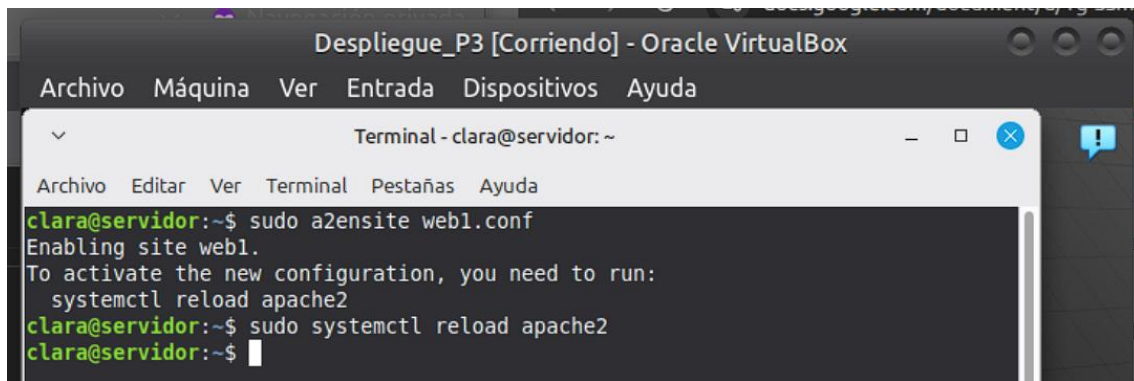
4.2 Activar el sitio web nº1

```
sudo a2ensite web1.conf
```

```
sudo systemctl reload apache2
```

El servidor web nº1 se activa mediante el comando `a2ensite web1.conf`.

Posteriormente, se recarga la configuración del servidor Apache para aplicar los cambios sin interrumpir el servicio.



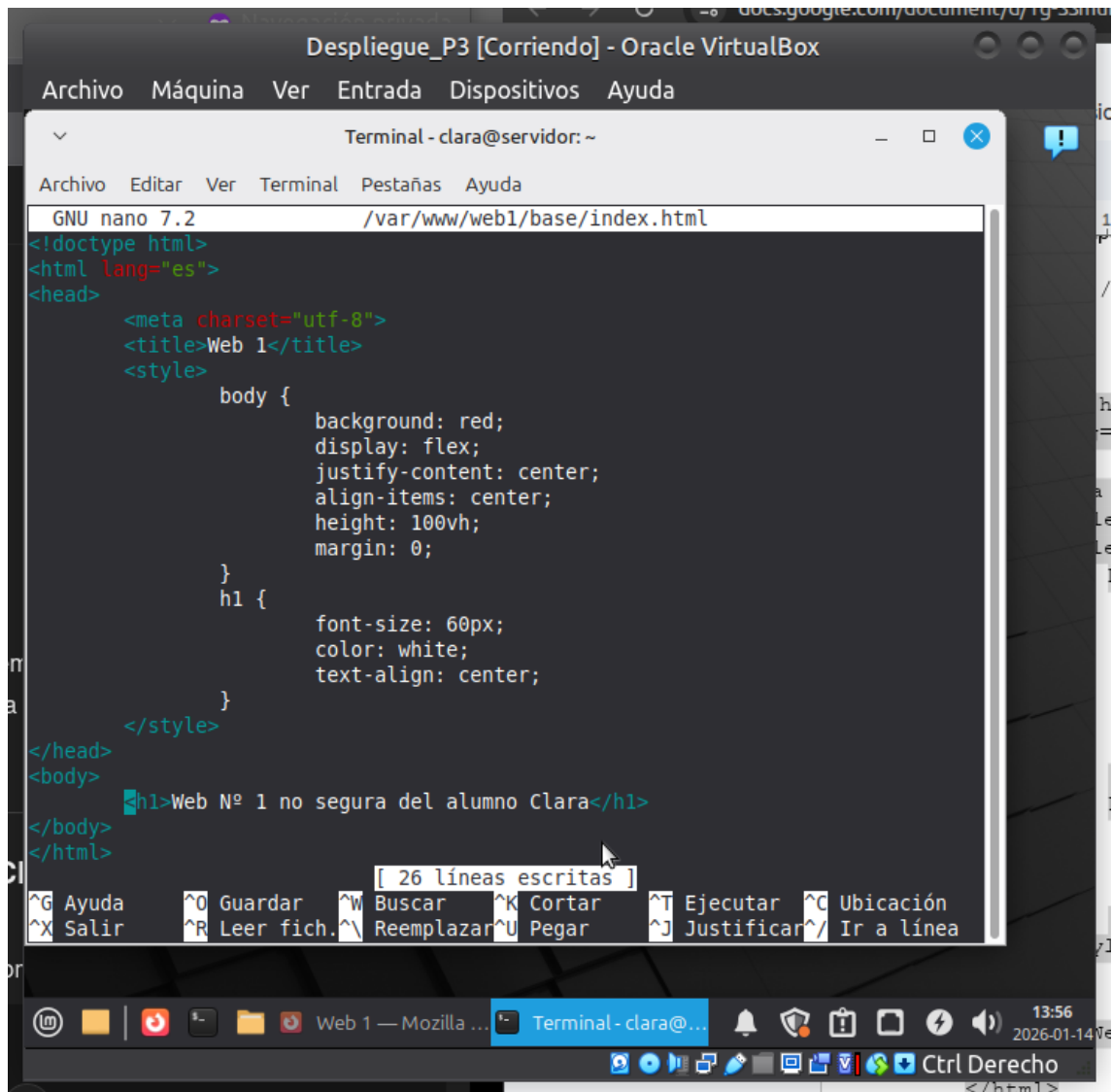
4.3 Crear el contenido HTML (fondo rojo)

Archivo principal **index.html**

```
sudo nano /var/www/web1/base/index.html
```

Contenido:

```
<!doctype html>
<html lang="es">
<head>
  <meta charset="utf-8">
  <title>Web 1</title>
  <style>
    body {
      background: red;
      display: flex;
      justify-content: center;
      align-items: center;
      height: 100vh;
      margin: 0;
    }
    h1 {
      font-size: 60px;
      color: white;
      text-align: center;
    }
  </style>
</head>
<body>
  <h1>Web N° 1 no segura del alumno Clara</h1>
</body>
</html>
```



En la terminal del sistema vamos a crear los ficheros HTML necesarios para completar el contenido mínimo del servidor web nº 1.

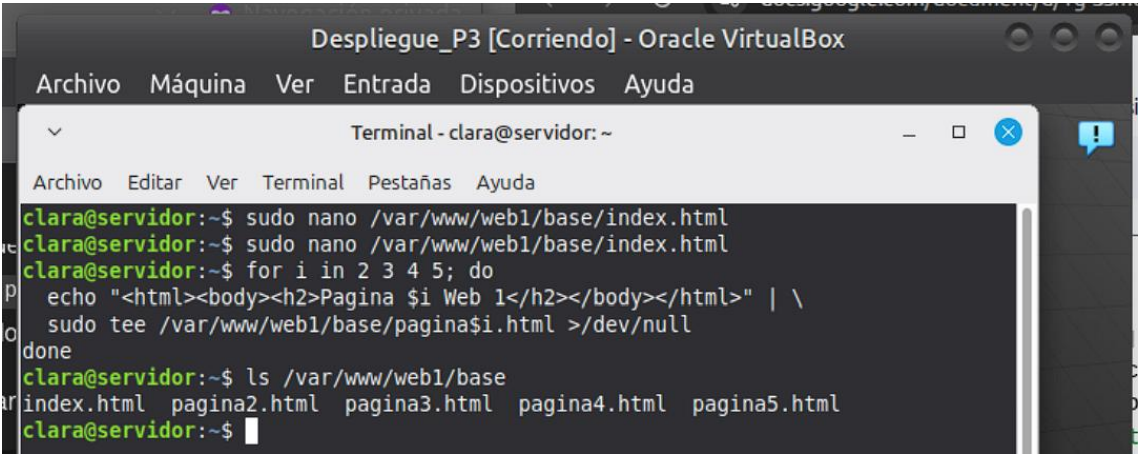
Mediante un bucle for, se generan de forma automática cuatro ficheros HTML adicionales (pagina2.html a pagina5.html) dentro del directorio /var/www/web1/base, complementando al fichero principal index.html.

Para ello, ejecuta el siguiente comando en la terminal:

```
for i in 2 3 4 5; do
    echo "<html><body><h2>Pagina $i Web 1</h2></body></html>" | \
    sudo tee /var/www/web1/base/pagina$i.html >/dev/null
done
```

Comprobamos que se han creado correctamente:

```
ls /var/www/web1/base
```



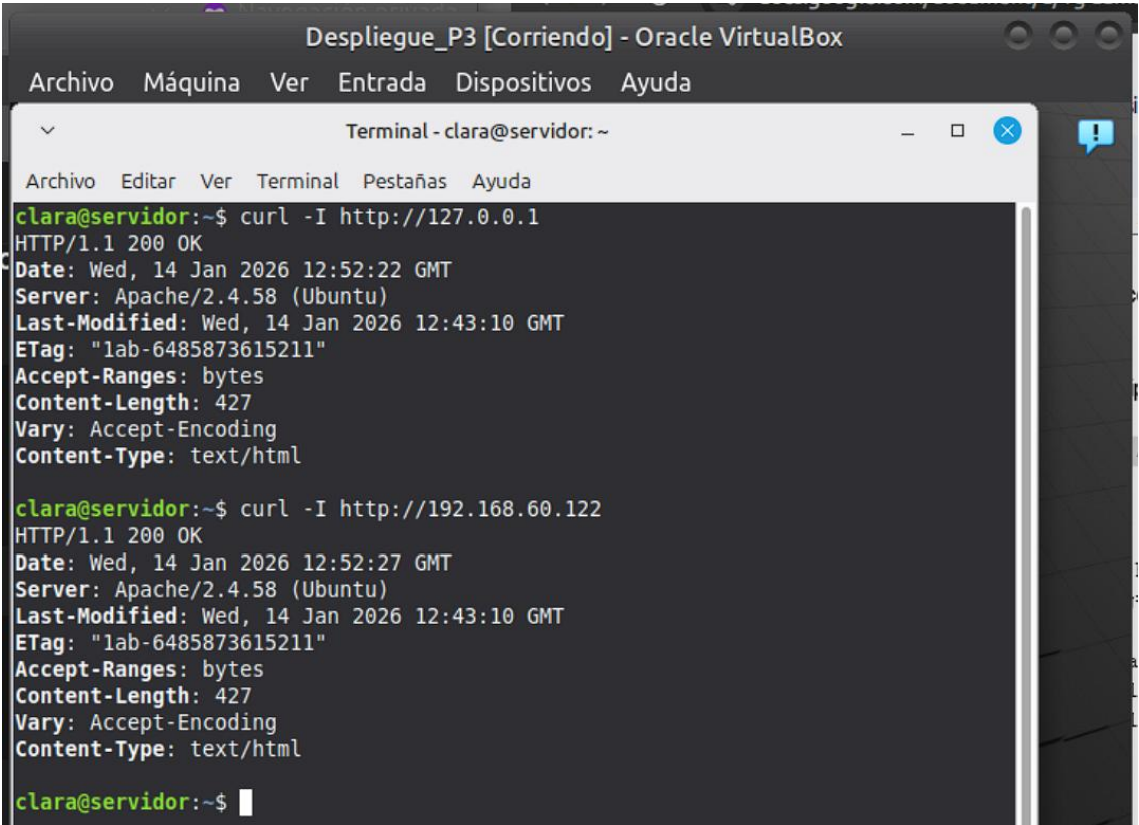
The screenshot shows a terminal window titled 'Terminal - clara@servidor: ~' within an Oracle VM VirtualBox environment. The terminal displays the following commands and output:

```
clara@servidor:~$ sudo nano /var/www/web1/base/index.html
clara@servidor:~$ sudo nano /var/www/web1/base/index.html
clara@servidor:~$ for i in 2 3 4 5; do
  echo "<html><body><h2>Pagina $i Web 1</h2></body></html>" | \
  sudo tee /var/www/web1/base/pagina$i.html >/dev/null
done
clara@servidor:~$ ls /var/www/web1/base
index.html pagina2.html pagina3.html pagina4.html pagina5.html
clara@servidor:~$
```

4.4 Comprobación del acceso a la Web nº1

```
curl -I http://127.0.0.1
```

```
curl -I http://192.168.60.122
```



The screenshot shows a terminal window titled 'Terminal - clara@servidor: ~' within an Oracle VM VirtualBox environment. The terminal displays the following commands and output:

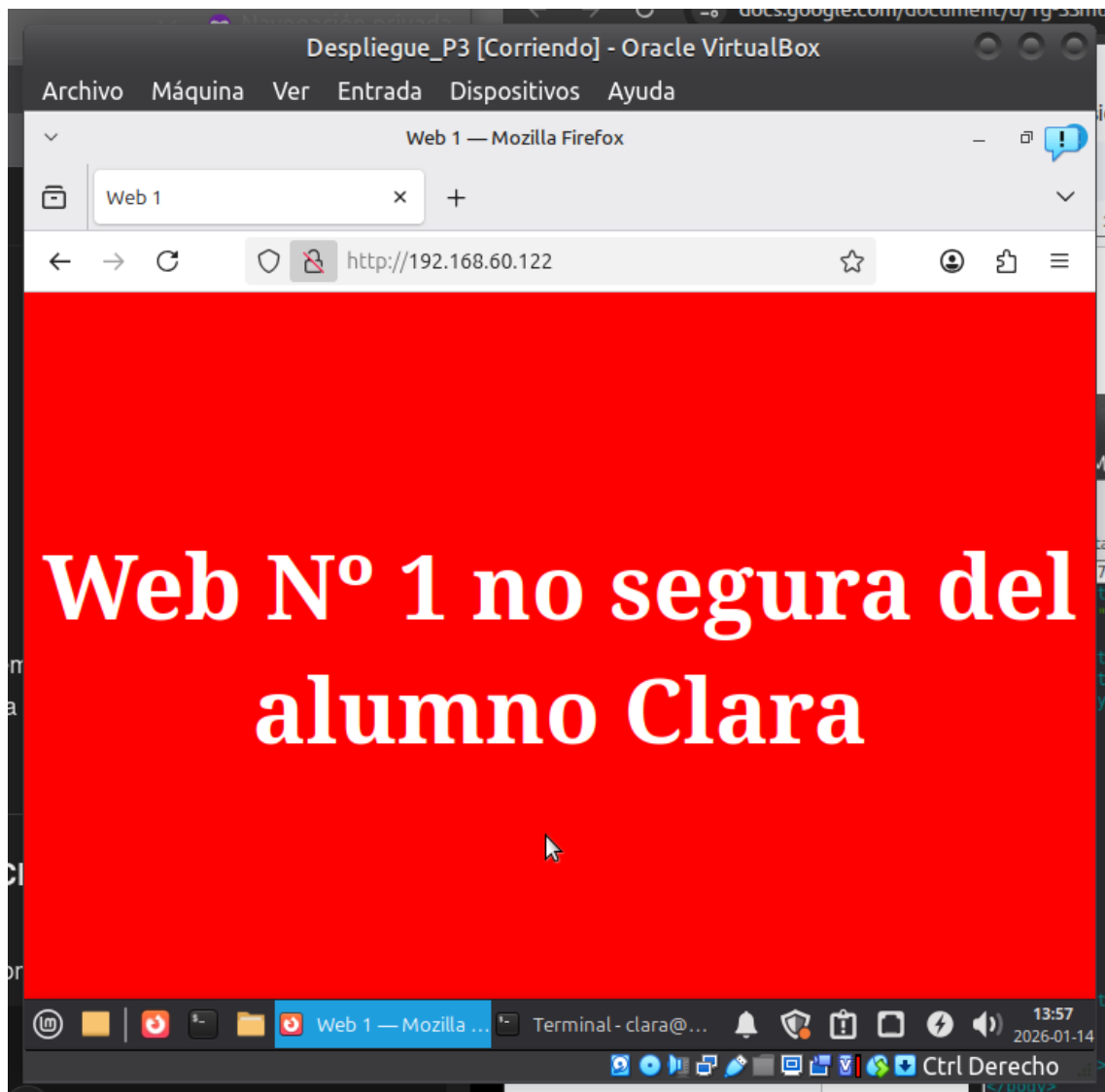
```
clara@servidor:~$ curl -I http://127.0.0.1
HTTP/1.1 200 OK
Date: Wed, 14 Jan 2026 12:52:22 GMT
Server: Apache/2.4.58 (Ubuntu)
Last-Modified: Wed, 14 Jan 2026 12:43:10 GMT
ETag: "1ab-6485873615211"
Accept-Ranges: bytes
Content-Length: 427
Vary: Accept-Encoding
Content-Type: text/html

clara@servidor:~$ curl -I http://192.168.60.122
HTTP/1.1 200 OK
Date: Wed, 14 Jan 2026 12:52:27 GMT
Server: Apache/2.4.58 (Ubuntu)
Last-Modified: Wed, 14 Jan 2026 12:43:10 GMT
ETag: "1ab-6485873615211"
Accept-Ranges: bytes
Content-Length: 427
Vary: Accept-Encoding
Content-Type: text/html

clara@servidor:~$
```

Acceso desde navegador:

<http://192.168.60.122>



Se verifica el correcto funcionamiento del servidor web nº 1 mediante el comando `curl -I`, comprobando la respuesta HTTP tanto desde la interfaz local (127.0.0.1) como desde la dirección IP asignada a la máquina virtual (192.168.60.122).

En ambos casos, el servidor responde con el código de estado **HTTP/1.1 200 OK**, lo que confirma que el servidor web no seguro del puerto 80 se encuentra operativo y accesible correctamente.

FASE 5: SERVIDOR WEB Nº 2 (PUERTO 81 - HTTP)

5.1 Crear el VirtualHost de la Web nº 2

En la terminal, crea el fichero de configuración:

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

Contenido del fichero:

```
<VirtualHost *:81>

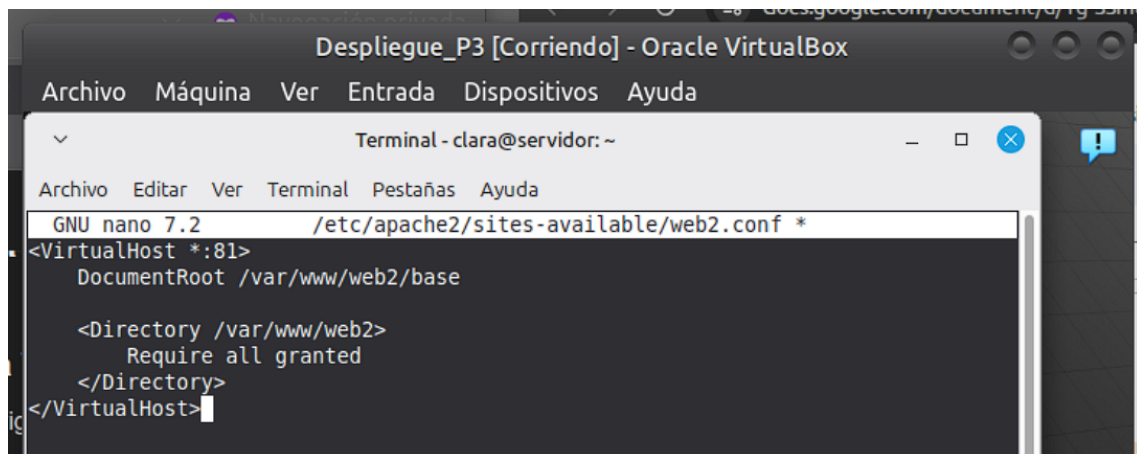
    DocumentRoot /var/www/web2/base

    <Directory /var/www/web2/base>

        Require all granted

    </Directory>

</VirtualHost>
```



Guardar y salir:

Ctrl + O

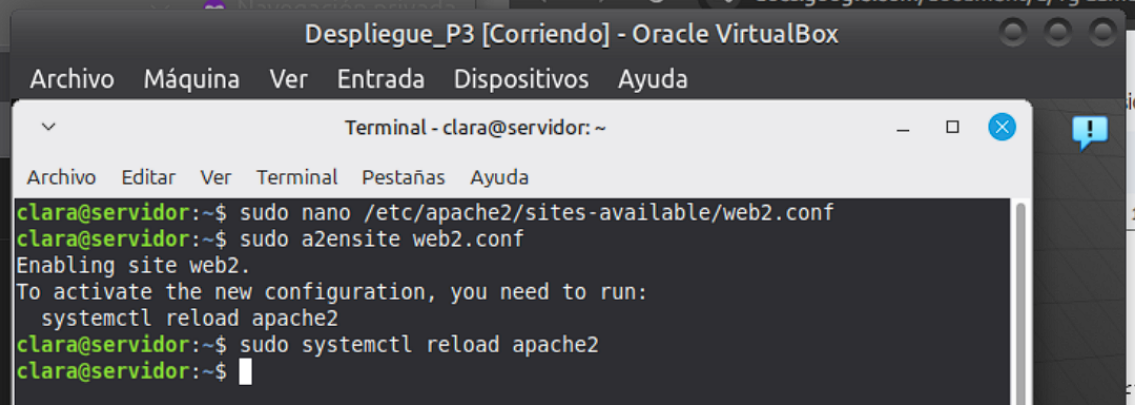
Enter

Ctrl + X

5.2 Activar el sitio web nº 2

```
sudo a2ensite web2.conf
```

```
sudo systemctl reload apache2
```



The screenshot shows a terminal window titled "Despliegue_P3 [Corriendo] - Oracle VirtualBox". The terminal output is as follows:

```
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda

Terminal - clara@servidor: ~
Archivo  Editar  Ver  Terminal  Pestañas  Ayuda

clara@servidor:~$ sudo nano /etc/apache2/sites-available/web2.conf
clara@servidor:~$ sudo a2ensite web2.conf
Enabling site web2.
To activate the new configuration, you need to run:
systemctl reload apache2
clara@servidor:~$ sudo systemctl reload apache2
clara@servidor:~$
```

Se habilita el servidor web nº 2 mediante el comando `a2ensite web2.conf` y se recarga la configuración del servidor Apache con `systemctl reload apache2`, aplicando los cambios sin interrumpir el servicio.

5.3 Crear el contenido HTML de la Web nº 2 (fondo amarillo)

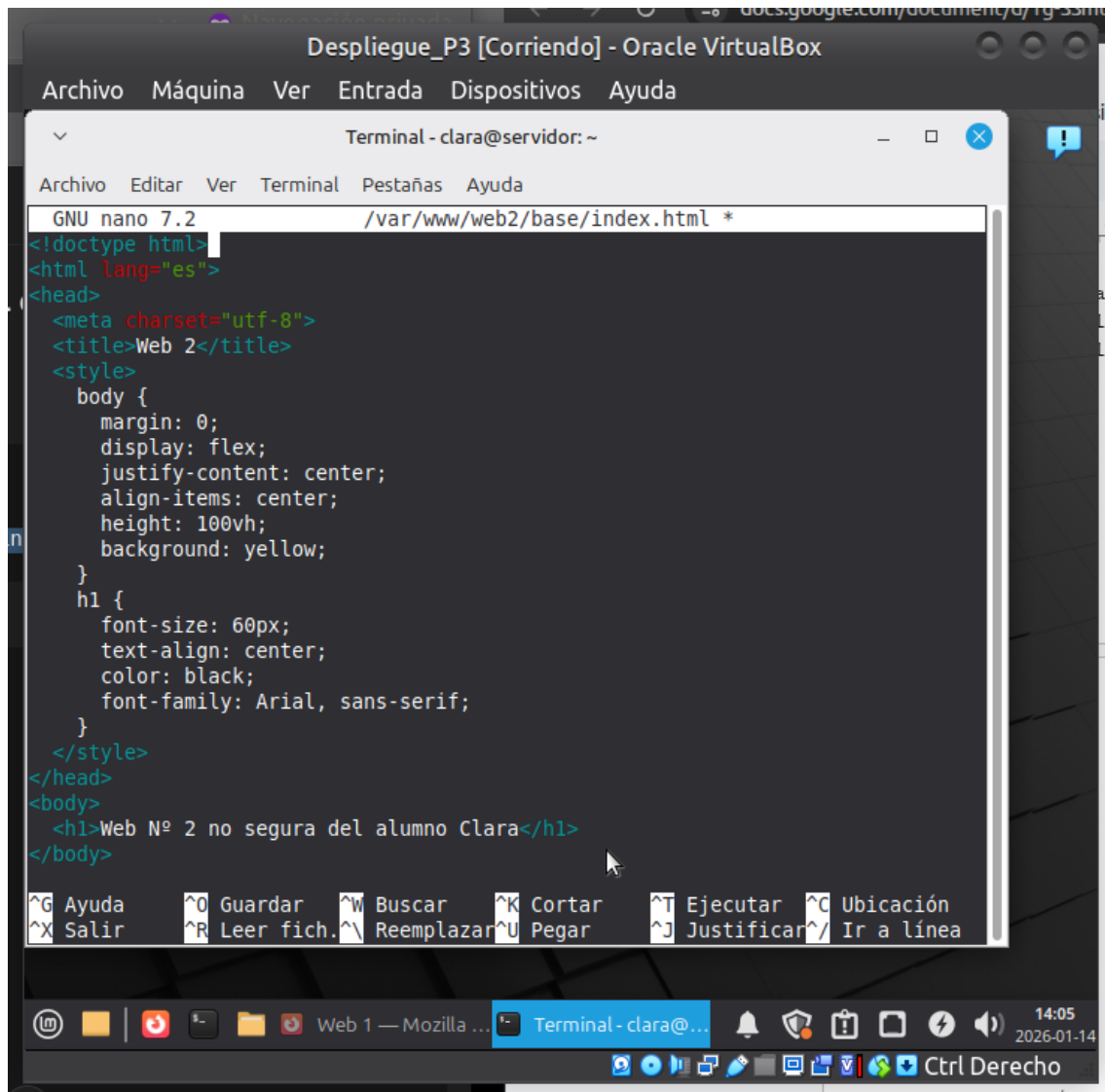
Página principal (index.html)

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

Contenido:

```
<!doctype html>
<html lang="es">
<head>
  <meta charset="utf-8">
  <title>Web 2</title>
  <style>
    body {
      margin: 0;
      display: flex;
      justify-content: center;
      align-items: center;
      height: 100vh;
      background: yellow;
    }
    h1 {
      font-size: 60px;
      text-align: center;
      color: black;
      font-family: Arial, sans-serif;
    }
  </style>
</head>
<body>
  <h1>Web N° 2 no segura del alumno Clara</h1>
</body>
</html>
```

A continuación, guarda y sal del editor.



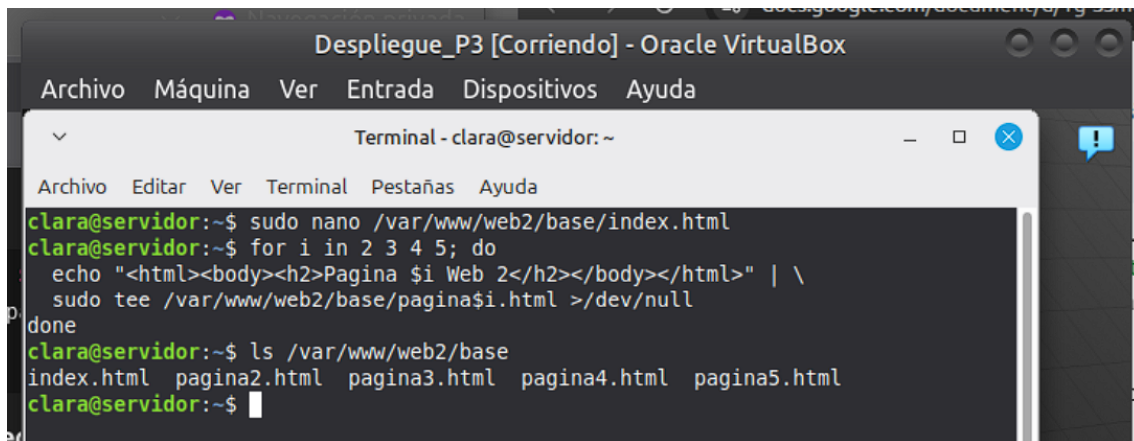
5.4 Crear los otros 4 HTML obligatorios (web 2)

Para ello, ejecuta el siguiente comando en la terminal:

```
for i in 2 3 4 5; do
    echo "<html><body><h2>Pagina $i Web 2</h2></body></html>" | \
    sudo tee /var/www/web2/base/pagina$i.html >/dev/null
done
```

Comprobamos que se han creado correctamente:

```
ls /var/www/web2/base
```



```
Despliegue_P3 [Corriendo] - Oracle VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda

Terminal - clara@servidor: ~
Archivo  Editar  Ver  Terminal  Pestañas  Ayuda

clara@servidor:~$ sudo nano /var/www/web2/base/index.html
clara@servidor:~$ for i in 2 3 4 5; do
  echo "<html><body><h2>Pagina $i Web 2</h2></body></html>" | \
  sudo tee /var/www/web2/base/pagina$i.html >/dev/null
done
clara@servidor:~$ ls /var/www/web2/base
index.html  pagina2.html  pagina3.html  pagina4.html  pagina5.html
clara@servidor:~$
```

5.5 Verificación del servidor web nº 2

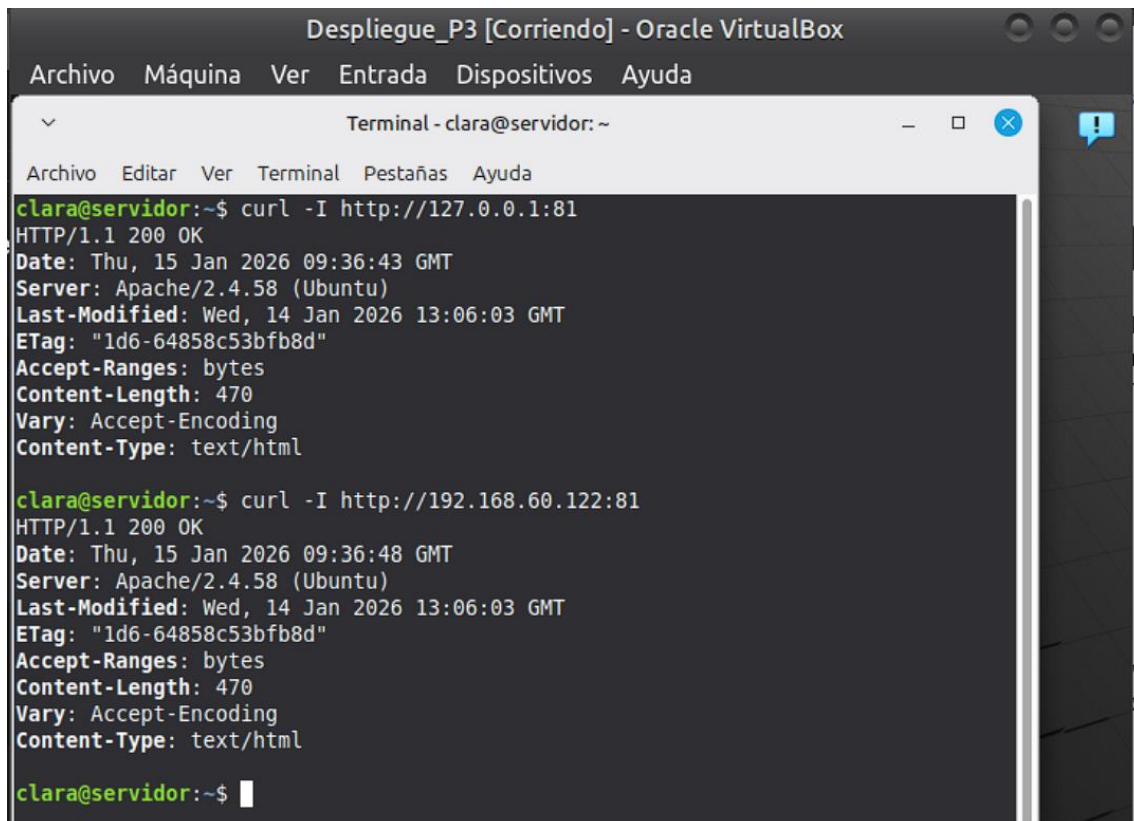
Ejecuta:

```
curl -I http://127.0.0.1:81
```

```
curl -I http://192.168.60.122:81
```

Resultado esperado

HTTP/1.1 200 OK



```
Despliegue_P3 [Corriendo] - Oracle VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda

Terminal - clara@servidor: ~
Archivo  Editar  Ver  Terminal  Pestañas  Ayuda

clara@servidor:~$ curl -I http://127.0.0.1:81
HTTP/1.1 200 OK
Date: Thu, 15 Jan 2026 09:36:43 GMT
Server: Apache/2.4.58 (Ubuntu)
Last-Modified: Wed, 14 Jan 2026 13:06:03 GMT
ETag: "1d6-64858c53bfb8d"
Accept-Ranges: bytes
Content-Length: 470
Vary: Accept-Encoding
Content-Type: text/html

clara@servidor:~$ curl -I http://192.168.60.122:81
HTTP/1.1 200 OK
Date: Thu, 15 Jan 2026 09:36:48 GMT
Server: Apache/2.4.58 (Ubuntu)
Last-Modified: Wed, 14 Jan 2026 13:06:03 GMT
ETag: "1d6-64858c53bfb8d"
Accept-Ranges: bytes
Content-Length: 470
Vary: Accept-Encoding
Content-Type: text/html

clara@servidor:~$
```

Con esto comprobamos el correcto funcionamiento del servidor web nº 2 mediante el comando `curl -I`, verificando la respuesta **HTTP** en el **puerto 81** tanto desde la interfaz local como desde la dirección IP de la máquina virtual. La respuesta **HTTP/1.1 200 OK** confirma que el servidor web no seguro del puerto 81 se encuentra operativo.

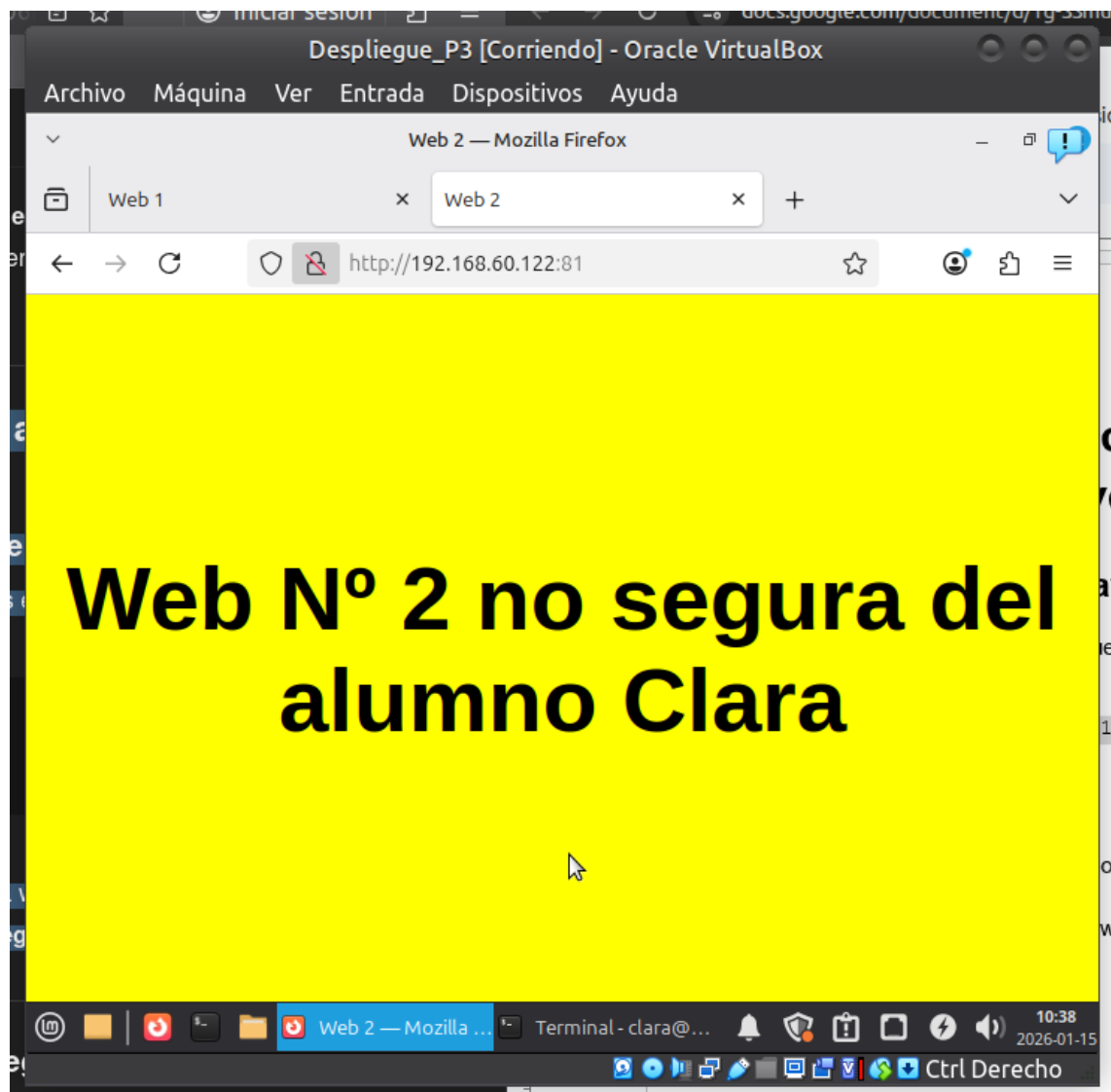
5.6 Comprobación del acceso a la Web nº 2 desde el navegador

Acceso desde el navegador web

Abre Firefox (o el navegador que tengas en la máquina virtual) y escribe en la barra de direcciones:

`http://192.168.60.122:81`

- El **:81** es obligatorio, porque esta web NO usa el puerto 80.
- No uses https, es una web NO segura.



FASE 6: SERVIDOR WEB Nº 3 (HTTPS - PUERTO 443)

En este caso el objetivo es configurar un servidor web seguro accesible mediante HTTPS en el puerto 443, utilizando un certificado autofirmado generado por el alumno, cumpliendo el enunciado de la práctica.

6.1 Activar el módulo SSL de Apache

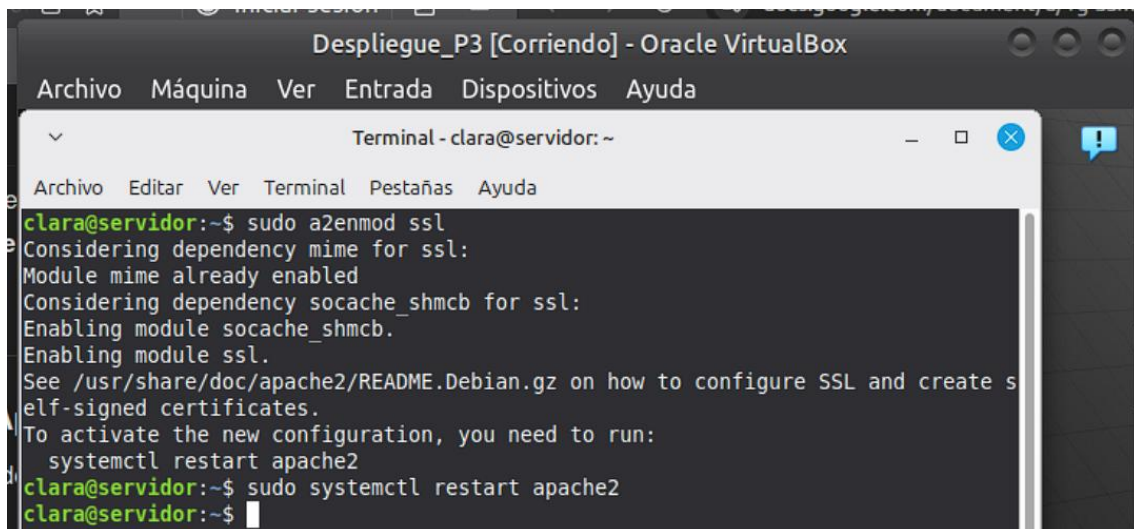
Apache necesita el módulo SSL para poder usar HTTPS.

```
sudo a2enmod ssl
```

Aplicar el cambio

```
sudo systemctl restart apache2
```

Ya hemos habilitado el módulo SSL de Apache mediante el comando `a2enmod ssl`, permitiendo al servidor web gestionar conexiones seguras mediante el protocolo HTTPS. Posteriormente, se reinicia el servicio Apache para aplicar los cambios realizados.



```
Despliegue_P3 [Corriendo] - Oracle VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda

Terminal - clara@servidor: ~
Archivo  Editar  Ver  Terminal  Pestañas  Ayuda

clara@servidor:~$ sudo a2enmod ssl
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create s
elf-signed certificates.
To activate the new configuration, you need to run:
    systemctl restart apache2
clara@servidor:~$ sudo systemctl restart apache2
clara@servidor:~$
```

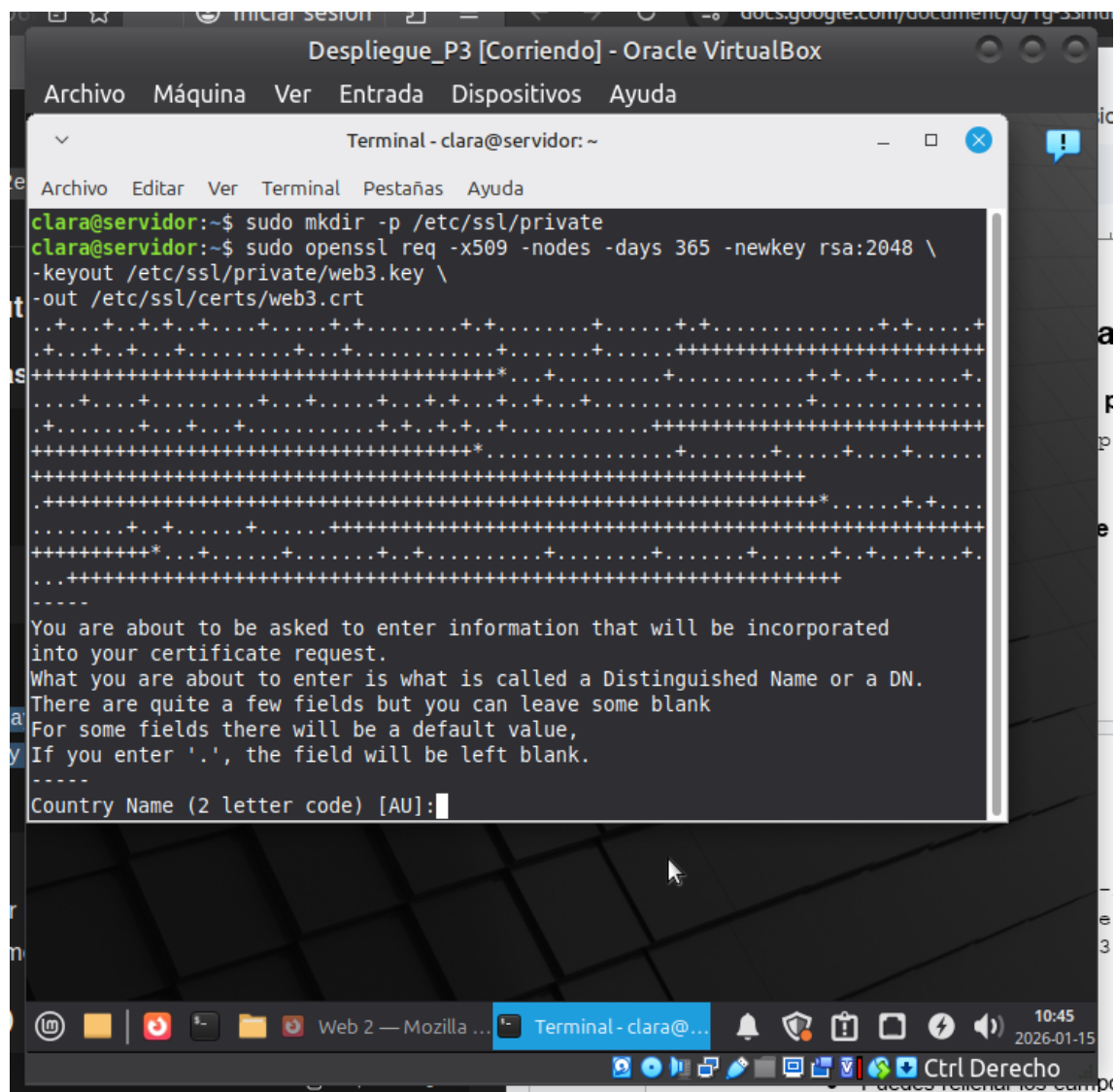
6.2 Crear el certificado SSL autofirmado

Crear directorio de claves privadas

```
sudo mkdir -p /etc/ssl/private
```

Generar certificado y clave

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 \  
-keyout /etc/ssl/private/web3.key \  
-out /etc/ssl/certs/web3.crt
```



Durante el proceso:

- Puedes rellenar los campos o pulsar **Enter** en todos.
- Es un **certificado autofirmado**

```
Country Name (2 letter code) [AU]:SP
State or Province Name (full name) [Some-State]:MADRID
Locality Name (eg, city) []:MADRID
Organization Name (eg, company) [Internet Widgits Pty Ltd]:IES SAN JUAN DE LA CR
UZ
Organizational Unit Name (eg, section) []:-
Common Name (e.g. server FQDN or YOUR name) []:CLARA
Email Address []:prueba@gmail.com
clara@servidor:~$
```

Se genera un certificado digital autofirmado mediante OpenSSL, creando una clave privada y un certificado X.509 con una validez de 365 días. Este certificado se utiliza para habilitar el acceso seguro HTTPS al servidor web nº 3. Se ajustan los permisos de los ficheros para garantizar la seguridad de la clave privada.

A continuación, debemos ajustar los permisos

```
sudo chmod 700 /etc/ssl/private

sudo chmod 600 /etc/ssl/private/web3.key

sudo chmod 644 /etc/ssl/certs/web3.crt
```

Comprobación de permisos cambiados

```
ls -ld /etc/ssl/private

ls -l /etc/ssl/private/web3.key

ls -l /etc/ssl/certs/web3.crt
```



```
Despliegue_P3 [Corriendo] - Oracle VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda

Terminal - clara@servidor: ~
Archivo  Editar  Ver  Terminal  Pestañas  Ayuda

clara@servidor:~$ sudo chmod 700 /etc/ssl/private
clara@servidor:~$ sudo chmod 600 /etc/ssl/private/web3.key
clara@servidor:~$ sudo chmod 644 /etc/ssl/certs/web3.crt
clara@servidor:~$ ls -ld /etc/ssl/private
drwx----- 2 root ssl-cert 4096 ene 15 10:45 /etc/ssl/private
clara@servidor:~$ ls -l /etc/ssl/private/web3.key
ls: no se puede acceder a '/etc/ssl/private/web3.key': Permiso denegado
clara@servidor:~$ ls -l /etc/ssl/certs/web3.crt
-rw-r--r-- 1 root root 1245 ene 15 10:46 /etc/ssl/certs/web3.crt
clara@servidor:~$
```

Con el objetivo de garantizar la seguridad del servidor web HTTPS, se configuran los permisos de los ficheros asociados al certificado digital.

El directorio `/etc/ssl/private` se establece con permisos **700**, restringiendo el acceso exclusivamente al usuario root. La clave privada del certificado (**web3.key**) se configura con permisos 600, impidiendo su acceso por parte de usuarios no autorizados.

Al intentar acceder a la clave privada desde un usuario sin privilegios, el sistema devuelve un mensaje de **“Permiso denegado”**, lo que confirma que la clave se encuentra correctamente protegida.

El certificado público (**web3.crt**) se configura con permisos **644**, permitiendo su lectura sin comprometer la seguridad del sistema.

6.3 Crear el VirtualHost HTTPS (web3)

Crear el fichero de configuración

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

Contenido

```
<VirtualHost *:443>

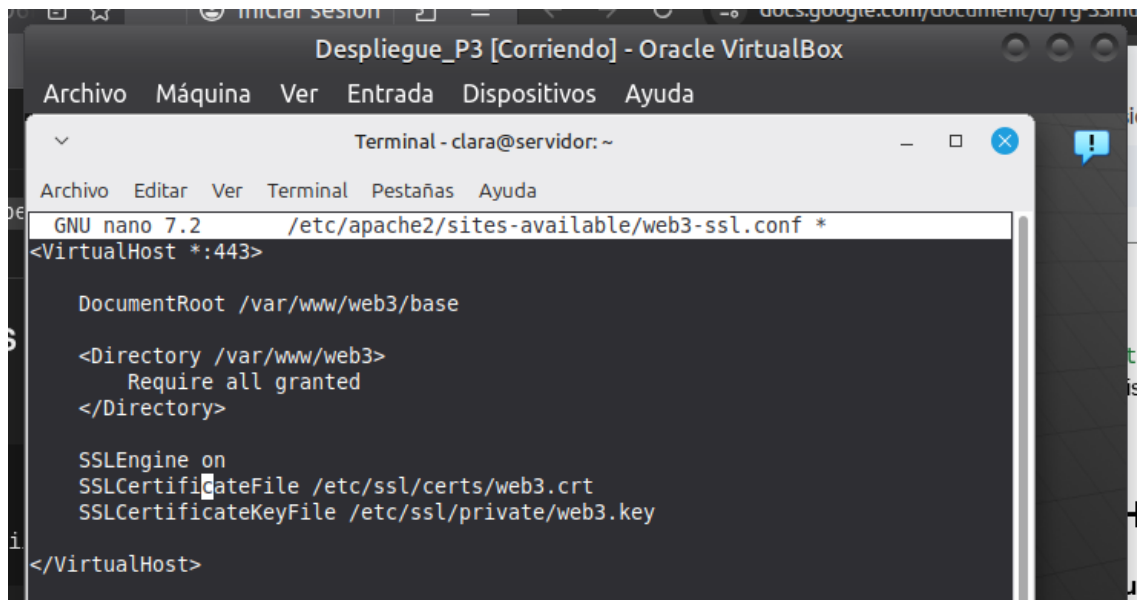
    DocumentRoot /var/www/web3/base

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

    SSLEngine on
    SSLCertificateFile /etc/ssl/certs/web3.crt
    SSLCertificateKeyFile /etc/ssl/private/web3.key

</VirtualHost>
```

A continuación, guarda y sal del editor.



Ya hemos configurado un VirtualHost en el **puerto 443** para el **servidor web nº 3**, **habilitando el uso de HTTPS mediante SSL**. Se asocia el **certificado autofirmado** generado previamente y se define el directorio raíz del sitio web seguro.

6.4 Activar el sitio web seguro

Comando

```
sudo a2ensite web3-ssl.conf
```

Aplicar cambios

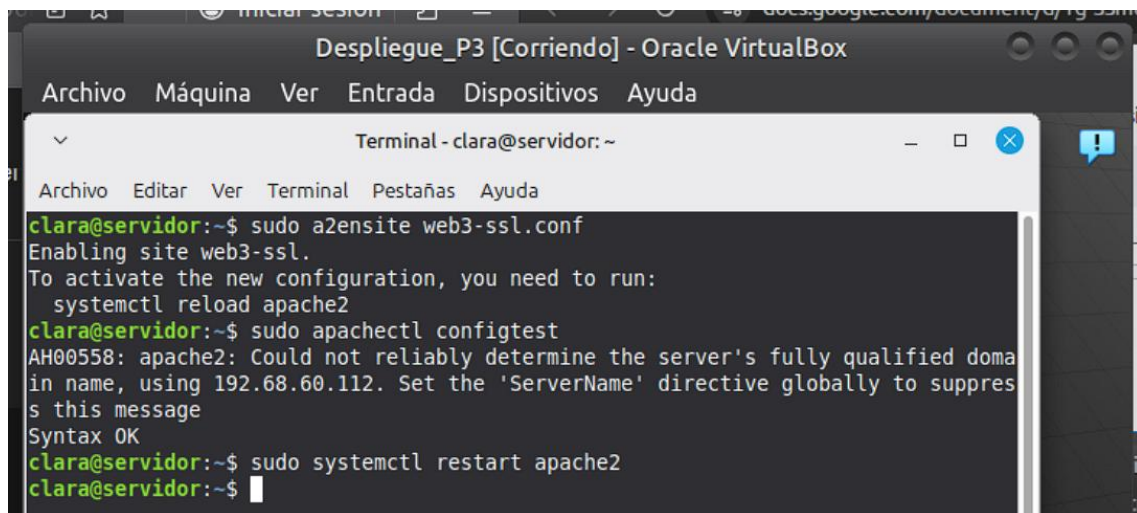
```
sudo apachectl configtest
```

```
sudo systemctl restart apache2
```

Debe aparecer:

```
Syntax OK
```

De esta forma habilita el sitio web seguro mediante `a2ensite` y se comprueba la sintaxis de la configuración con `apachectl configtest`. Finalmente, se reinicia el servicio Apache para aplicar la configuración HTTPS.



```
Despliegue_P3 [Corriendo] - Oracle VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda

Terminal - clara@servidor: ~
Archivo  Editar  Ver  Terminal  Pestañas  Ayuda

clara@servidor:~$ sudo a2ensite web3-ssl.conf
Enabling site web3-ssl.
To activate the new configuration, you need to run:
  systemctl reload apache2
clara@servidor:~$ sudo apachectl configtest
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 192.68.60.112. Set the 'ServerName' directive globally to suppress this message
Syntax OK
clara@servidor:~$ sudo systemctl restart apache2
clara@servidor:~$
```

6.5 Crear la página HTML de la Web nº 3 (verde)

Ejecuta:

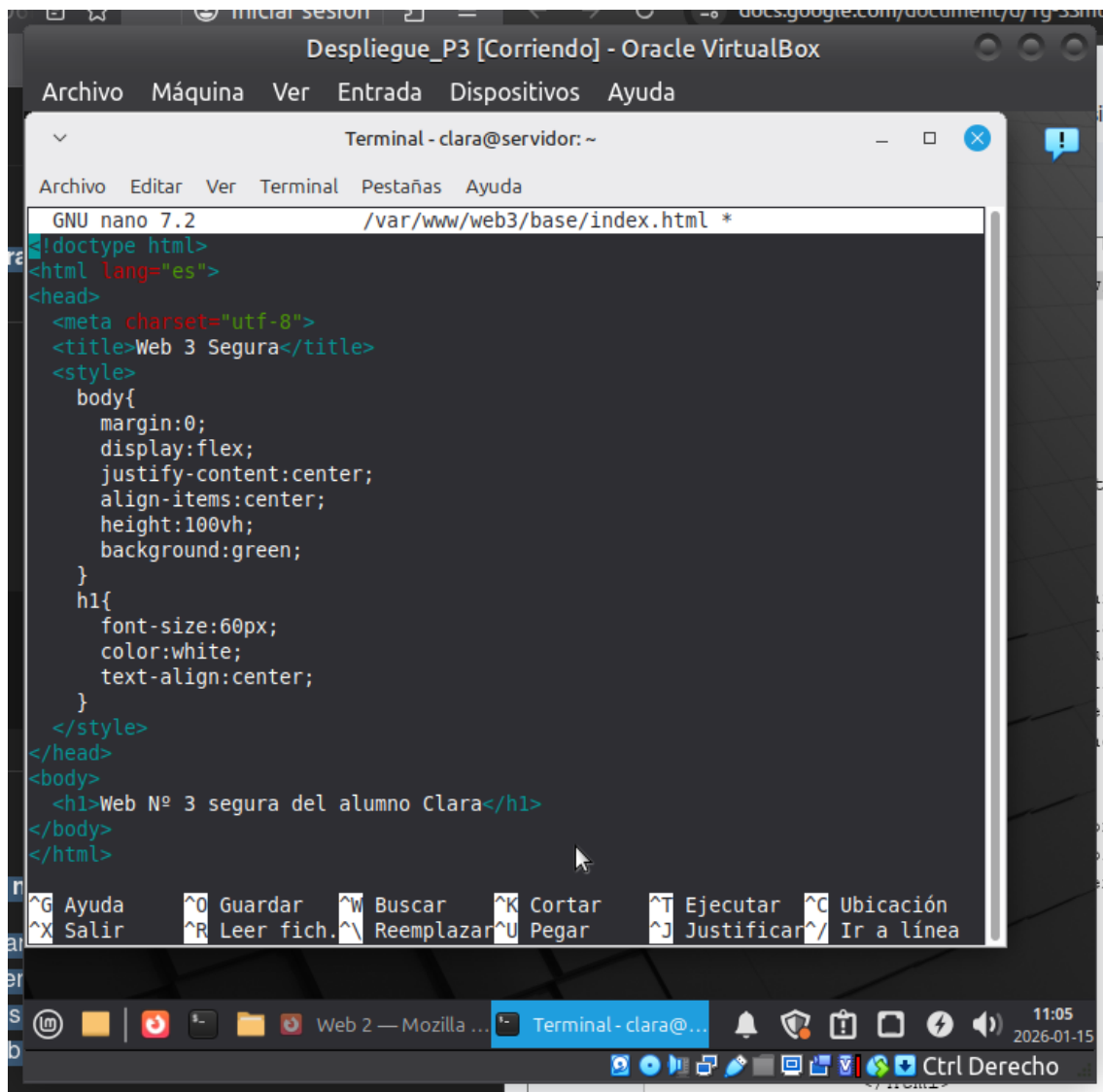
```
sudo nano /var/www/web3/base/index.html
```

Contenido

```
<!doctype html>
<html lang="es">
<head>
  <meta charset="utf-8">
  <title>Web 3 Segura</title>
  <style>
    body{
      margin:0;
      display:flex;
      justify-content:center;
      align-items:center;
      height:100vh;
      background:green;
    }
    h1{
      font-size:60px;
      color:white;
      text-align:center;
    }
  </style>
</head>
<body>
  <h1>Web Nº 3 segura del alumno Clara</h1>
</body>
</html>
```

Guarda y sal del editor.

Ya hemos creado el fichero principal index.html del servidor web nº 3 mediante el editor de texto nano, definiendo una página web con fondo de color verde y texto centrado de gran tamaño, conforme a los requisitos establecidos en el enunciado de la práctica.



6.6 Creación de los ficheros HTML adicionales de la Web nº 3

El servidor web nº 3 debe disponer de cinco ficheros HTML en su directorio base.

Mediante un bucle for que ejecutaremos en la terminal, se generan automáticamente cuatro páginas HTML adicionales (pagina2.html a pagina5.html), complementando al fichero principal index.html.

```
for i in 2 3 4 5; do

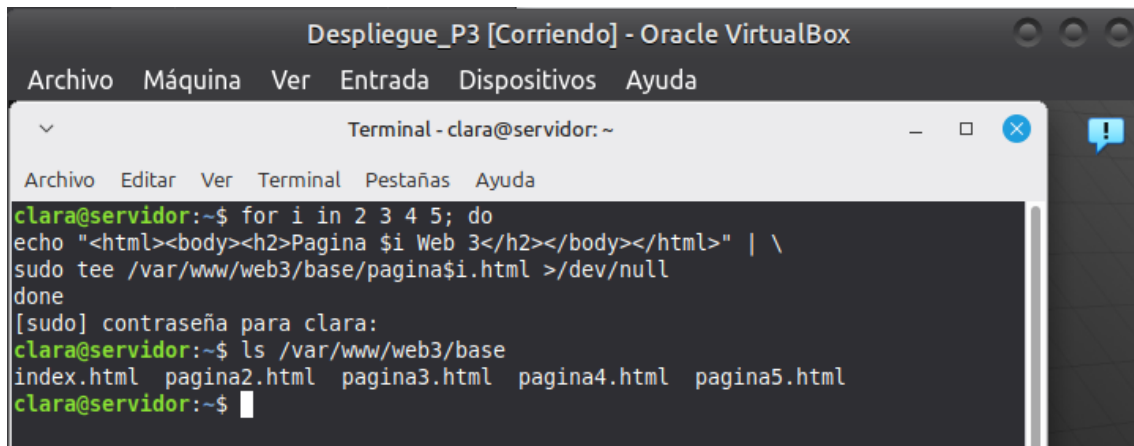
    echo "<html><body><h2>Pagina $i Web 3</h2></body></html>" | \

    sudo tee /var/www/web3/base/pagina$i.html >/dev/null

done
```

Comprobamos que se hayan creado correctamente:

```
ls /var/www/web3/base
```



```
Despliegue_P3 [Corriendo] - Oracle VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda

Terminal - clara@servidor: ~
Archivo  Editar  Ver  Terminal  Pestañas  Ayuda

clara@servidor:~$ for i in 2 3 4 5; do
echo "<html><body><h2>Pagina $i Web 3</h2></body></html>" | \
sudo tee /var/www/web3/base/pagina$i.html >/dev/null
done
[sudo] contraseña para clara:
clara@servidor:~$ ls /var/www/web3/base
index.html  pagina2.html  pagina3.html  pagina4.html  pagina5.html
clara@servidor:~$
```

6.7 Verificación del servidor web seguro desde terminal

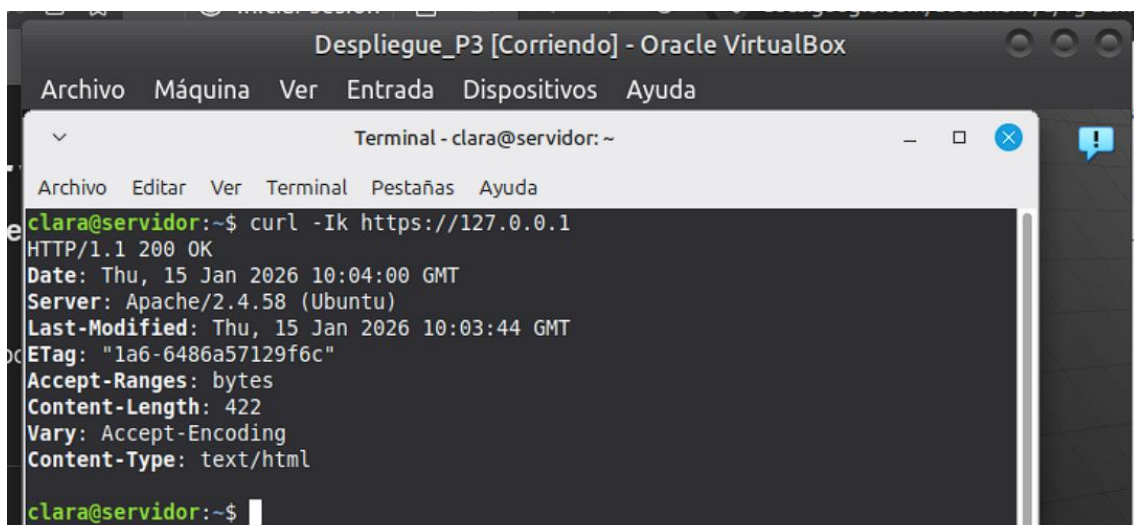
A continuación, vamos a comprobar que el servidor web nº 3 responde correctamente a peticiones HTTPS en el puerto 443.

Comando

```
curl -Ik https://127.0.0.1
```

Resultado esperado

```
HTTP/1.1 200 OK
```



```
Despliegue_P3 [Corriendo] - Oracle VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda

Terminal - clara@servidor: ~
Archivo  Editar  Ver  Terminal  Pestañas  Ayuda

clara@servidor:~$ curl -Ik https://127.0.0.1
HTTP/1.1 200 OK
Date: Thu, 15 Jan 2026 10:04:00 GMT
Server: Apache/2.4.58 (Ubuntu)
Last-Modified: Thu, 15 Jan 2026 10:03:44 GMT
ETag: "1a6-6486a57129f6c"
Accept-Ranges: bytes
Content-Length: 422
Vary: Accept-Encoding
Content-Type: text/html

clara@servidor:~$
```

Esto indica que el servidor HTTPS se encuentra operativo y responde correctamente a las peticiones seguras.

Se verifica el funcionamiento del servidor web seguro mediante el comando `curl -Ik`, realizando una petición HTTPS al puerto 443. La respuesta *HTTP/1.1 200 OK* confirma que el servidor web nº 3 está correctamente configurado y operativo.

6.8 Verificación del servidor web seguro desde el navegador

Comprueba el acceso real al servidor HTTPS desde un navegador web.

```
sudo nano /etc/apache2
```

Acceso desde el navegador

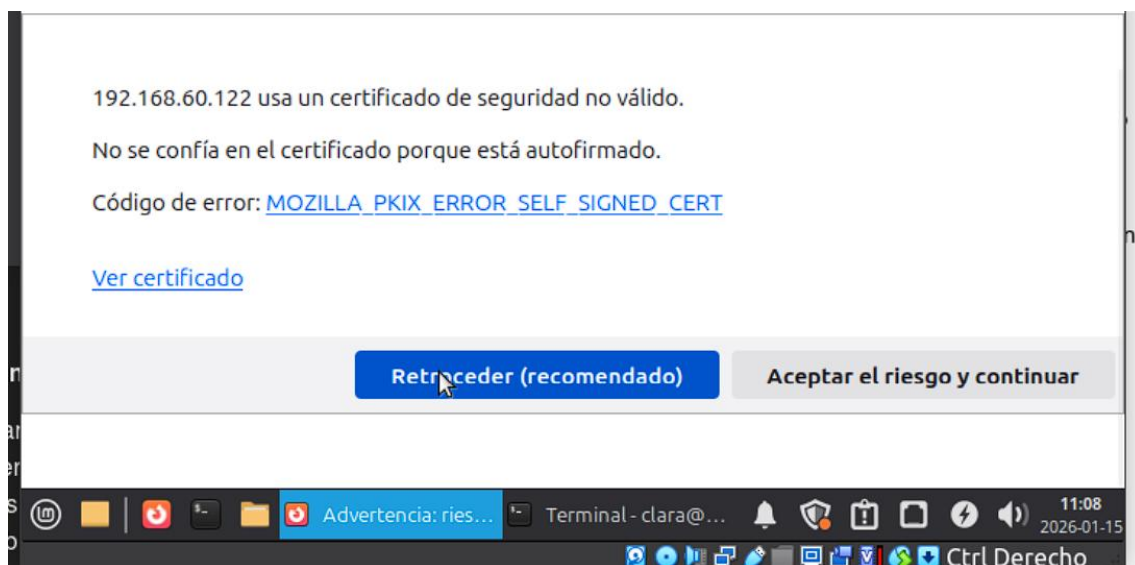
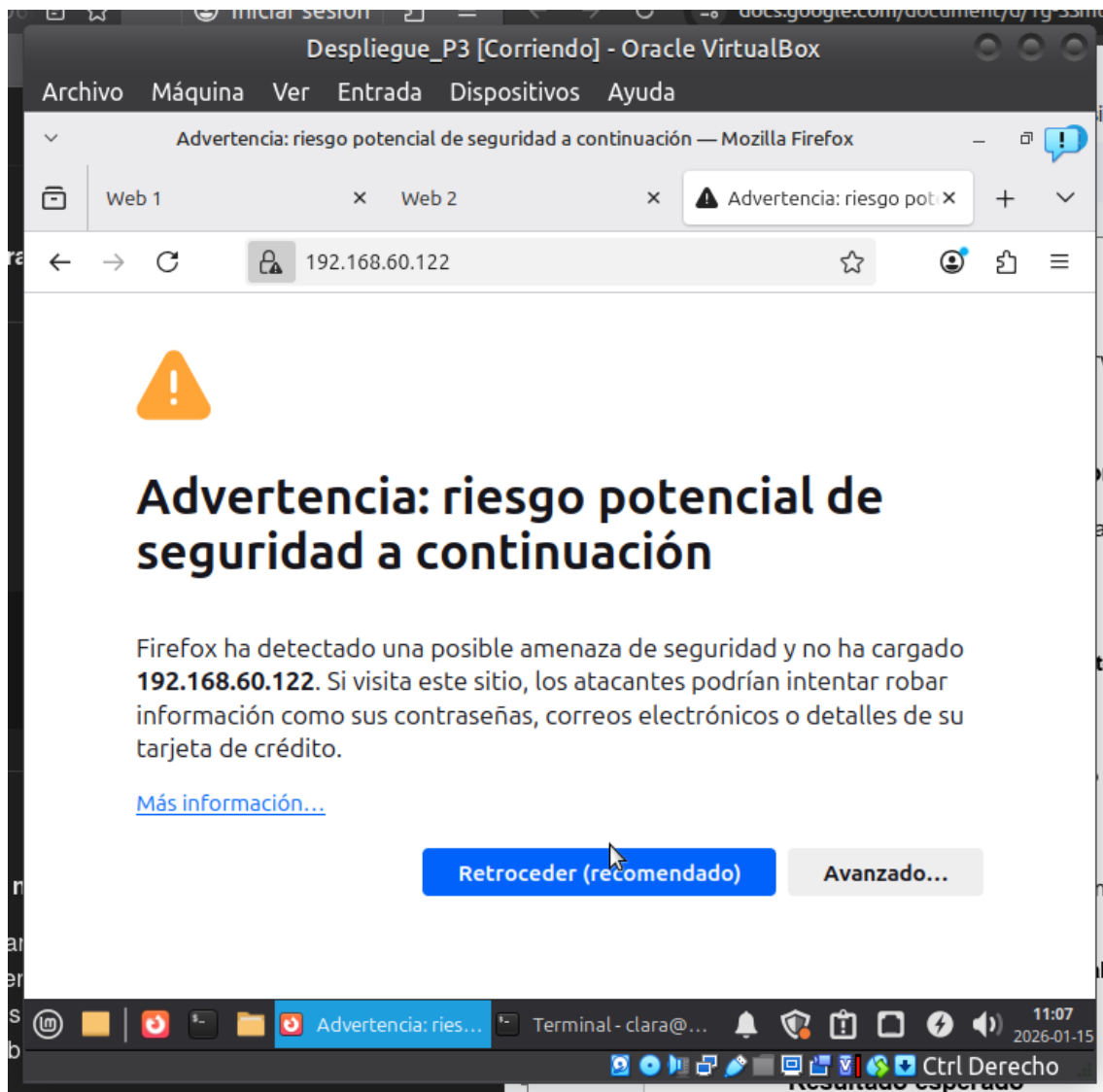
En la barra de direcciones del navegador, introducir:

```
https://192.168.60.122
```

Al tratarse de un **certificado autofirmado**, el navegador mostrará un aviso de seguridad.

Aceptación del certificado

1. Pulsar **Avanzado**
2. Aceptar el riesgo y continuar

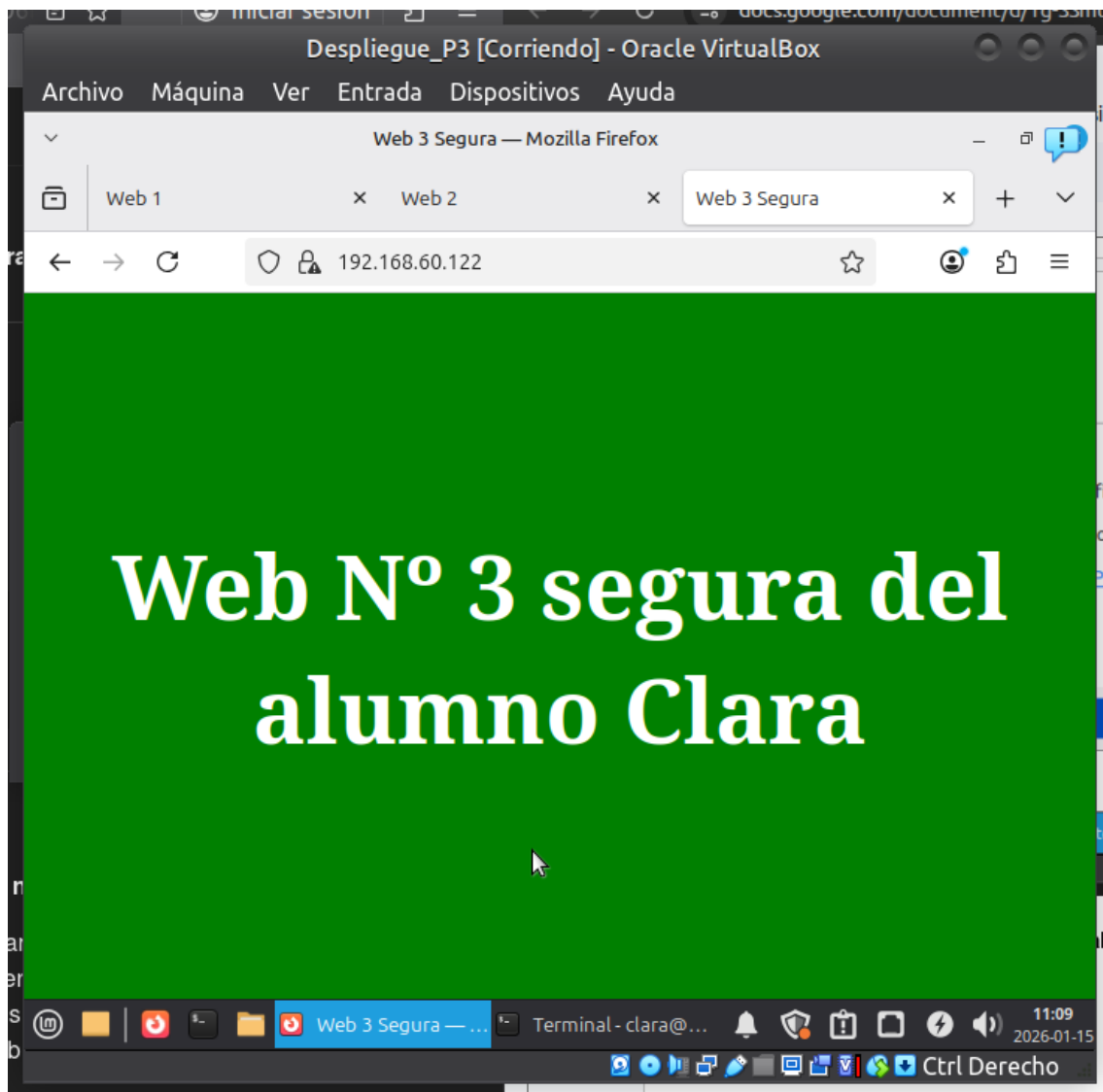


Este comportamiento es **normal y esperado** en certificados autofirmados.

Si clicamos en “ver certificado”, podemos observar los datos del certificado digital autofirmado, como el emisor, el sujeto, el periodo de validez y el algoritmo de cifrado utilizado, comprobando que ha sido generado por el propio alumno y que corresponde al servidor web seguro configurado en esta práctica.

Certificado

CLARA	
Nombre del asunto	
País	SP
Estado/Provincia	MADRID
Localidad	MADRID
Organización	IES SAN JUAN DE LA CRUZ
Unidad organizativa	-
Nombre común	CLARA
Dirección de correo electrónico	prueba@gmail.com
Nombre del emisor	
País	SP
Estado/Provincia	MADRID
Localidad	MADRID
Organización	IES SAN JUAN DE LA CRUZ
Unidad organizativa	-
Nombre común	CLARA
Dirección de correo electrónico	prueba@gmail.com
Validez	
No antes	Frí, 16 Jan 2026 11:48:40 GMT
No después	Sat, 16 Jan 2027 11:48:40 GMT
Información de clave	



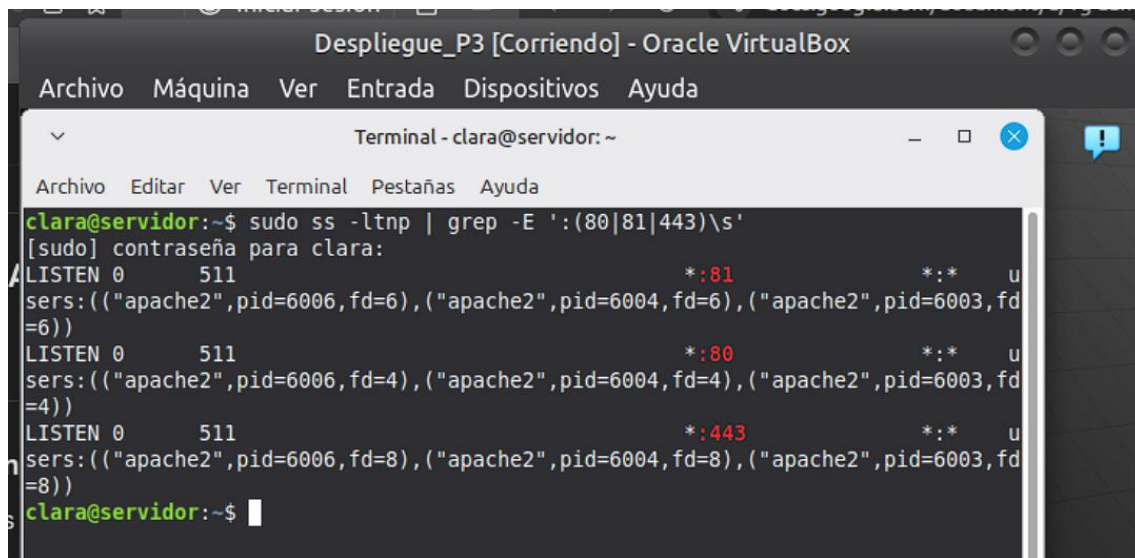
Se accede al **servidor web nº 3** mediante un **navegador web** utilizando el **protocolo HTTPS**. Al emplearse un **certificado autofirmado**, el navegador muestra un **aviso de seguridad** que es aceptado manualmente. Tras ello, se visualiza correctamente la página web segura, confirmando el correcto funcionamiento del servidor HTTPS.

6.9 Verificación final de puertos en escucha

Una vez finalizada la configuración de los tres servidores web definidos en la práctica, se realiza una comprobación final de los puertos en los que el servidor Apache se encuentra escuchando.

Para ello, se ejecuta el siguiente comando:

```
sudo ss -ltnp | grep -E ':(80|81|443)\s'
```



```
Despliegue_P3 [Corriendo] - Oracle VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda

Terminal - clara@servidor: ~
Archivo  Editar  Ver  Terminal  Pestañas  Ayuda

clara@servidor:~$ sudo ss -ltnp | grep -E ':(80|81|443)\s'
[sudo] contraseña para clara:
LISTEN 0      511          *:81          *.*          U
sers: (("apache2",pid=6006,fd=6),("apache2",pid=6004,fd=6),("apache2",pid=6003,fd=6))
LISTEN 0      511          *:80          *.*          U
sers: (("apache2",pid=6006,fd=4),("apache2",pid=6004,fd=4),("apache2",pid=6003,fd=4))
LISTEN 0      511          *:443         *.*          U
sers: (("apache2",pid=6006,fd=8),("apache2",pid=6004,fd=8),("apache2",pid=6003,fd=8))
clara@servidor:~$
```

El resultado muestra que Apache escucha correctamente en los **puertos 80 y 81**, correspondientes a los **servidores web no seguros**, así como en el **puerto 443**, correspondiente al **servidor web seguro configurado mediante HTTPS**. Esta verificación confirma la correcta implantación de la arquitectura web solicitada en el enunciado de la práctica.

INFORMACIÓN Y CONTENIDOS IMPORTANTES

1. JUSTIFICACIÓN DE LA ARQUITECTURA UTILIZADA

La arquitectura implementada en esta práctica se basa en el modelo **cliente-servidor**, utilizando una única máquina virtual con sistema operativo Linux Mint XFCE y el servidor web Apache. Sobre esta máquina se han configurado **tres servidores web diferenciados**, todos ellos alojados en el mismo sistema, pero correctamente separados a nivel lógico.

La diferenciación entre los servidores web se realiza mediante el uso de **VirtualHost por puerto**, lo que permite a Apache gestionar múltiples servicios web de forma simultánea. Cada servidor dispone de su propio directorio de contenidos y se accede exclusivamente mediante dirección IP, sin utilización de servicio DNS, cumpliendo estrictamente las condiciones del enunciado.

Esta arquitectura permite optimizar recursos, simplificar la administración del servidor y demostrar el funcionamiento de Apache como servidor web multipuerto con distintos niveles de seguridad

2. DIFERENCIA ENTRE SERVIDORES, PUERTOS Y NIVELES DE SEGURIDAD

En esta práctica se han configurado tres servidores web diferenciados mediante el uso de distintos puertos de escucha y protocolos:

- El **servidor web nº 1** utiliza el **protocolo HTTP** y escucha en el **puerto 80**, que es el puerto estándar para servicios web no seguros.
- El **servidor web nº 2** utiliza también **HTTP**, pero escucha en el **puerto 81**, un puerto alternativo que permite alojar un segundo servidor web no seguro en la misma máquina sin conflicto con el puerto 80.
- El **servidor web nº 3** utiliza el **protocolo HTTPS** y escucha en el **puerto 443**, el puerto estándar para comunicaciones web seguras, empleando cifrado SSL/TLS mediante un certificado digital autofirmado.

Desde el punto de vista de la seguridad, los servidores web nº 1 y nº 2 se consideran **no seguros**, ya que la información se transmite en texto plano, mientras que el servidor web nº 3 proporciona un **canal de comunicación cifrado**, garantizando un mayor nivel de protección de los datos.

3. DESCRIPCIÓN DE HTTP Y HTTPS

HTTP (HyperText Transfer Protocol) es un protocolo de comunicación web basado en el modelo petición–respuesta. Se trata de un protocolo sin estado y **no cifrado**, por lo que los datos transmitidos pueden ser interceptados. En esta práctica, HTTP se utiliza en los servidores web nº 1 y nº 2, adecuados para entornos de prueba o contenidos que no requieren confidencialidad.

HTTPS es la versión segura de HTTP. Utiliza el protocolo SSL/TLS para **cifrar la comunicación** entre cliente y servidor, garantizando la confidencialidad, integridad y autenticidad de los datos. En esta práctica se emplea HTTPS en el servidor web nº 3 mediante un **certificado digital autofirmado**, suficiente para entornos de laboratorio y aprendizaje.

4. RELACIÓN ENTRE PROTOCOLOS, PUERTOS Y FUNCIONAMIENTO DEL SERVIDOR APACHE

Apache asocia cada protocolo a un **puerto de escucha específico**, de modo que cuando recibe una petición entrante identifica el puerto de destino y dirige la solicitud al **VirtualHost correspondiente**. Este mecanismo permite que un único servidor Apache gestione múltiples servicios web diferenciados, utilizando distintos protocolos y niveles de seguridad sin interferencias.

Gracias a esta relación entre protocolos, puertos y VirtualHost, es posible alojar simultáneamente servidores web HTTP y HTTPS en la misma máquina virtual, garantizando su correcto funcionamiento y diferenciación.

CONCLUSIÓN

Mediante el desarrollo de esta práctica se ha configurado correctamente un entorno web basado en Apache sobre una máquina virtual con Linux Mint XFCE, alojando **tres servidores web diferenciados en una misma máquina** mediante el uso de VirtualHost por puerto.

Durante el proceso se han verificado la conectividad de red, la instalación y el estado del servidor Apache, la correcta configuración de los puertos 80, 81 y 443, así como la organización de directorios, permisos y contenidos exigidos. Además, se ha implementado un servidor web seguro mediante HTTPS, utilizando un certificado digital autofirmado y comprobando su funcionamiento tanto desde terminal como desde navegador.

Las pruebas realizadas confirman que los tres servidores web se encuentran operativos, accesibles mediante dirección IP y correctamente diferenciados, cumpliendo así todos los requisitos técnicos y documentales establecidos en el enunciado de la práctica.