ALEBUNTU
No Problem

Manual de APACHE2

Revisado: 20/02/2021

ALEBUNTU
No Problem

Índice

Requisitos **3**

Introducción **4**

Importar máquina virtual. **5**

Configuración de Red Del VirtualBox **6**
Servidor 7

Configuración de red **9**
Máquina Virtual 9
Maquina real 10
Máquina Virtual 13

Instalación del BIND9 **14**
Maquina Real 14

Instalacion del APACHE2 **24**
Maquina Real 24

Prácticas con APACHE2 **27**
PRACTICA 1 28
PRÁCTICA 2 30
PRÁCTICA 3 33
PRÁCTICA 4,5,6,7 36
Comprobaciones 45

ALEBUNTU
No Problem

Requisitos

1. Tener instalada la última versión de VirtualBox.

Si la teneis instalada tendréis un acceso directo como este:

2. Haber descargado el .ova para hacer las máquinas virtuales.

Si lo habéis descargado tendréis un archivo como este.

ALEBUNTU
No Problem

Introducción

Bienvenidos al manual de APACHE2 de Alebuntu, en esta práctica veremos cómo hacer una práctica de servidor web mediante una máquina virtual Ubuntu.

En esta práctica trataremos de hacer un servidor web con APACHE2.

Un servidor web principalmente sirve para proporcionar páginas web, lo que conseguimos metiendo una ip o una url y podemos ver una página web.

La práctica se dará finalizada al hacer unas cuantas prácticas con los ficheros de configuración de APACHE2.

El servidor web de APACHE2 se basa en unos cuantos ficheros de configuración y en ficheros html.

Antes de esta práctica debemos haber cumplido los dos requisitos:

¿Tenéis instalado última versión de virtualbox? Sí

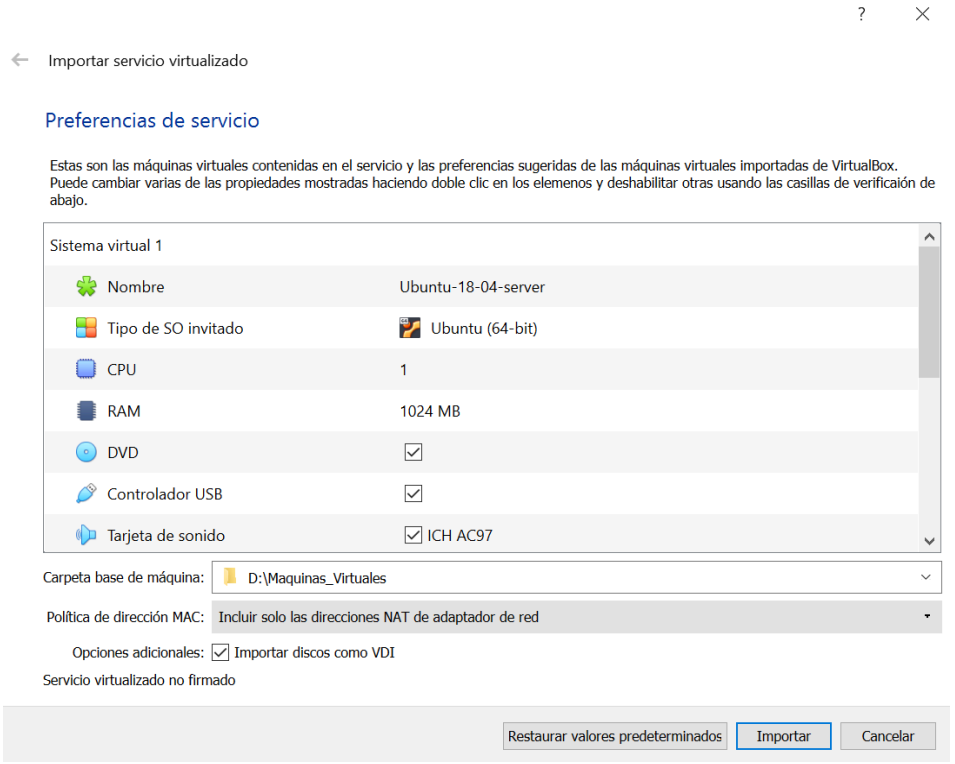
¿Os habéis descargado el .ova para importar máquinas virtuales? Sí

Perfecto habéis cumplido los requisitos por lo tanto podemos empezar la práctica.

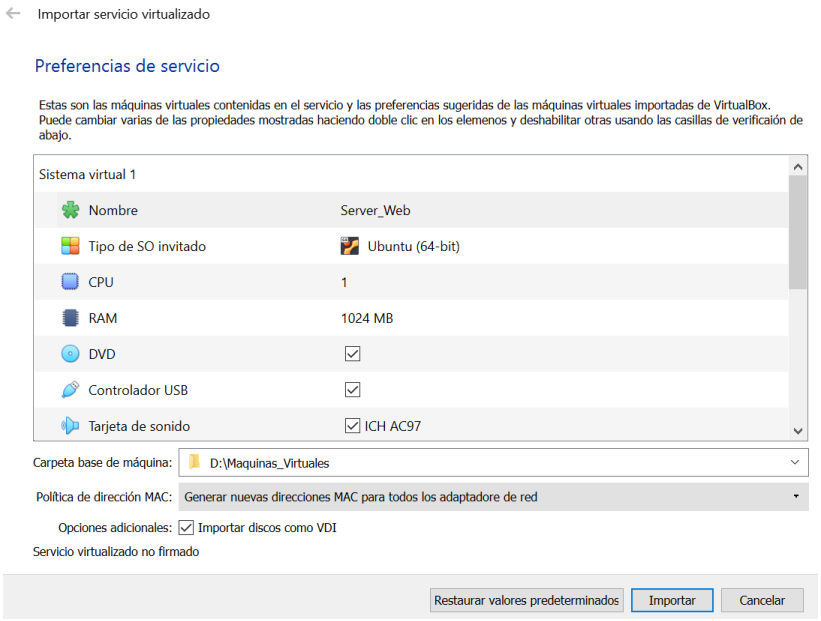
Importar máquina virtual.

Para pinchamos dos veces rápidamente en el archivo .ova

Luego saldrá la siguiente imagen:

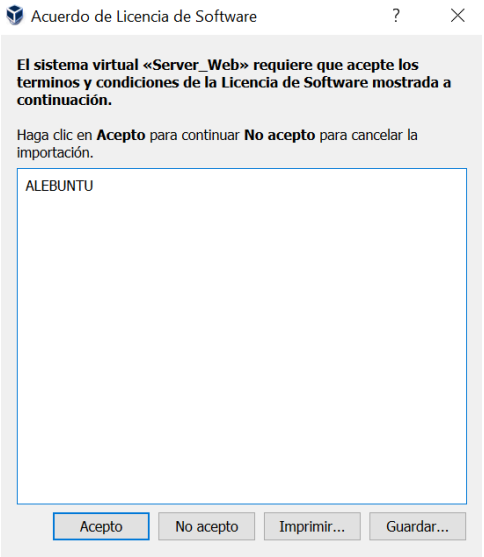


Luego cambiamos los siguientes datos: *nombre y política de dirección MAC.*
Debe quedar una cosa así:

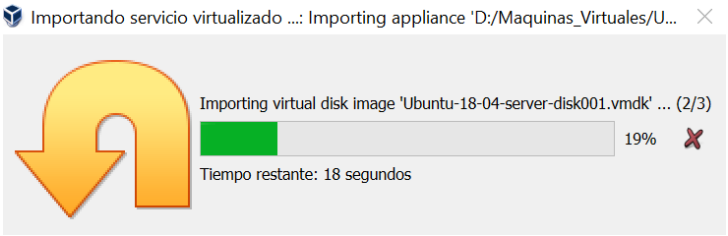


Por último le damos a importar

Nos saldrá la siguiente imagen:

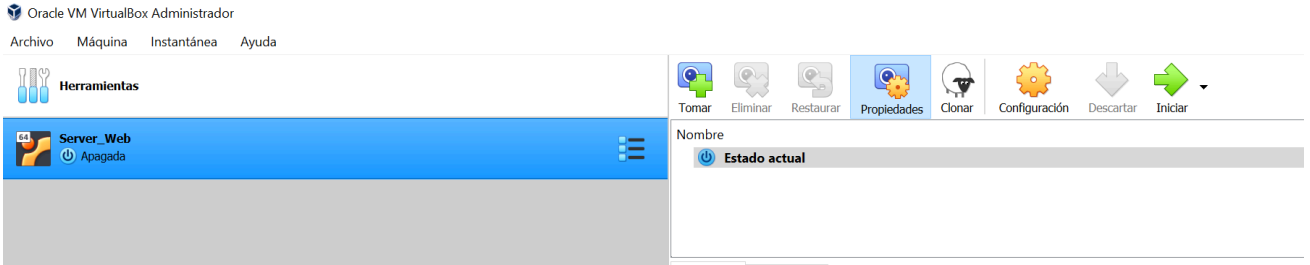


Aceptamos los acuerdos de licencia y saldrá una pantalla de carga:



Una vez que termine la pantalla de carga, que tardará unos minutos tendremos importada la máquina virtual del servidor.

Y al final del proceso nos quedará una cosa así:



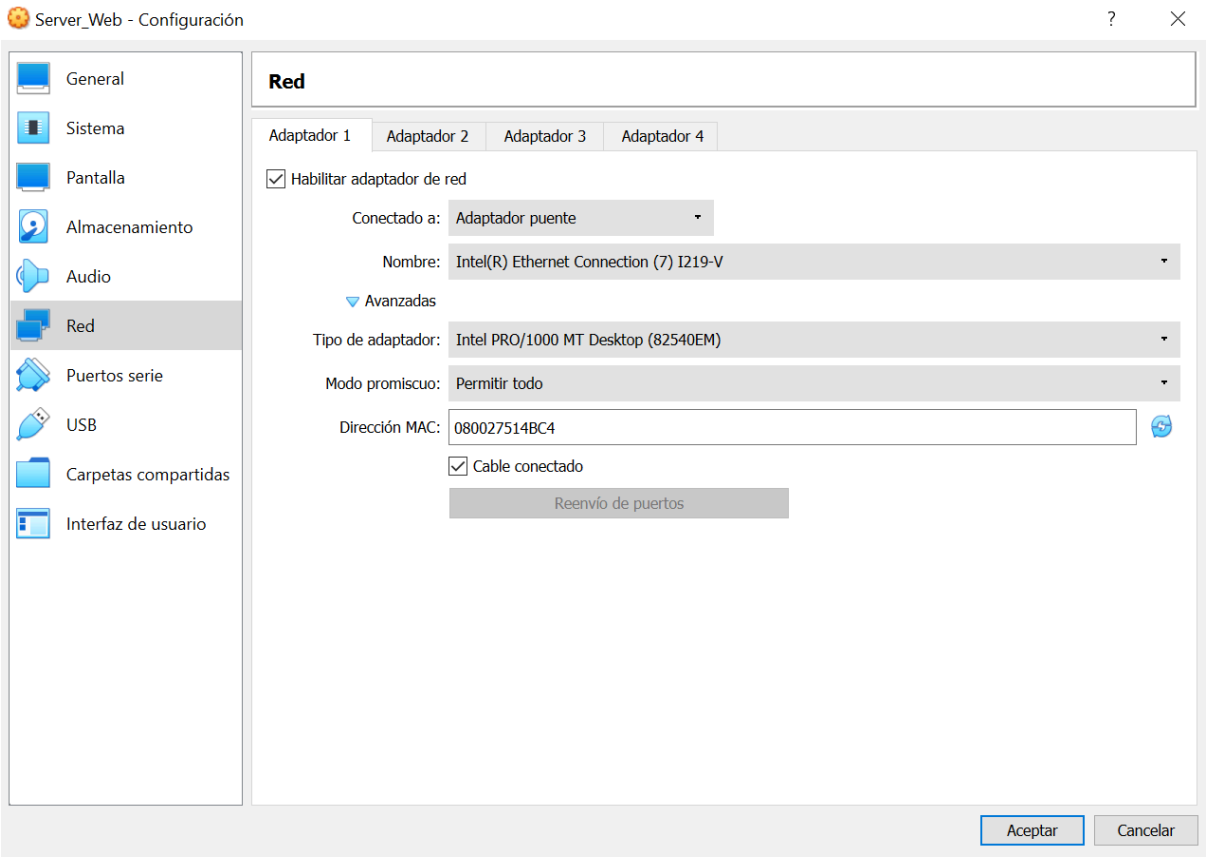
Configuración de Red Del VirtualBox

Una vez que tenemos una máquina servidor, antes de iniciar la máquina virtual, configuraremos la tarjeta de red de la máquina virtual.

Servidor

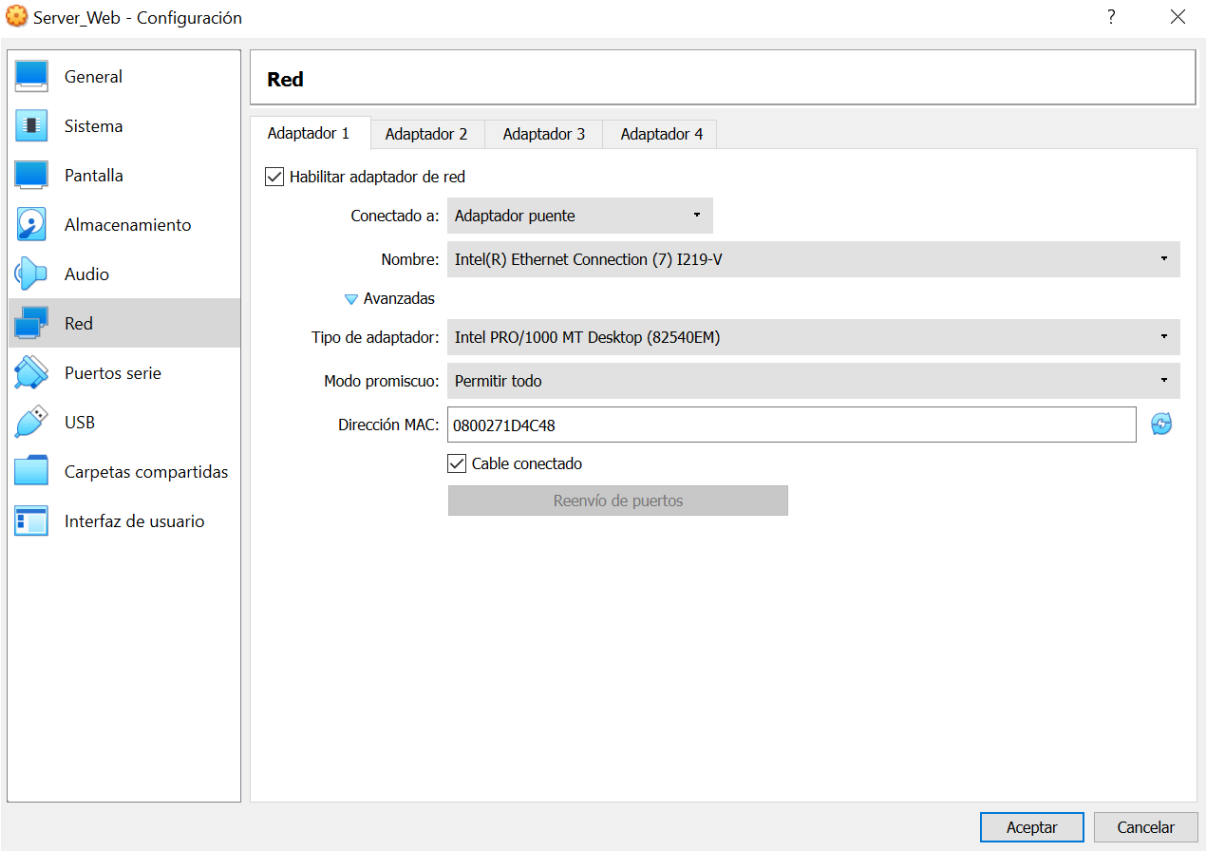
Servidor antes de iniciar la máquina virtual**.**

Primero para el servidor comprobamos que el adaptador 1 esta como adaptador puente y el modo promiscuo está en permitir todo.



Luego reinicializar la MAC unas cuantas veces para hacerlo solo hay que pinchar en el icono azul que está al lado de la MAC.

Y como podremos ver se ha cambiado la MAC:



Por último le damos a Aceptar y habremos terminado la configuración de la máquina virtual del servidor.

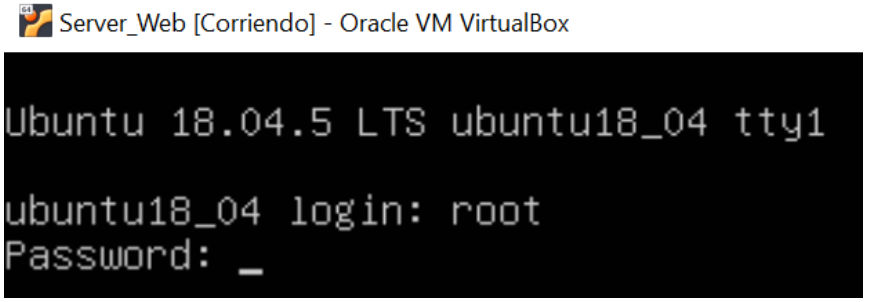
Ahora ya estáis listos para la máquina servidor.

Configuración de red

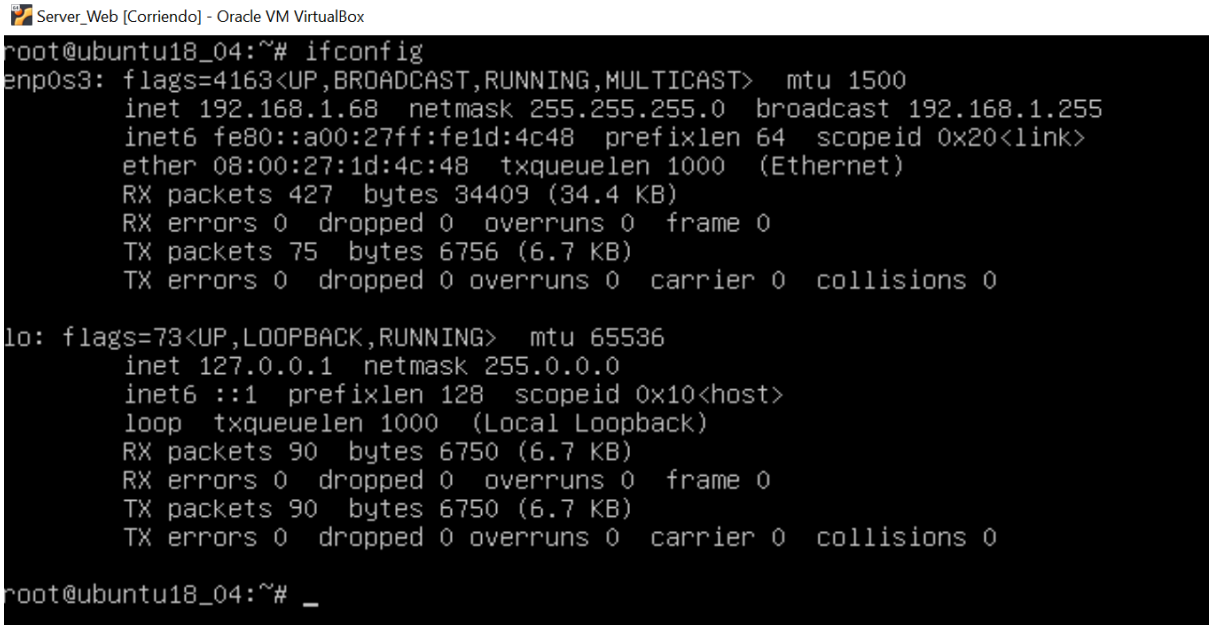
Máquina Virtual

Iniciamos como root.

Login: *root*
La contraseña para iniciar como root es : *usuario*



Miramos la ip de nuestra máquina virtual para poder conectarnos por ssh:

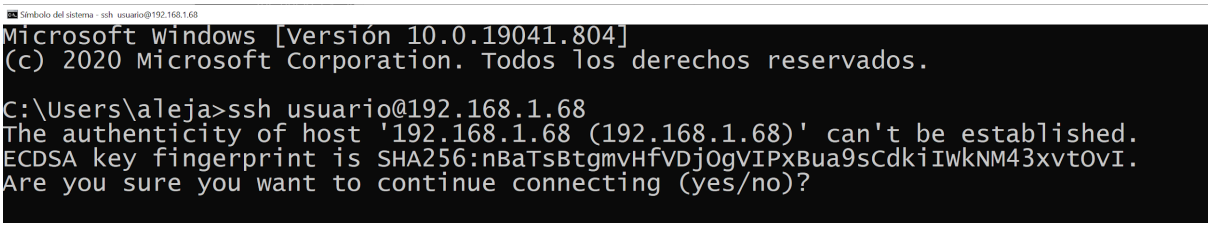


Para conectarnos por ssh necesitamos una ip y un usuario, en este caso la ip es *192.168.1.68* y el nombre del usuario al que nos podemos conectar es *usuario*.

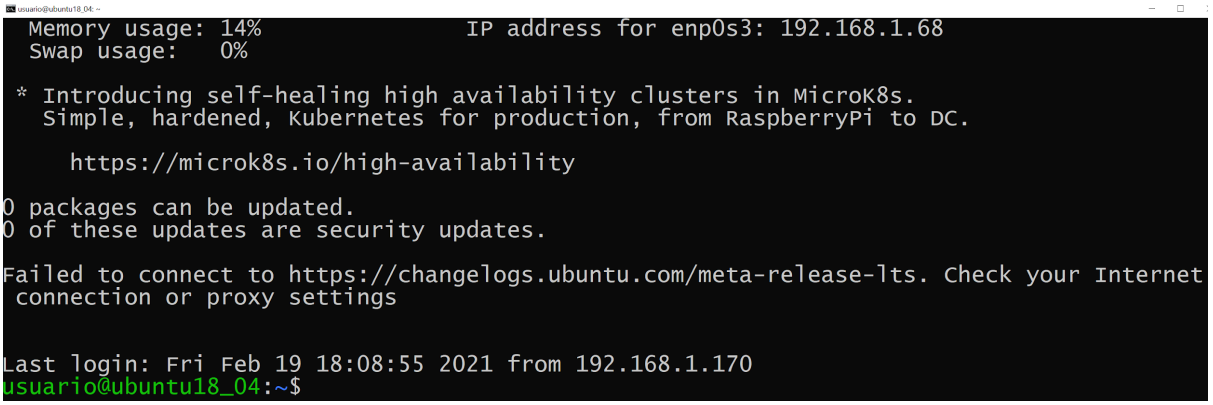
ALEBUNTU
No Problem

Maquina real

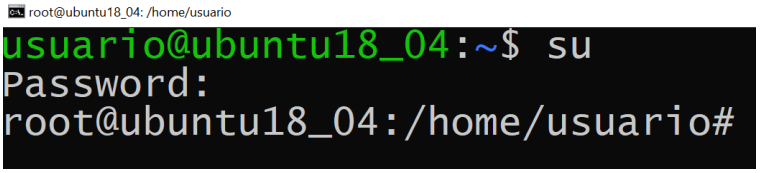
Para conectarnos por ssh, nos vamos a la terminal de nuestro ordenador real y hacemos ssh a la máquina virtual: **ssh usuario@192.168.1.68**



Si lo hemos conseguido en la máquina real nos saldrá una cosa así:

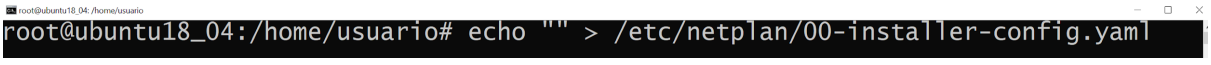


Ahora en ese misma terminal nos iniciamos como root y nos olvidamos de la terminal de la máquina virtual.



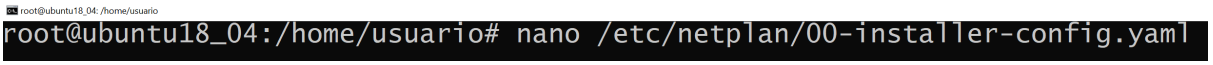
Borramos todo lo que tenga el fichero de configuración de red:

echo "" > /etc/netplan/00-installer-config.yaml



Ahora configuramos la tarjeta de red del servidor mediante el archivo de configuración de netplan:

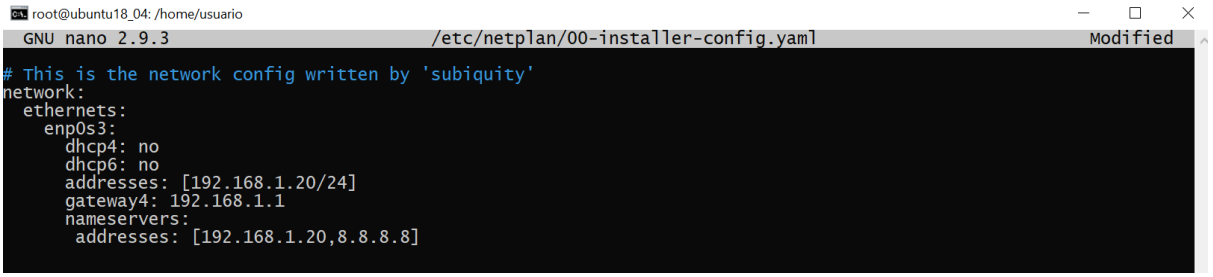
nano /etc/netplan/00-installer-config.yaml



Añadimos la siguiente configuración al fichero:

 # This is the network config written by 'subiquity'
network:
ethernets:
enp0s3:
dhcp4: no
dhcp6: no
addresses: [192.168.1.20/24]
gateway4: 192.168.1.1
nameservers:
addresses: [192.168.1.20,8.8.8.8]

Tiene que quedar una cosa así (Tener todo en cuenta, es decir que este todo bien tabulado):



Por último pulsamos control + x
Pulsamos Y y despues Enter y habremos guardado la configuración.

Actualizamos el servicio de red: **netplan apply**

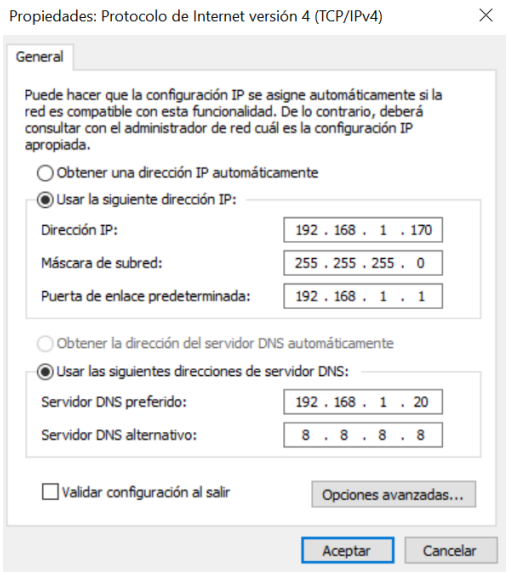
```
root@ubuntu18_04: /home/usuario
root@ubuntu18_04:/home/usuario# netplan apply
root@ubuntu18_04:/home/usuario#
```

Si todo va bien el comando no devolverá ningún error.

Y nos habrá sacado de la conexión de ssh debido al cambio de ip de la máquina virtual.

```
Símbolo del sistema
root@ubuntu2004server:/home/usuario# netplan apply
Connection reset by 192.168.1.59 port 22

C:\Users\aleja>
```

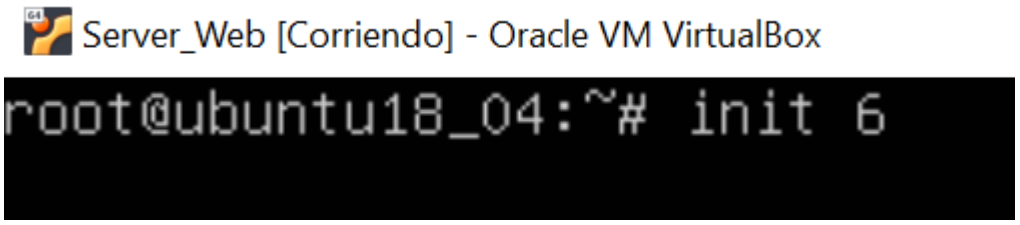
Posdata para la práctica recordar poner como dns en la máquina real lo siguiente:



Máquina Virtual

Por último reiniciamos la máquina virtual desde la cmd de la propia máquina virtual ya que al haber trastocado la configuración de red ,se nos habrá cortado la conexión por ssh que teníamos desde la máquina real.

Comando para reiniciar la máquina: **init 6**



ALEBUNTU
No Problem

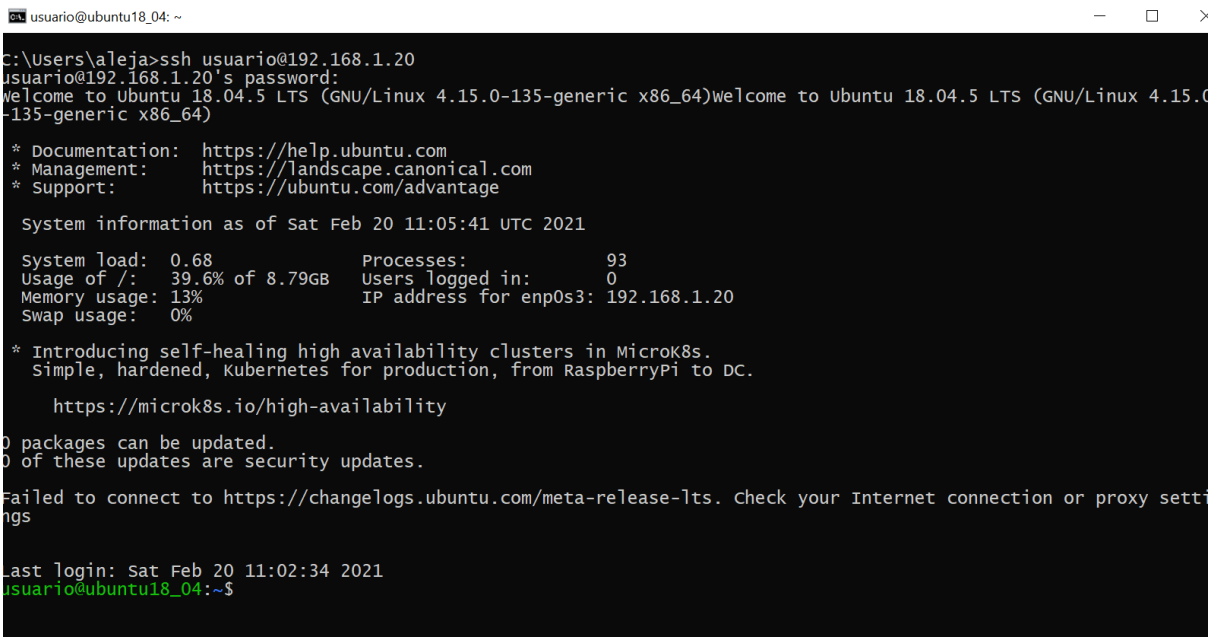
Instalación del BIND9

Os preguntareis para que instalaremos y configuraremos el servidor DNS porque vamos a simular que tenemos una página web como otra cualquiera con una url y para ello necesitamos un servidor DNS.

Maquina Real

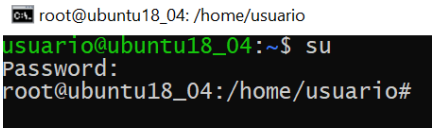
Para ello metiéndonos de nuevo con la nueva ip asignada a la máquina virtual nos metemos por ssh al servidor:

ssh usuario@192.168.1.20

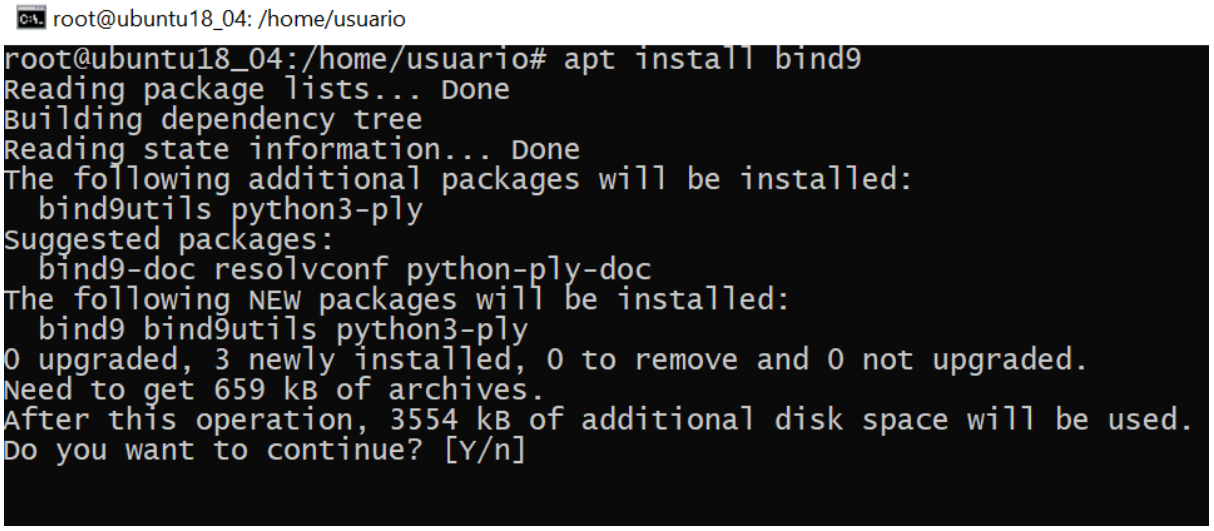


Una vez dentro, en la terminal de la máquina real nos ponemos a instalar el servidor DNS.

Comando para ser root: **su**

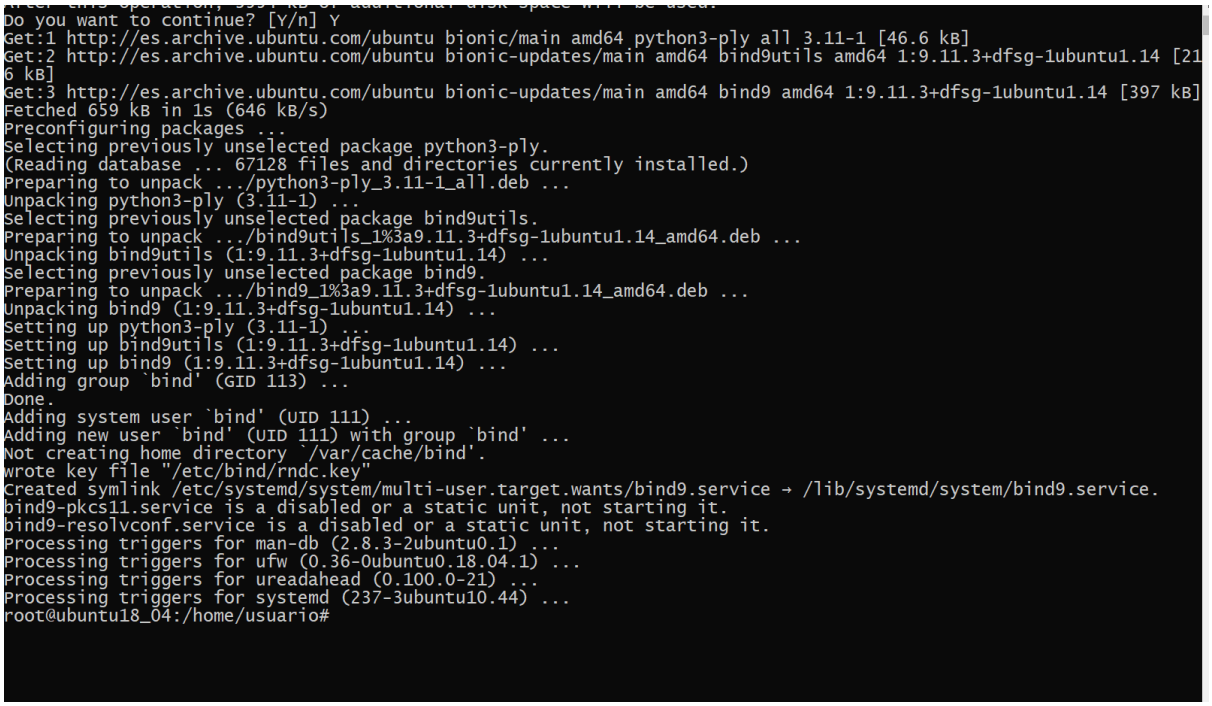


Comando de instalación: **apt install bind9**



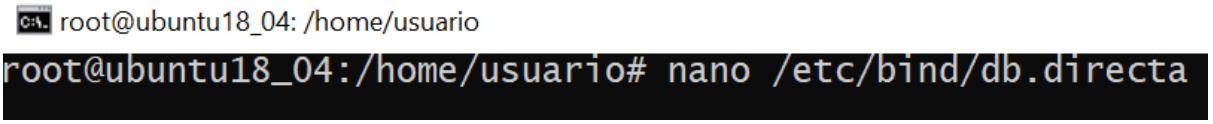
Pulsamos Y y después ENTER.

Una vez terminada la instalación nos quedará una cosa así:



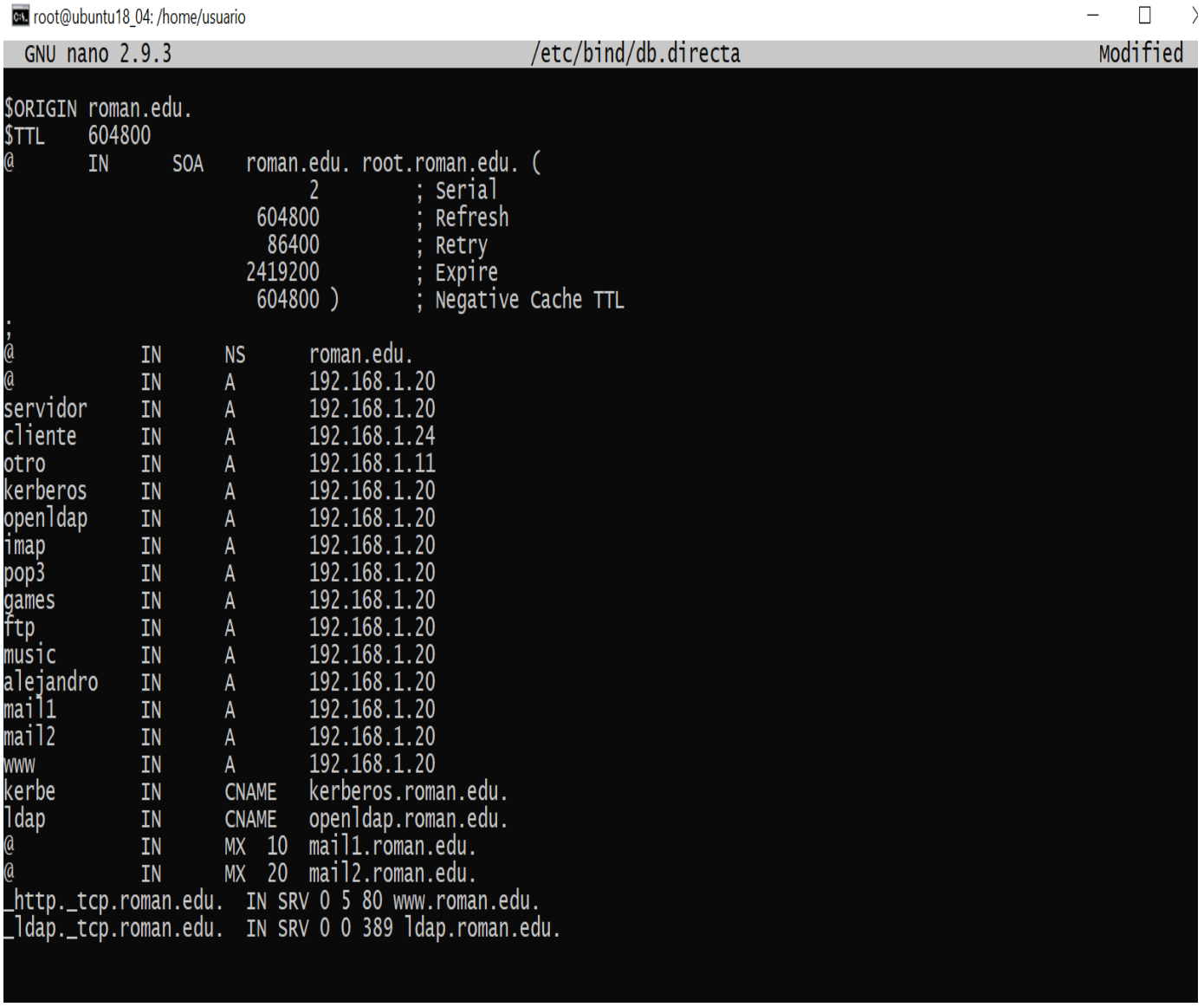
Comando para introducir la zona directa:

nano /etc/bind/db.directa




```
$ORIGIN roman.edu.
$TTL    604800
@       IN      SOA     roman.edu. root.roman.edu. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@           IN      NS      roman.edu.
@           IN      A       192.168.1.20
servidor    IN      A       192.168.1.20
cliente     IN      A       192.168.1.24
otro        IN      A       192.168.1.11
kerberos    IN      A       192.168.1.20
openldap    IN      A       192.168.1.20
imap        IN      A       192.168.1.20
pop3        IN      A       192.168.1.20
games       IN      A       192.168.1.20
ftp         IN      A       192.168.1.20
music       IN      A       192.168.1.20
alejandro   IN      A       192.168.1.20
mail1       IN      A       192.168.1.20
mail2       IN      A       192.168.1.20
www         IN      A       192.168.1.20
kerbe       IN      CNAME   kerberos.roman.edu.
ldap        IN      CNAME   openldap.roman.edu.
@           IN      MX  10  mail1.roman.edu.
@           IN      MX  20  mail2.roman.edu.
_http._tcp.roman.edu.  IN SRV 0 5 80 www.roman.edu.
_ldap._tcp.roman.edu.  IN SRV 0 0 389 ldap.roman.edu.
```

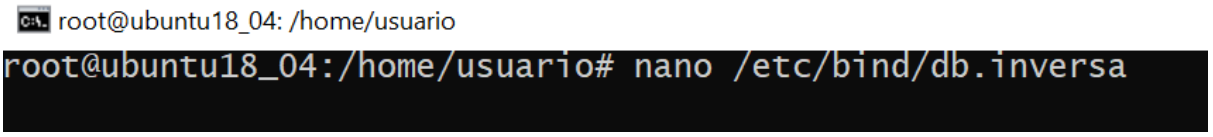
Quedará una cosa así:



Después pulsamos CONTROL + X , pulsamos Y y después ENTER y se habrá guardado ese archivo de configuración.

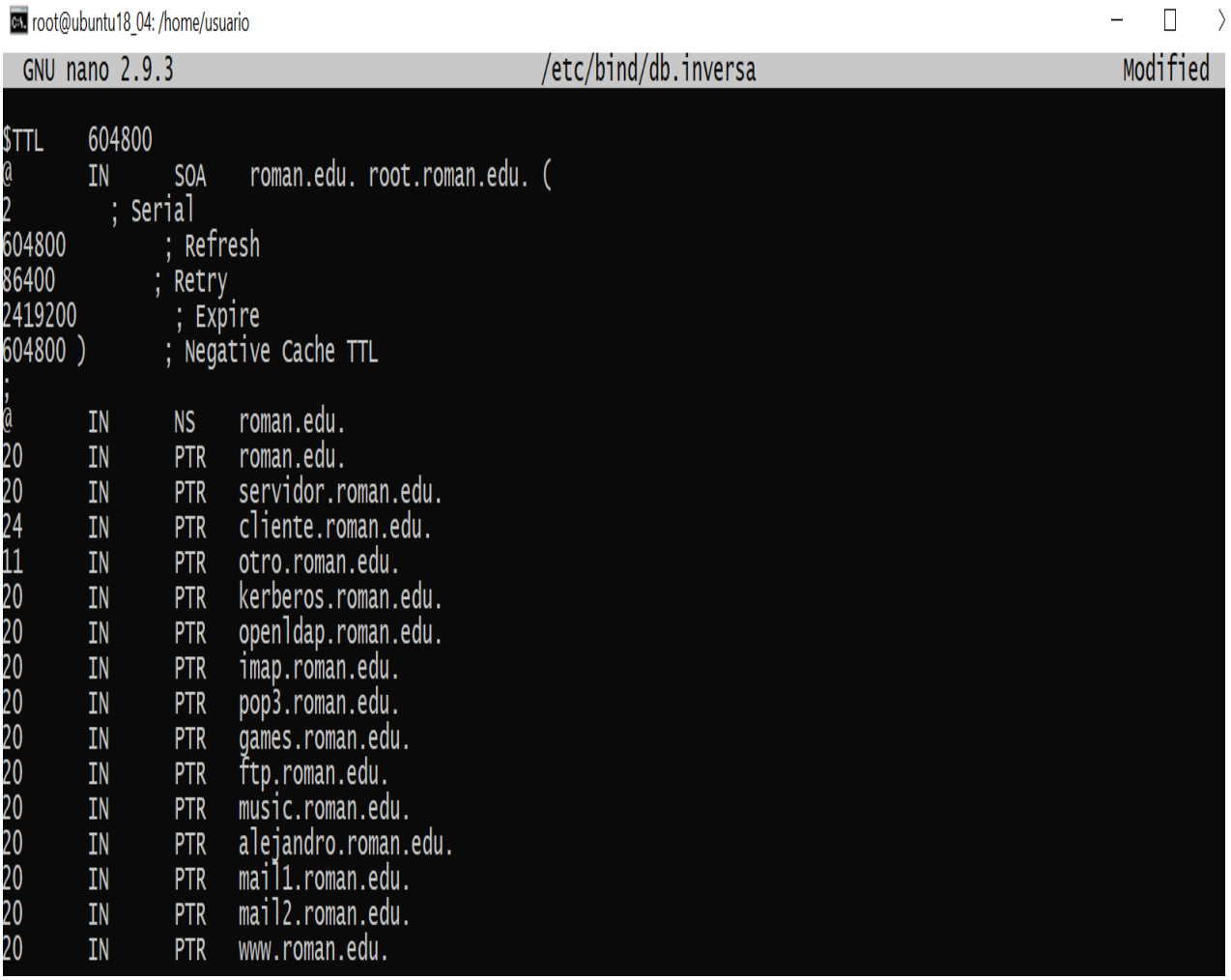
Comando para introducir la zona inversa:

nano /etc/bind/db.inversa



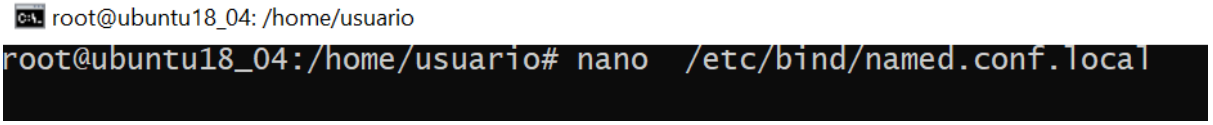

```
$TTL    604800
@       IN      SOA     roman.edu. root.roman.edu. (
2         ; Serial
604800        ; Refresh
86400         ; Retry
2419200         ; Expire
604800 )      ; Negative Cache TTL
;
@       IN      NS      roman.edu.
20      IN      PTR     roman.edu.
20      IN      PTR     servidor.roman.edu.
24      IN      PTR     cliente.roman.edu.
11      IN      PTR     otro.roman.edu.
20      IN      PTR     kerberos.roman.edu.
20      IN      PTR     openldap.roman.edu.
20      IN      PTR     imap.roman.edu.
20      IN      PTR     pop3.roman.edu.
20      IN      PTR     games.roman.edu.
20      IN      PTR     ftp.roman.edu.
20      IN      PTR     music.roman.edu.
20      IN      PTR     alejandro.roman.edu.
20      IN      PTR     mail1.roman.edu.
20      IN      PTR     mail2.roman.edu.
20      IN      PTR     www.roman.edu.
```

Quedará una cosa así:



Después pulsamos CONTROL + X , pulsamos Y y después ENTER y se habrá guardado ese archivo de configuración.

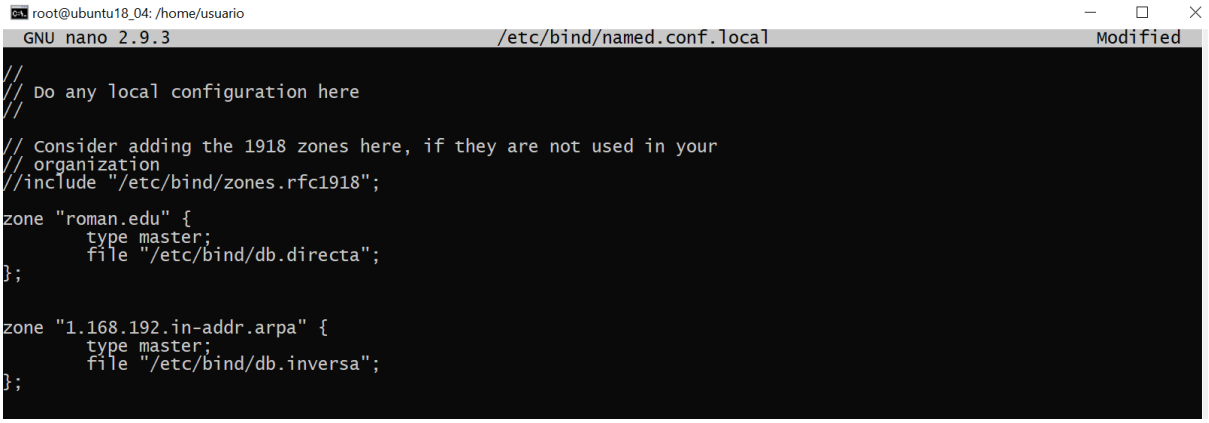
Comando para configurar donde están almacenadas las zonas:
nano /etc/bind/named.conf.local



zone "roman.edu" {
type master;
file "/etc/bind/db.directa";
};

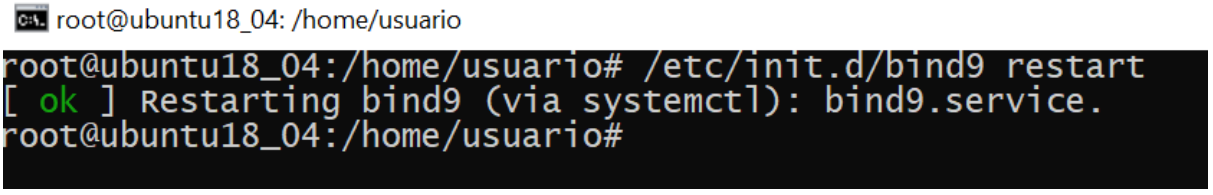
zone "1.168.192.in-addr.arpa" {
type master;
file "/etc/bind/db.inversa";
};

Quedará una cosa así:

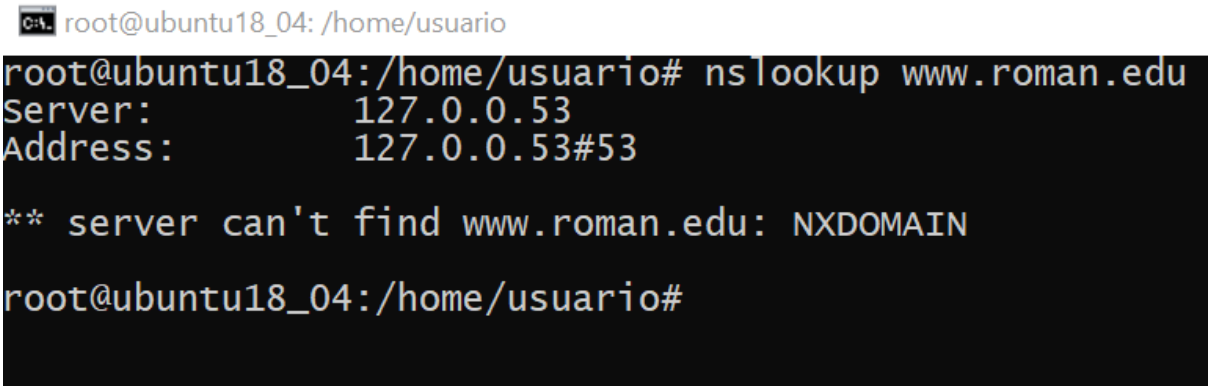


Después pulsamos CONTROL + X , pulsamos Y y después ENTER y se habrá guardado ese archivo de configuración.

Al terminar de configurar el servidor dns, reiniciamos el servicio:
/etc/init.d/bind9 restart

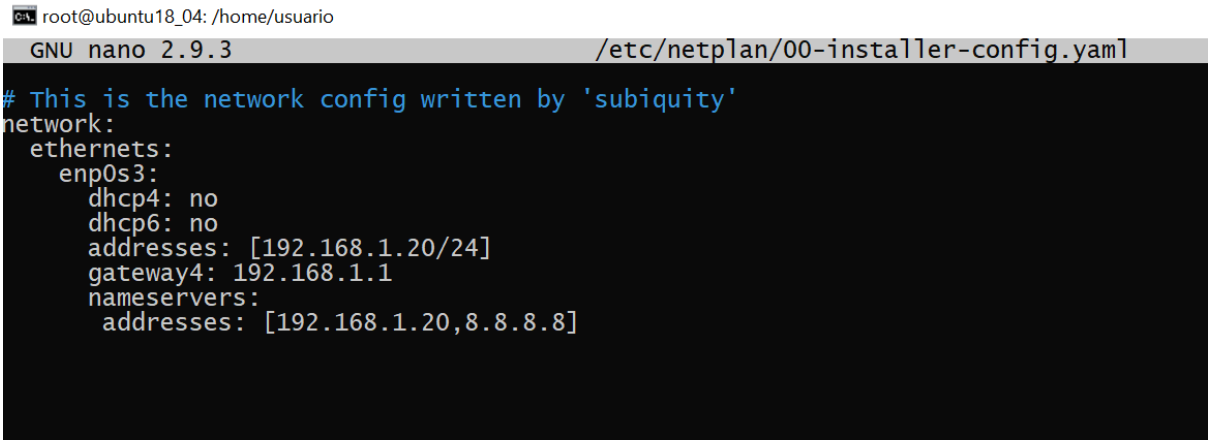


Si hacemos la comprobación veremos que esta fallida:
nslookup www.roman.edu

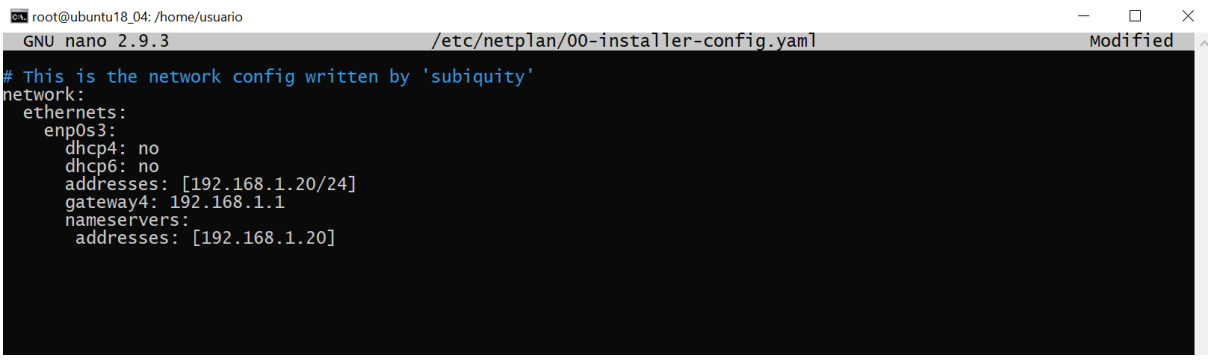


Eso es porque ahora mismo no está cogiendo bien el servidor dns, para ello editamos el archivo de configuración de la red:
nano /etc/netplan/00-installer-config.yaml

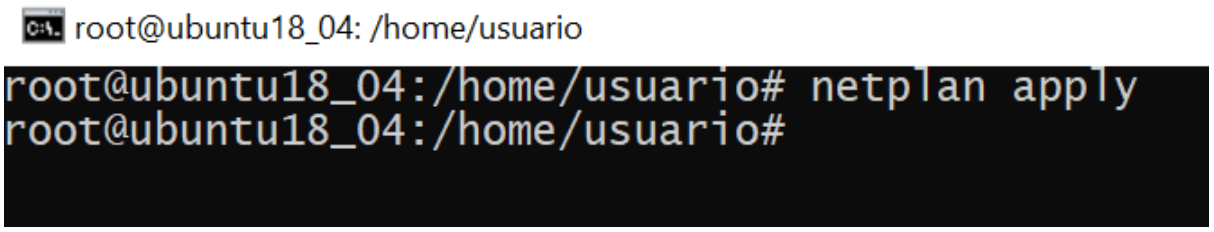
Antes:



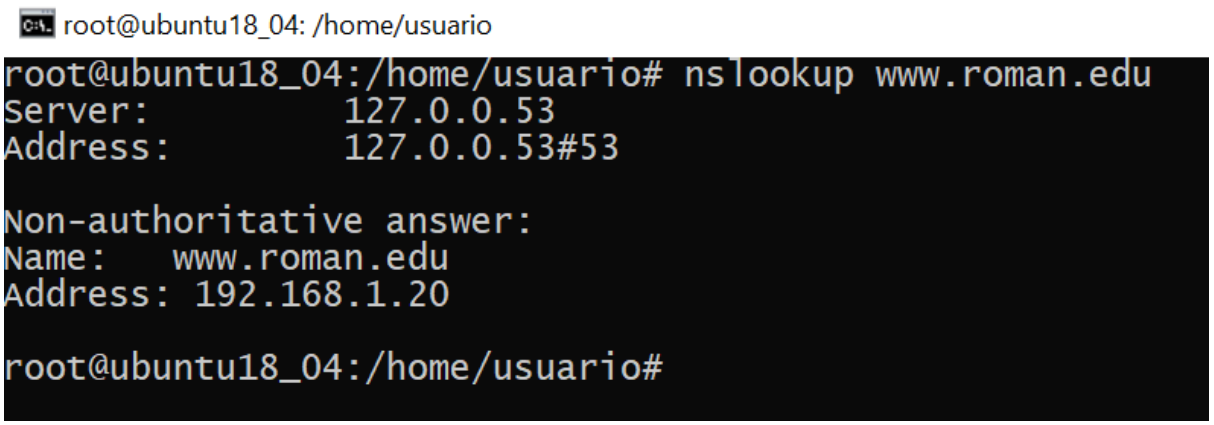
Después:



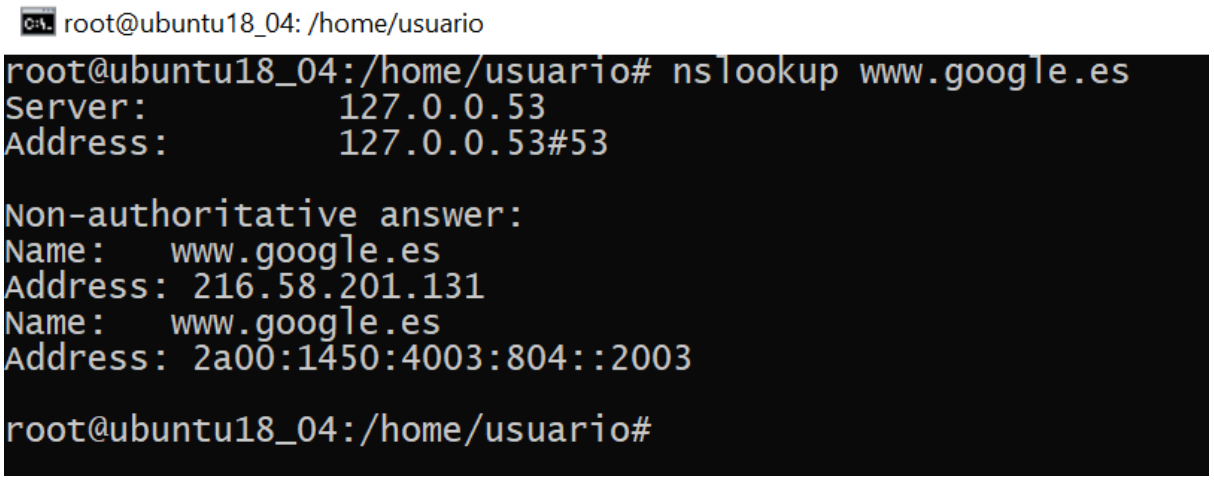
Después pulsamos CONTROL + X , pulsamos Y y después ENTER y se habrá guardado ese archivo de configuración.
Reiniciamos el servicio de red para que se aplique la nueva configuración de red: **netplan apply**



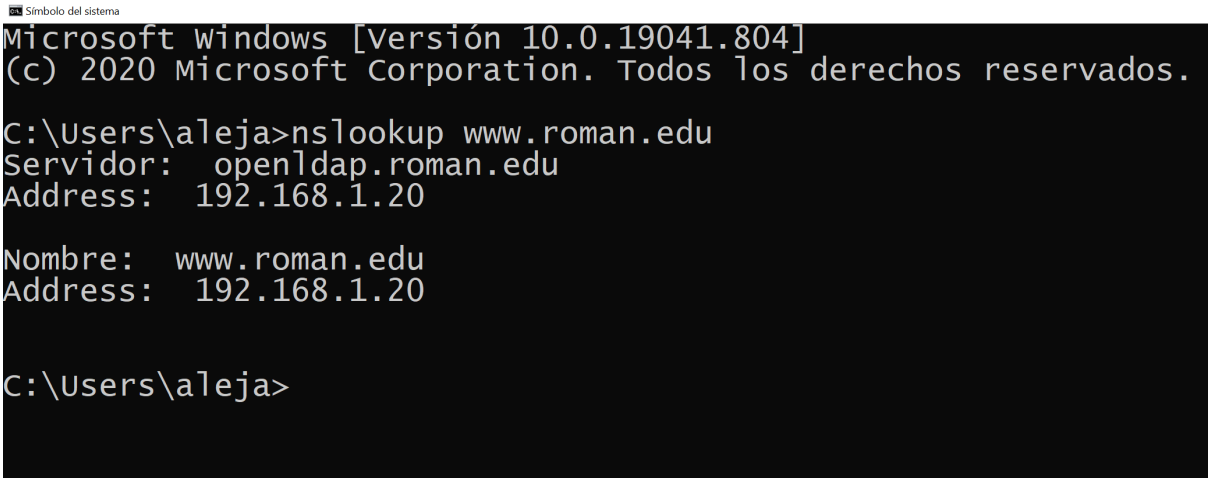
Al hacer la comprobación veremos que ya funciona el servidor dns en la máquina virtual: **nslookup www.roman.edu**



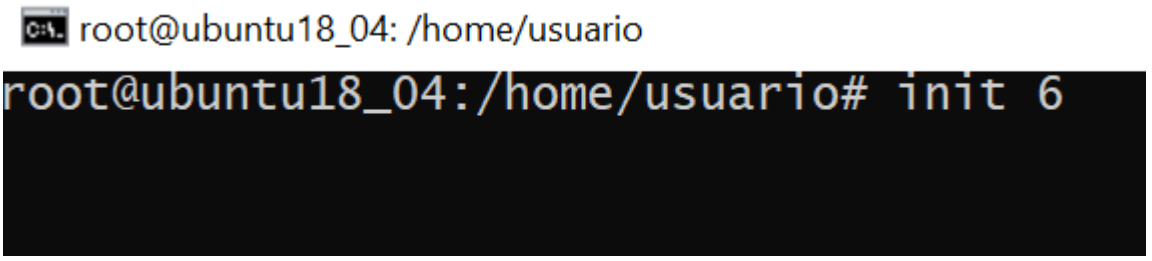
Además podemos comprobar que además podemos acceder a los dns de fuera y con eso comprobamos que está todo correctamente configurado:
nslookup www.google.es



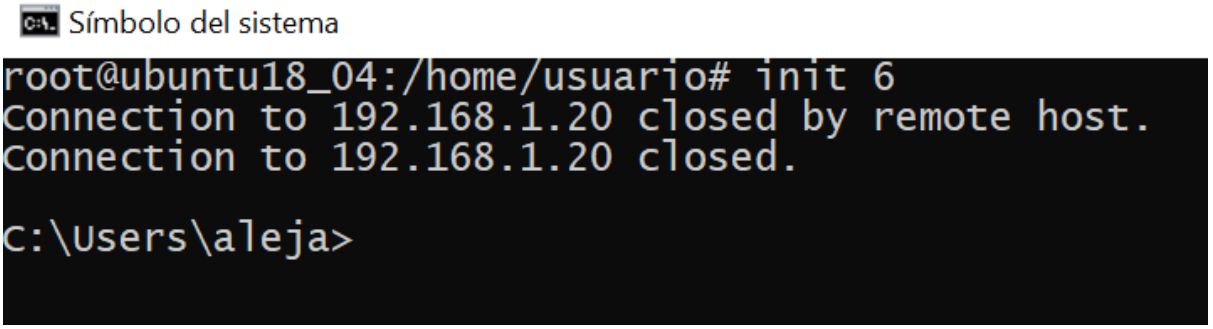
Para terminar hacemos una comprobación con la máquina real:



Por último reiniciamos la máquina virtual: **init 6**



Nos quedará una cosa así:



ALEBUNTU
No Problem

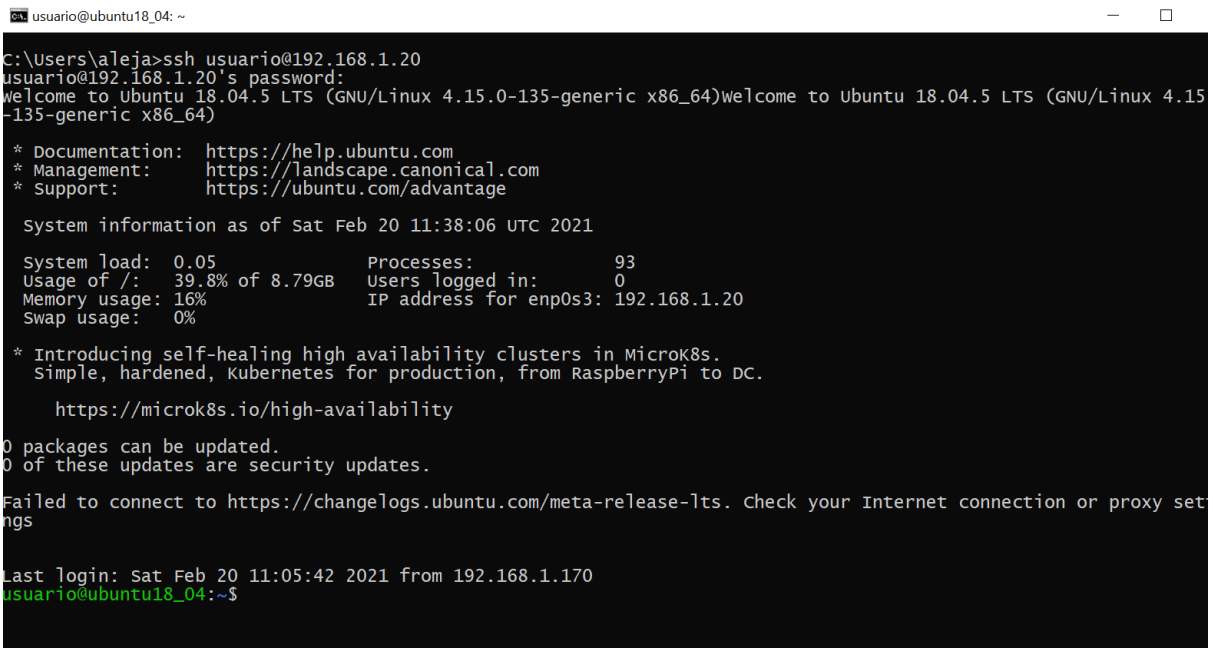
Instalacion del APACHE2

Una vez terminada la instalación y configuración del servidor DNS ya nos podemos ponernos con el servidor web de APACHE2, con el que podremos proporcionar una página web.

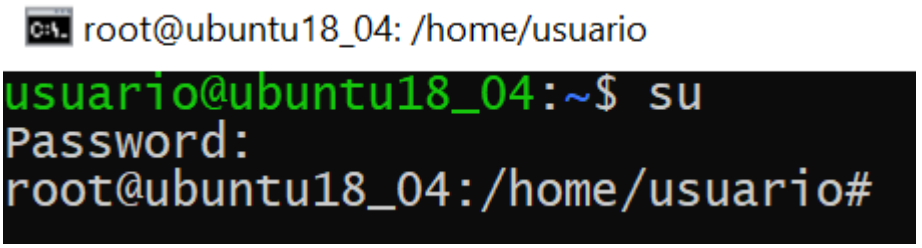
Maquina Real

Para ello metiéndonos de nuevo con la nueva ip asignada a la máquina virtual nos metemos por ssh al servidor, para ello esperamos un poco a que se reinicie la maquina virtual y comenzamos:

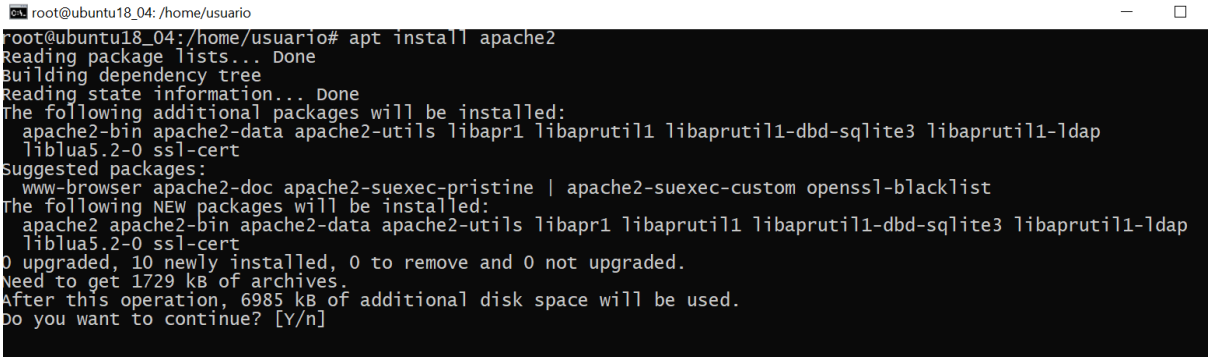
ssh usuario@192.168.1.20



Nos ponemos como root: **su**

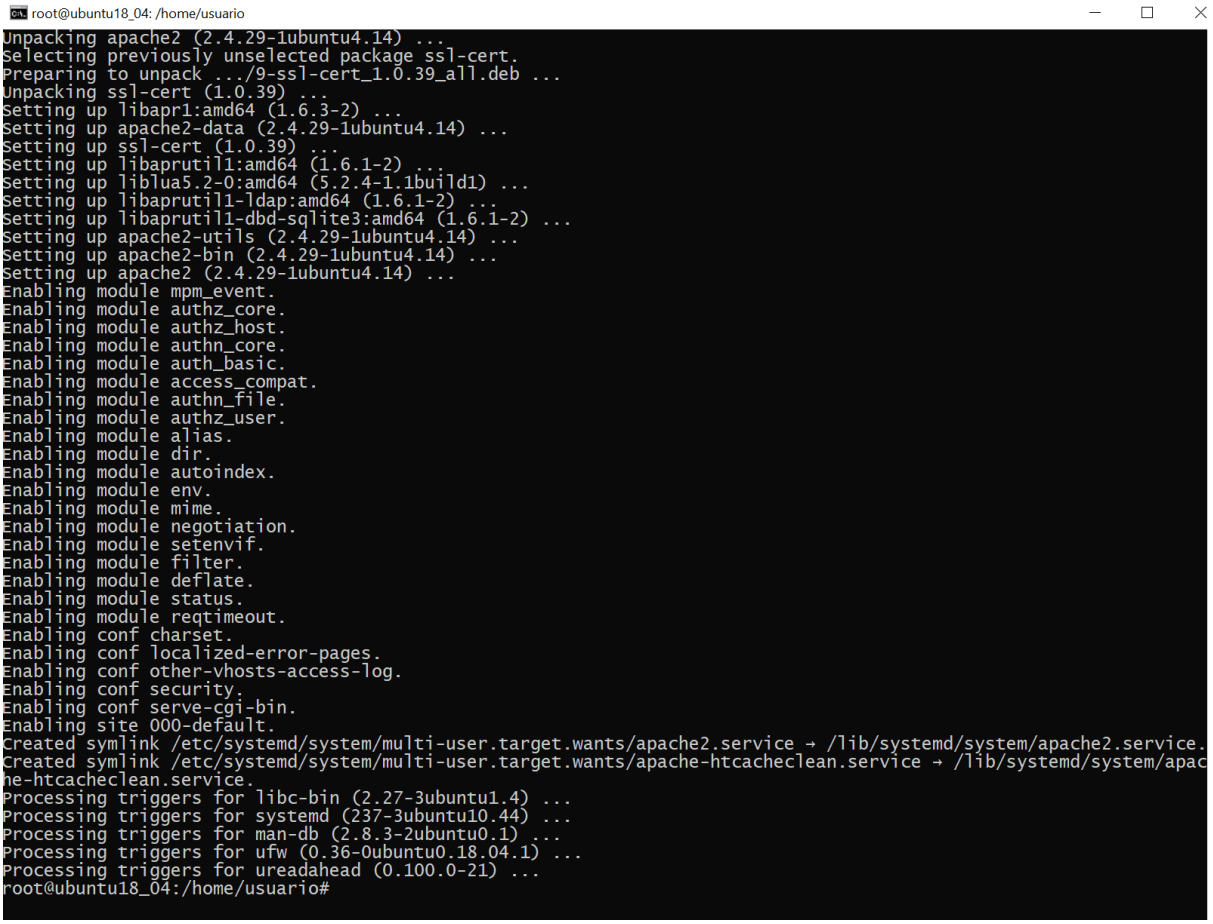


Comando de instalación: **apt install apache2**



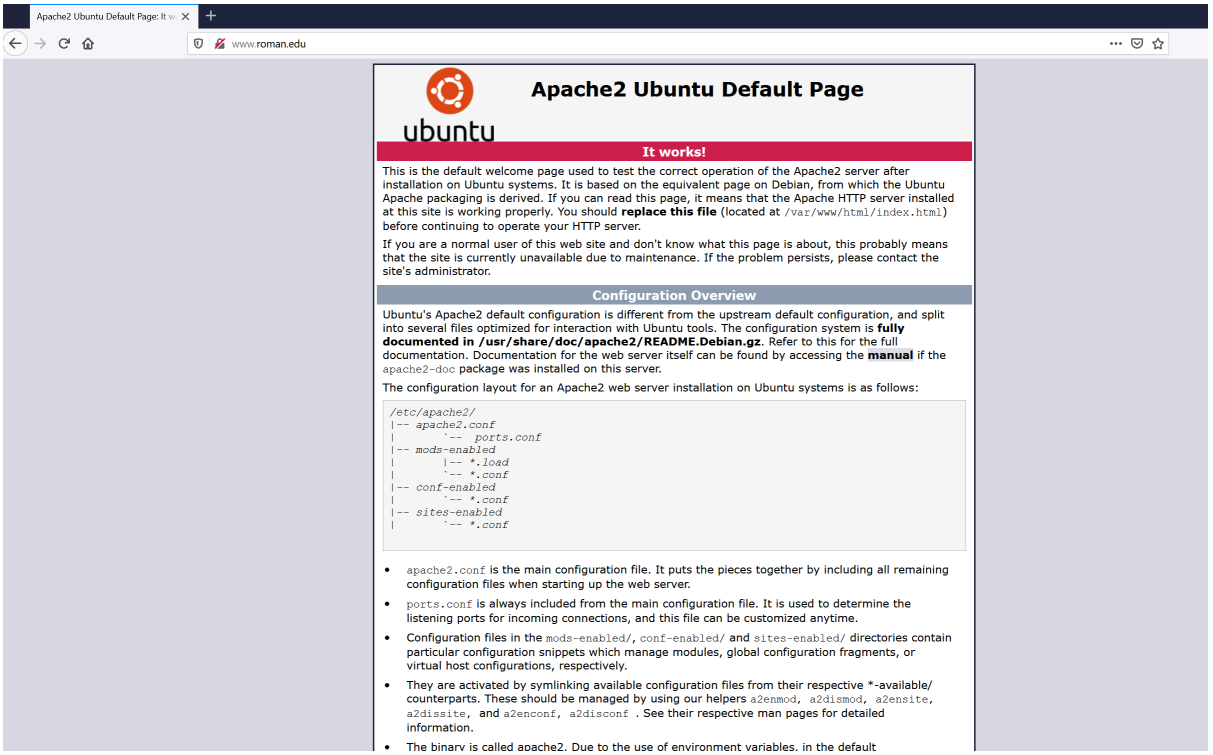
Pulsamos Y y después ENTER.

Una vez terminada la instalación nos quedará una cosa así:



Enhorabuena ya tiene una página web en un dominio ahora sí pones en cualquier navegador web de tu ordenador: http://www.roman.edu

Podrás acceder a tu propia página web servida por tu servidor web de APACHE2.



ALEBUNTU
No Problem

Prácticas con APACHE2

Vale, ya tenéis un servidor web correctamente configurado.
¿Acabamos aquí? NO

Ahora haremos una serie de prácticas sencillas para manejar APACHE2.

Serán las siguientes:

1.- Cambia la página web de APACHE2 y pon: HOLA ME LLAMO ---------.
[En los guiones ponemos nuestro nombre.]

2.- Hacer que la página web que coja el servidor APACHE2 por defecto será un archivo creado por nosotros que se llamará ***hola_mundo.html***

3.- Haz que la página web solo se pueda ver en el puerto 8080.

4.- Crear dos usuarios virtuales de APACHE2.
Uno se llamará admin y el otro se llamará usuario.

5.- Hacer que cada usuario vaya a una página web diferente.

6.- Hacer mostrar la pagina del admin haga falta una contraseña para ver la página web.

7.- Hacer que salga un mensaje de error si ponemos una ruta errónea.
Por ejemplo http://www.roman.edu/dwdedwe

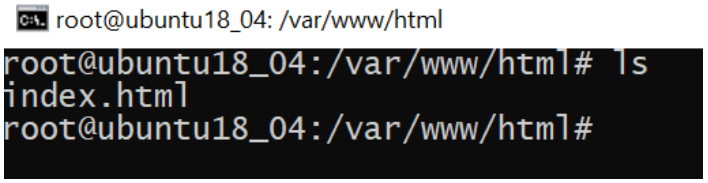
ALEBUNTU
No Problem

PRACTICA 1

Empezamos con la primera práctica, para conseguir lo que nos piden necesitamos modificar el archivo html que tiene el servidor APACHE2 por defecto en UBUNTU SERVER 18.04, que es / var/www/html/.

Primero nos vamos a la carpeta /var/www/html/: **cd /var/www/html/**

Comprobamos que está el archivo en dicha ruta: **ls**



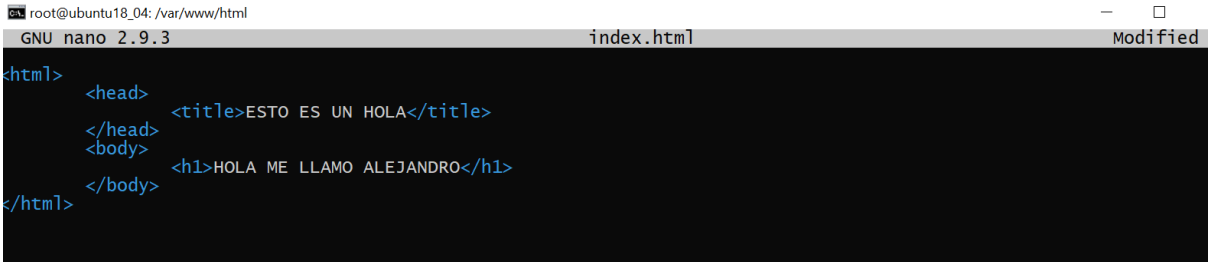
Lo vaciamos: **echo "" > index.html**

Luego hacemos una página web sencilla: **nano index.html**

Algo parecido a esto tiene que tener nuestro index:

```
<html>
    <head>
        <title>ESTO ES UN HOLA</title>
    </head>
    <body>
        <h1>HOLA ME LLAMO ALEJANDRO</h1>
    </body>
</html>
```

Y cuando lo editemos será una cosa así:



Después pulsamos CONTROL + X , pulsamos Y y después ENTER y se habrá guardado ese archivo html.

Para comprobar que ha funcionado nos vamos a un explorador web y ponermos: **http://www.roman.edu**

HOLA ME LLAMO ALEJANDRO

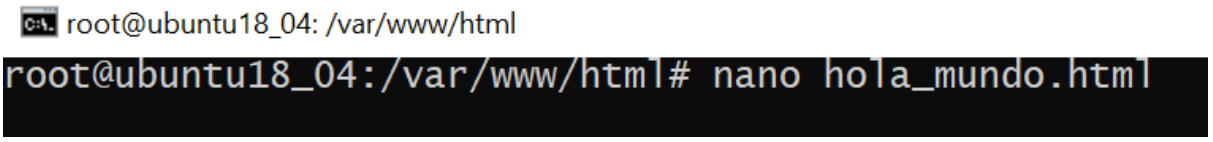
Con esto habremos terminado la primera práctica.

ALEBUNTU
No Problem

PRÁCTICA 2

Ya sabemos editar el archivo que nos proporciona la página web por defecto, pero ahora queremos que ese archivo que coja por defecto sea otro.

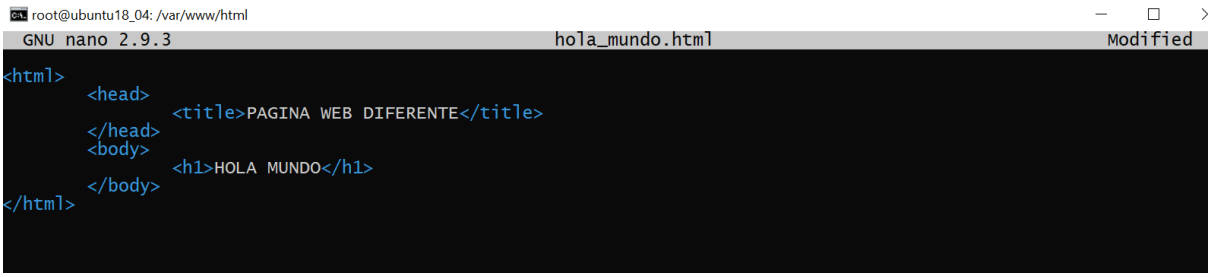
Primero creamos otra página web llamada hola_mundo.html:
nano hola_mundo.html



Algo parecido a esto tiene que tener nuestro index:

```
<html>
    <head>
        <title>PAGINA WEB DIFERENTE</title>
    </head>
    <body>
        <h1>HOLA MUNDO</h1>
    </body>
</html>
```

Y cuando lo editemos será una cosa así:



Después pulsamos CONTROL + X , pulsamos Y y después ENTER y se habrá guardado ese archivo html.

Ahora para conseguir lo que queremos en esta práctica editaremos un archivo muy importante en la configuración de APACHE2 , que es 000-default.conf y está en la siguiente ruta /etc/apache2/sites-available/.

Directamente lo editaremos poniendo el siguiente comando:
nano /etc/apache2/sites-available/000-default.conf

```
root@ubuntu18_04: /var/www/html
root@ubuntu18_04:/var/www/html# nano /etc/apache2/sites-available/000-default.conf
```

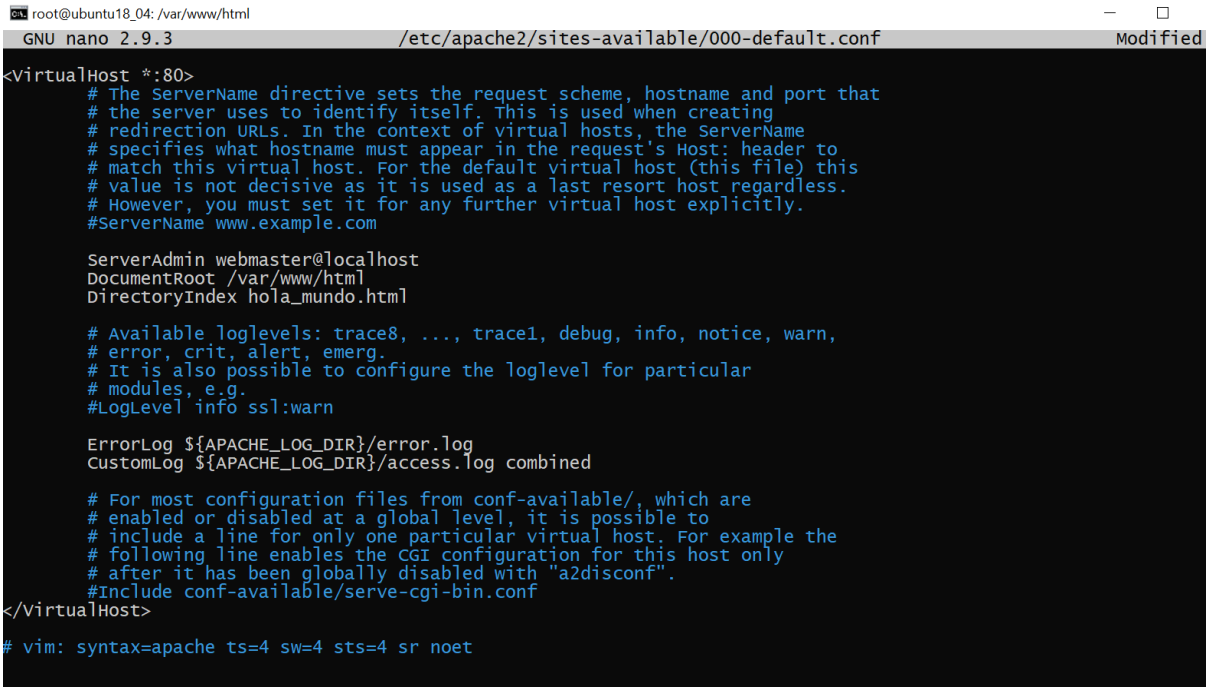
Antes:

```
root@ubuntu18_04: /var/www/html
  GNU nano 2.9.3                    /etc/apache2/sites-available/000-default.conf

<VirtualHost *:80>
        # The ServerName directive sets the request scheme, hostname and port that
        # the server uses to identify itself. This is used when creating
        # redirection URLs. In the context of virtual hosts, the ServerName
        # specifies what hostname must appear in the request's Host: header to
        # match this virtual host. For the default virtual host (this file) this
        # value is not decisive as it is used as a last resort host regardless.
        # However, you must set it for any further virtual host explicitly.
        #ServerName www.example.com

        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/html

        # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
        # error, crit, alert, emerg.
        # It is also possible to configure the loglevel for particular
        # modules, e.g.
        #LogLevel info ssl:warn

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

        # For most configuration files from conf-available/, which are
        # enabled or disabled at a global level, it is possible to
        # include a line for only one particular virtual host. For example the
        # following line enables the CGI configuration for this host only
        # after it has been globally disabled with "a2disconf".
        #Include conf-available/serve-cgi-bin.conf
</VirtualHost>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

Si queremos que el archivo html que coja por defecto no sea index , pondremos la siguiente directiva :

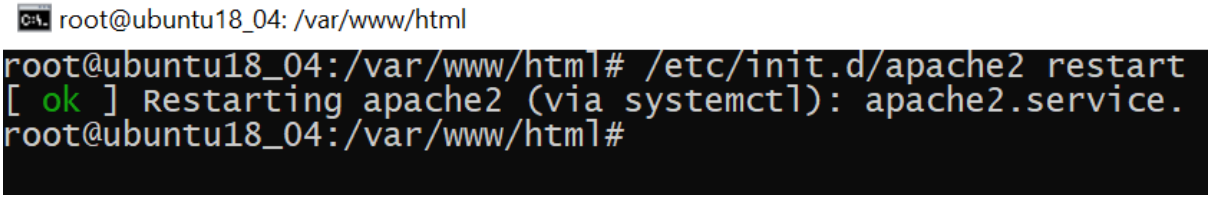
DirectoryIndex hola_mundo.html

Después:

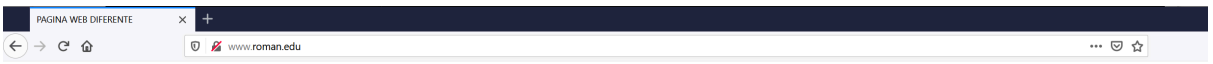


Después pulsamos CONTROL + X , pulsamos Y y después ENTER y se habrán guardado los cambios en ese archivo de configuración.

Para que se apliquen los cambios recordad reiniciar el servicio para ello hacer el siguiente comando: **/etc/init.d/apache2 restart**



Comprobamos con un explorador web poniendo: http://www.roman.edu



HOLA MUNDO

Con esto habremos terminado la segunda práctica.

ALEBUNTU
No Problem

PRÁCTICA 3

En esta práctica queremos que una página web solo se pueda ver en un determinado puerto, en nuestro caso el 8080. Para ello tocaremos 2 archivos de configuración , el ports.conf y el 000-defatult.conf .

Primero editamos el ports.conf: **nano /etc/apache2/ports.conf**

```
root@ubuntu18_04: /var/www/html
root@ubuntu18_04:/var/www/html# nano /etc/apache2/ports.conf
```

Luego añadimos en el archivo de configuración que escuche en el puerto 8080.

Antes:

```
root@ubuntu18_04: /var/www/html
  GNU nano 2.9.3                                /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>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

Después:

```
root@ubuntu18_04: /var/www/html
  GNU nano 2.9.3                        /etc/apache2/ports.conf                 Modified

# 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 8080

<IfModule ssl_module>
        Listen 443
</IfModule>

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

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

Después pulsamos CONTROL + X , pulsamos Y y después ENTER y se habrán guardado los cambios en ese archivo de configuración.

Con esto tenemos configurado que también las páginas web puedan ser vistas en el puerto 8080 si hubiese alguna.

Ahora para indicar que la páginas web sean vistas solo en el puerto 8080 editamos el 000-default.conf:
nano /etc/apache2/sites-available/000-default.conf

```
root@ubuntu18_04: /var/www/html
root@ubuntu18_04:/var/www/html# nano /etc/apache2/sites-available/000-default.conf
```

Antes:

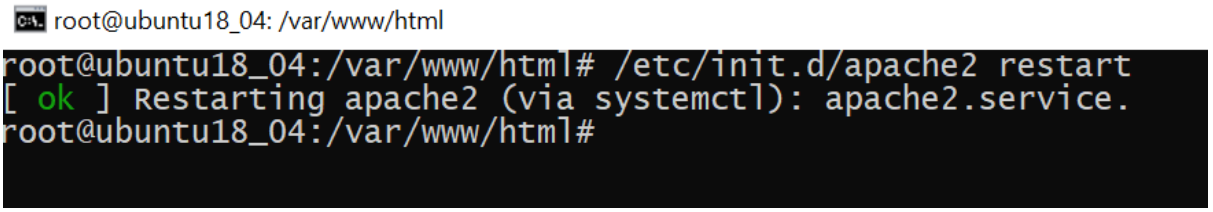
```
root@ubuntu18_04: /var/www/html
 GNU nano 2.9.3                    /etc/apache2/sites-available/000-default.conf

<VirtualHost *:80>
        # The ServerName directive sets the request scheme, hostname and port that
        # the server uses to identify itself. This is used when creating
        # redirection URLs. In the context of virtual hosts, the ServerName
        # specifies what hostname must appear in the request's Host: header to
        # match this virtual host. For the default virtual host (this file) this
        # value is not decisive as it is used as a last resort host regardless.
        # However, you must set it for any further virtual host explicitly.
        #ServerName www.example.com

        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/html
        DirectoryIndex hola_mundo.html

        # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
        # error, crit, alert, emerg.
        # It is also possible to configure the loglevel for particular
        # modules, e.g.
        #LogLevel info ssl:warn

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

        # For most configuration files from conf-available/, which are
        # enabled or disabled at a global level, it is possible to
        # include a line for only one particular virtual host. For example the
        # following line enables the CGI configuration for this host only
        # after it has been globally disabled with "a2disconf".
        #Include conf-available/serve-cgi-bin.conf
</VirtualHost>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

Después:

```
root@ubuntu18_04: /var/www/html
 GNU nano 2.9.3                    /etc/apache2/sites-available/000-default.conf                    Modified

<VirtualHost *:8080>
        # The ServerName directive sets the request scheme, hostname and port that
        # the server uses to identify itself. This is used when creating
        # redirection URLs. In the context of virtual hosts, the ServerName
        # specifies what hostname must appear in the request's Host: header to
        # match this virtual host. For the default virtual host (this file) this
        # value is not decisive as it is used as a last resort host regardless.
        # However, you must set it for any further virtual host explicitly.
        #ServerName www.example.com

        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/html
        DirectoryIndex hola_mundo.html

        # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
        # error, crit, alert, emerg.
        # It is also possible to configure the loglevel for particular
        # modules, e.g.
        #LogLevel info ssl:warn

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

        # For most configuration files from conf-available/, which are
        # enabled or disabled at a global level, it is possible to
        # include a line for only one particular virtual host. For example the
        # following line enables the CGI configuration for this host only
        # after it has been globally disabled with "a2disconf".
        #Include conf-available/serve-cgi-bin.conf
</VirtualHost>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

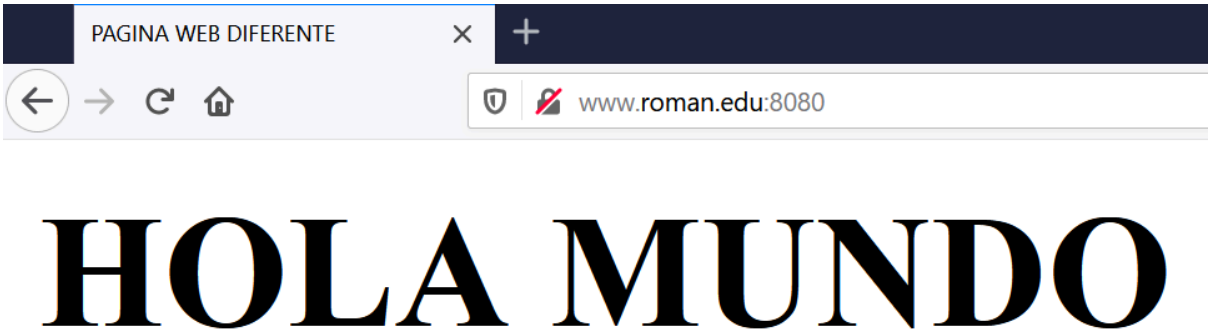
Después pulsamos CONTROL + X , pulsamos Y y después ENTER y se habrán guardado los cambios en ese archivo de configuración.

Por último reiniciamos el servicio de APACHE2: **/etc/init.d/apache2 restart**



Veremos que no podemos acceder con http://www.roman.edu

Para ello ponemos: http://www.roman.edu:8080



Con esto ya hemos finalizado la práctica 3.

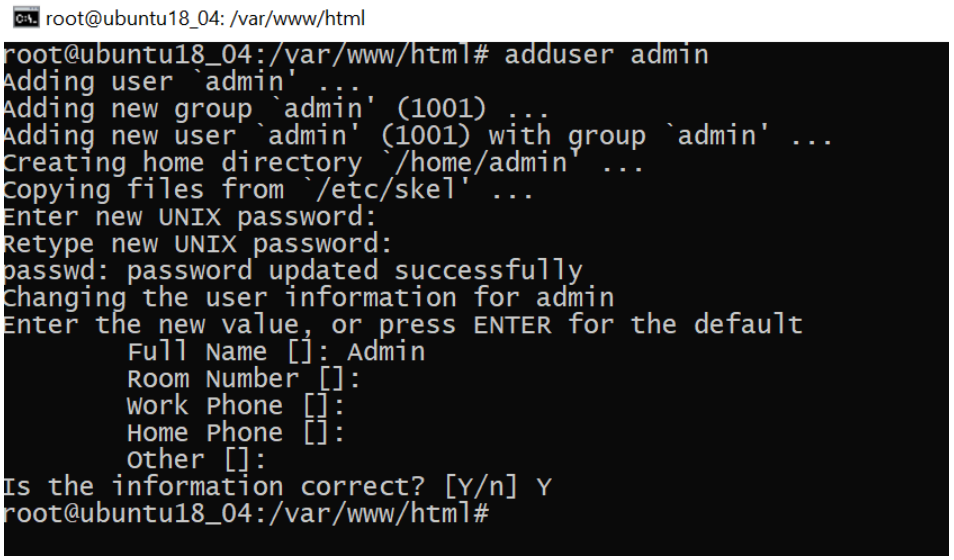
ALEBUNTU
No Problem

PRÁCTICA 4,5,6,7

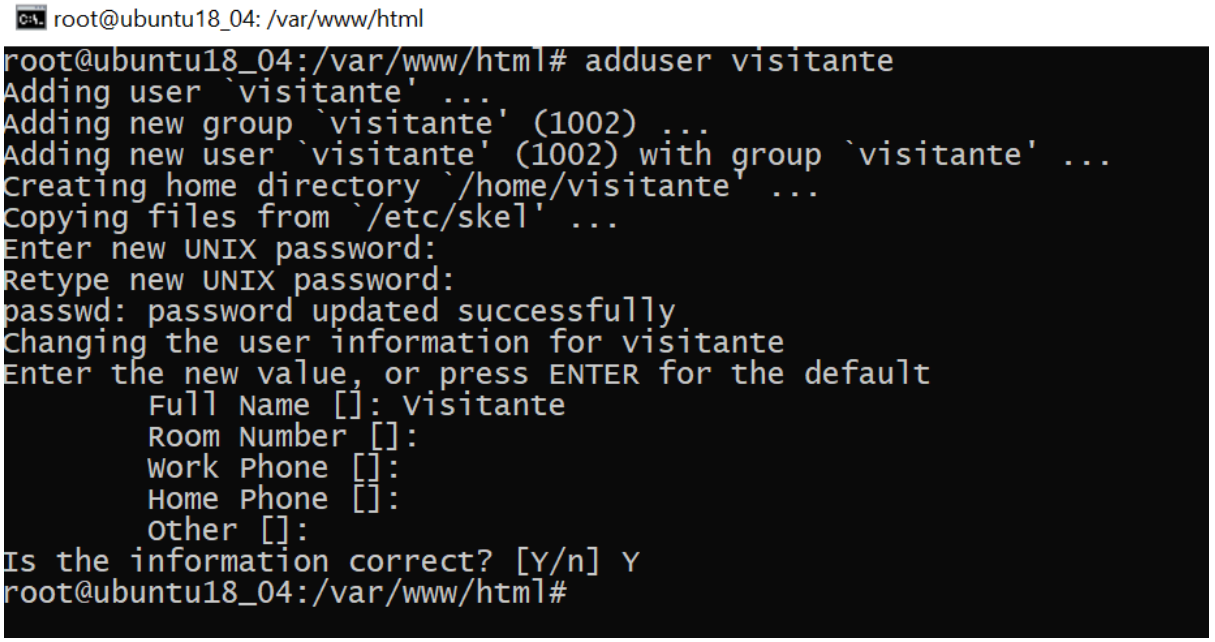
Para esta práctica crearemos dos usuarios virtuales para apache2.

Para ello crearemos los usuarios:

adduser admin

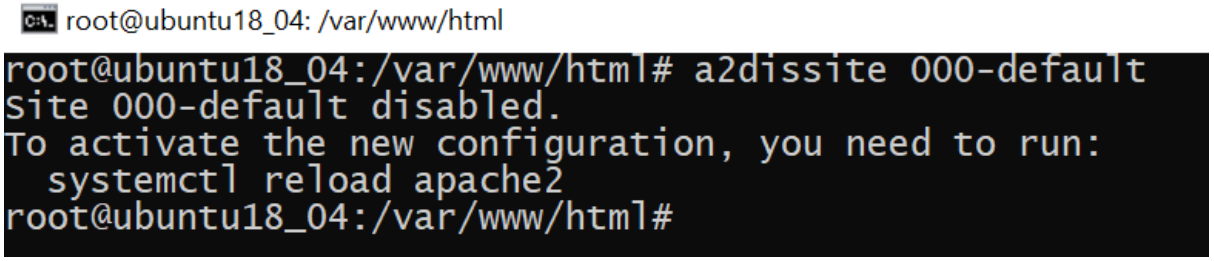


adduser visitante



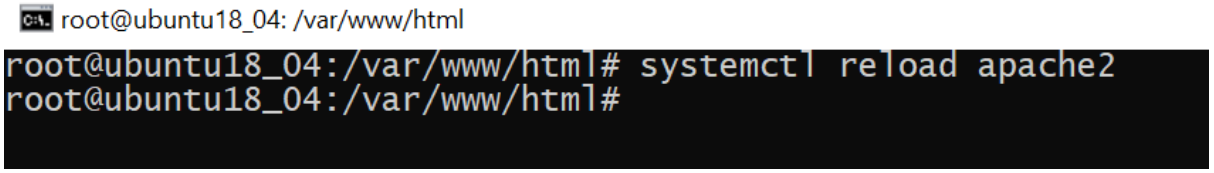
Desactivamos la configuración por defecto , o también como se le suele llamar el sitio 000-default:

a2dissite 000-default



Reiniciamos el servicio de APACHE2:

systemctl reload apache2



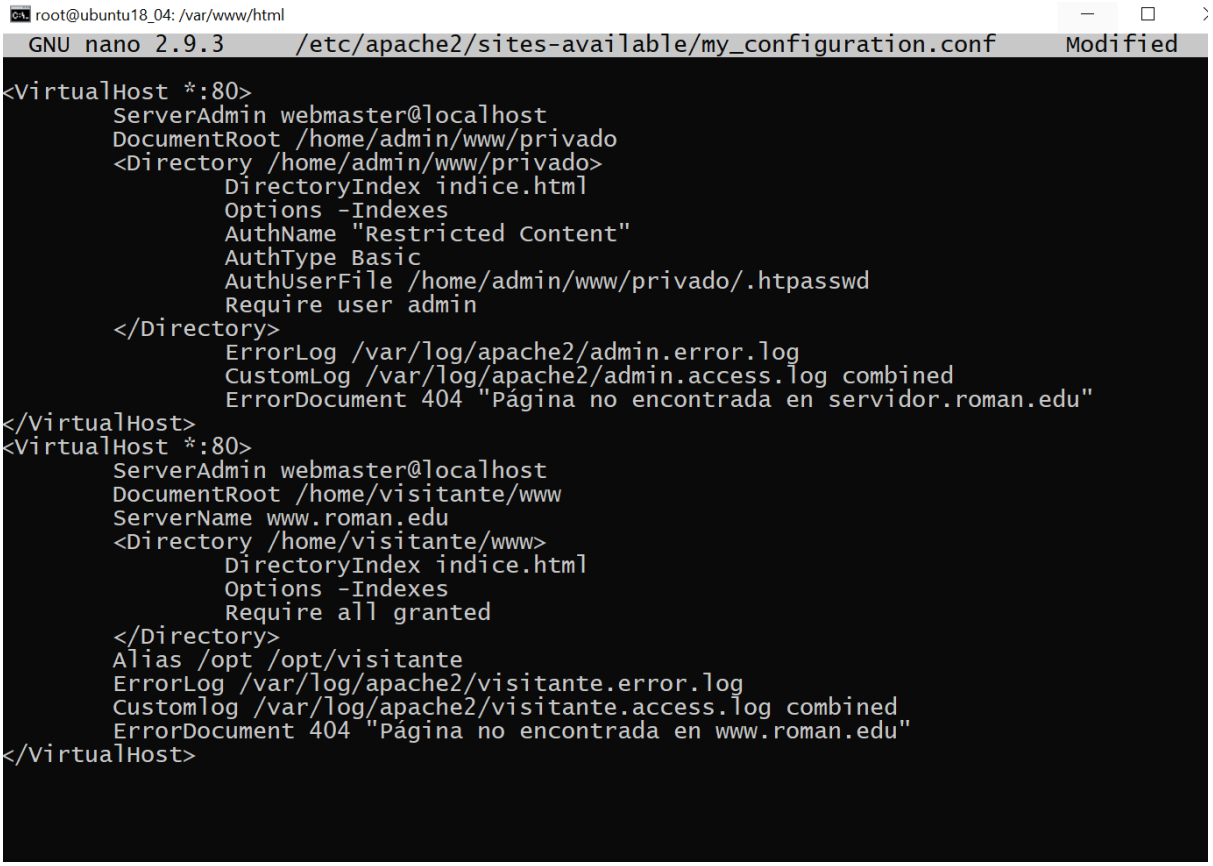
Creamos nuestra propia configuración:

nano /etc/apache2/sites-available/my_configuration.conf

Lo que debemos añadir para el usuario admin:

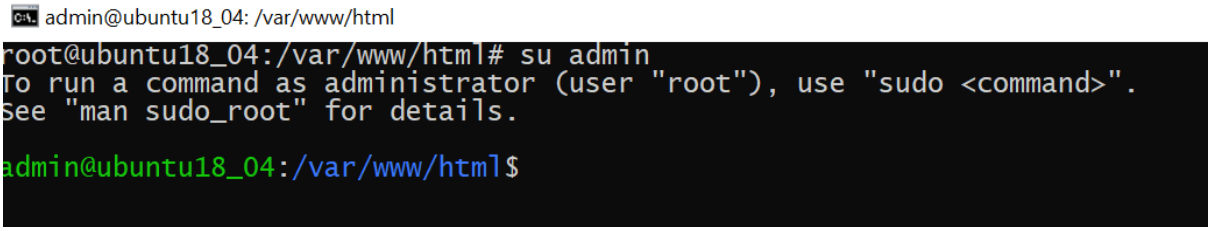
```
<VirtualHost *:80>
        ServerAdmin webmaster@localhost
        DocumentRoot /home/admin/www/privado
        <Directory /home/admin/www/privado>
                DirectoryIndex indice.html
                Options -Indexes
                AuthName "Restricted Content"
                AuthType Basic
                AuthUserFile /home/admin/www/privado/.htpasswd
                Require user admin
        </Directory>
                ErrorLog /var/log/apache2/admin.error.log
                CustomLog /var/log/apache2/admin.access.log combined
                ErrorDocument 404 "Página no encontrada en servidor.roman.edu"
</VirtualHost>
<VirtualHost *:80>
        ServerAdmin webmaster@localhost
        DocumentRoot /home/visitante/www
        ServerName www.roman.edu
        <Directory /home/visitante/www>
                DirectoryIndex indice.html
                Options -Indexes
                Require all granted
        </Directory>
        Alias /opt /opt/visitante
        ErrorLog /var/log/apache2/visitante.error.log
        Customlog /var/log/apache2/visitante.access.log combined
        ErrorDocument 404 "Página no encontrada en www.roman.edu"
</VirtualHost>
```

Se tiene que ver una cosa así:

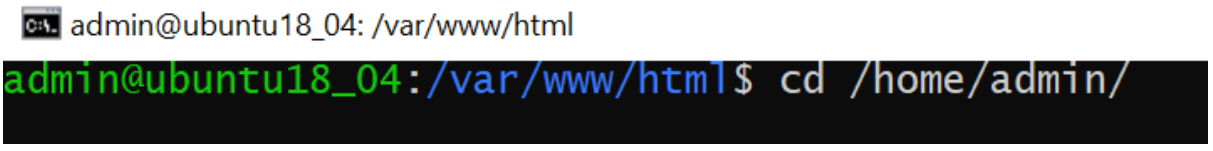


Después pulsamos CONTROL + X , pulsamos Y y después ENTER y se habrán guardado los cambios en ese archivo de configuración.

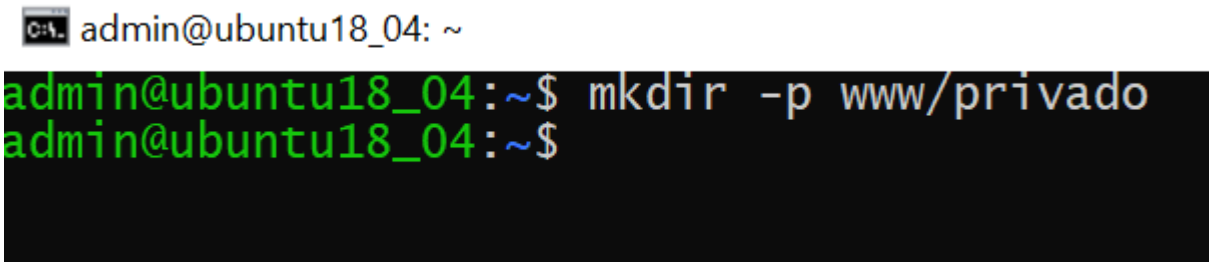
Nos ponemos como usuario admin: **su admin**



Y nos vamos a la ruta del home: **cd /home/admin/**

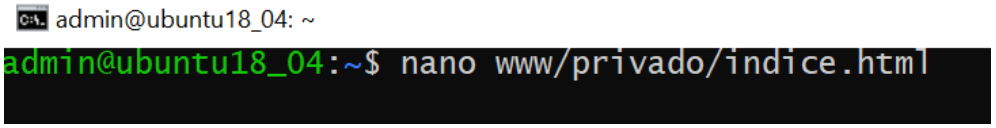


Creamos la ruta ~/www/privado/: **mkdir -p www/privado**



Creamos una página web en /www/privado/ :

nano www/privado/indice.html

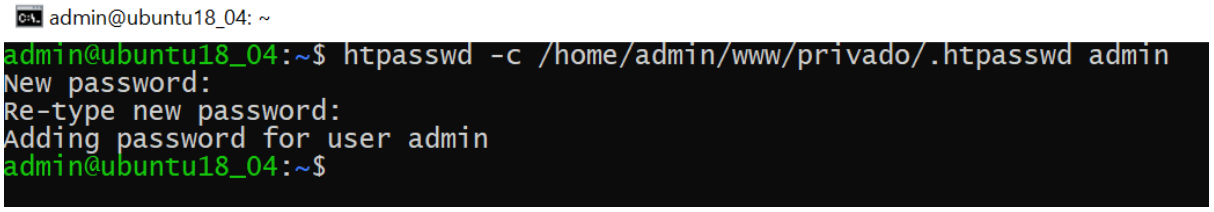


Ponemos lo siguiente:

```
<html>
    <head>
        <title>Página de admin</title>
    </head>
    <body>
        <h1>Esto es de admin</h1>
    </body>
</html>
```

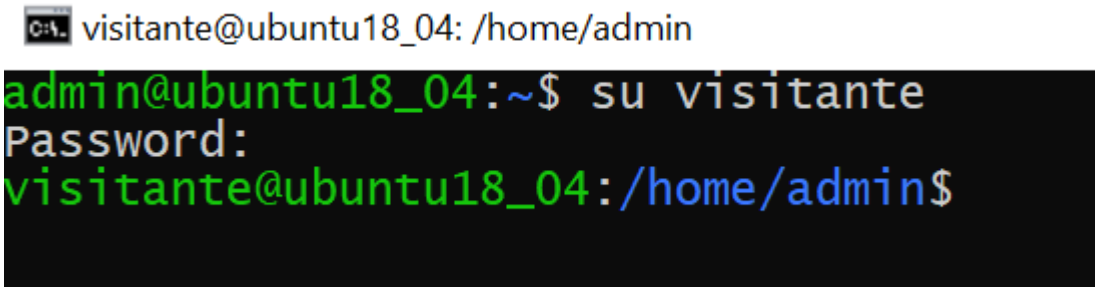
Después pulsamos CONTROL + X , pulsamos Y y después ENTER y se habrán guardado los cambios en ese archivo de html.

Por último ponemos password para el usuario virtual de apache:
htpasswd -c /home/admin/www/privado/.htpasswd admin

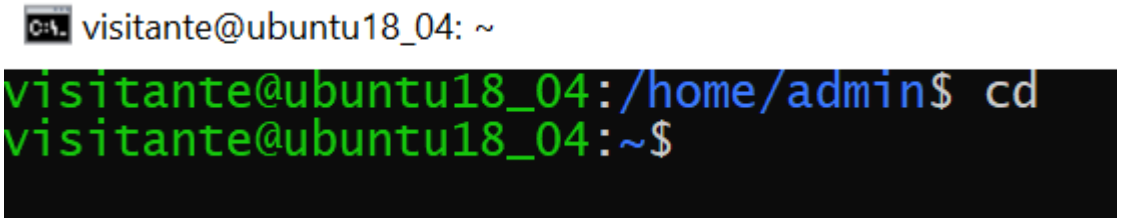


Ahora creamos las rutas pertinentes de las otras carpetas con el otro usuario.

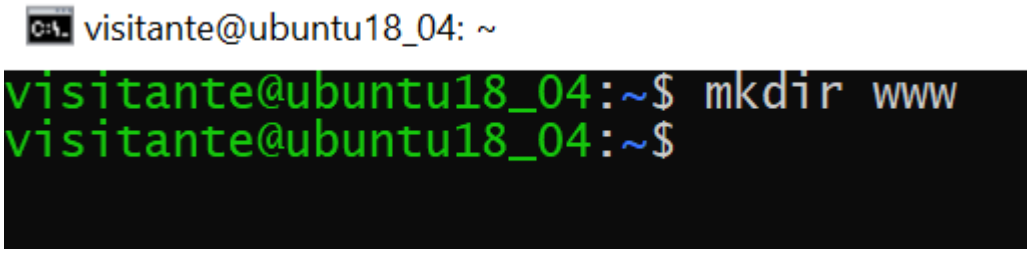
Iniciamos como usuario visitante: **su visitante**



Nos vamos a la carpeta home de visitante: **cd**

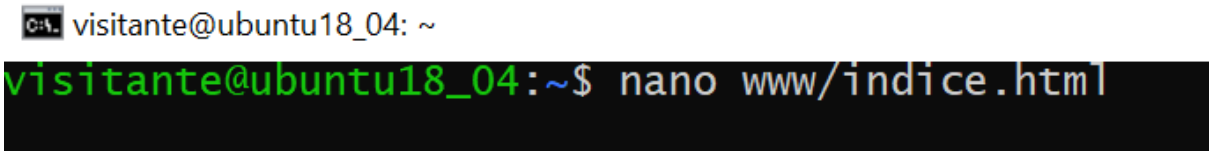


Creamos la ruta ~/www/privado/: **mkdir www**



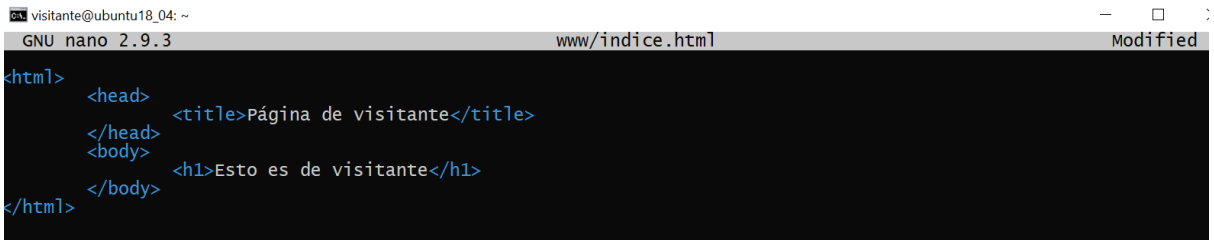
Creamos una página web en /www/privado/ :

nano www/indice.html



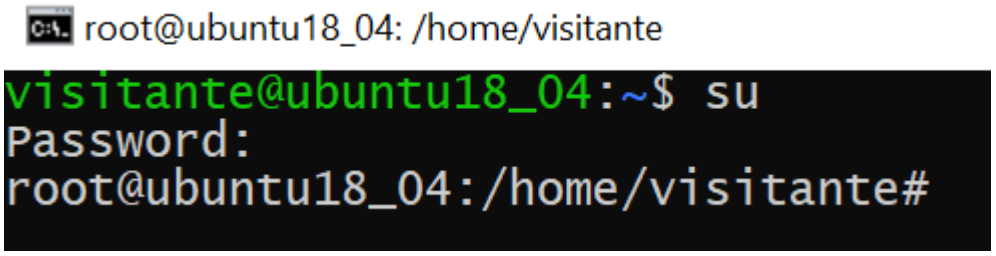
Ponemos lo siguiente:

<html>
    <head>
        <title>Página de visitante</title>
    </head>
    <body>
        <h1>Esto es de visitante</h1>
    </body>
</html>

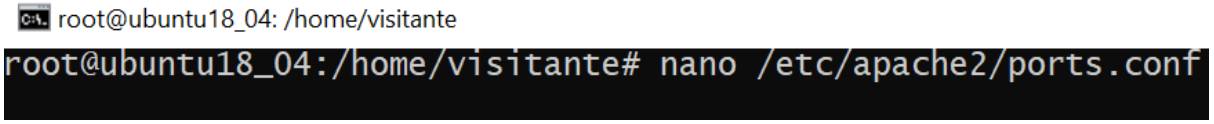


Después pulsamos CONTROL + X , pulsamos Y y después ENTER y se habrán guardado los cambios en ese archivo de html.

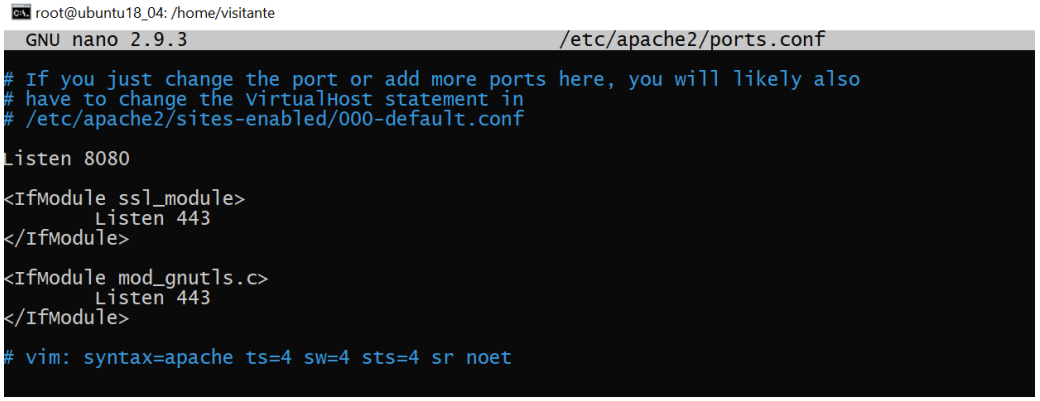
Por último volvemos como root: **su**



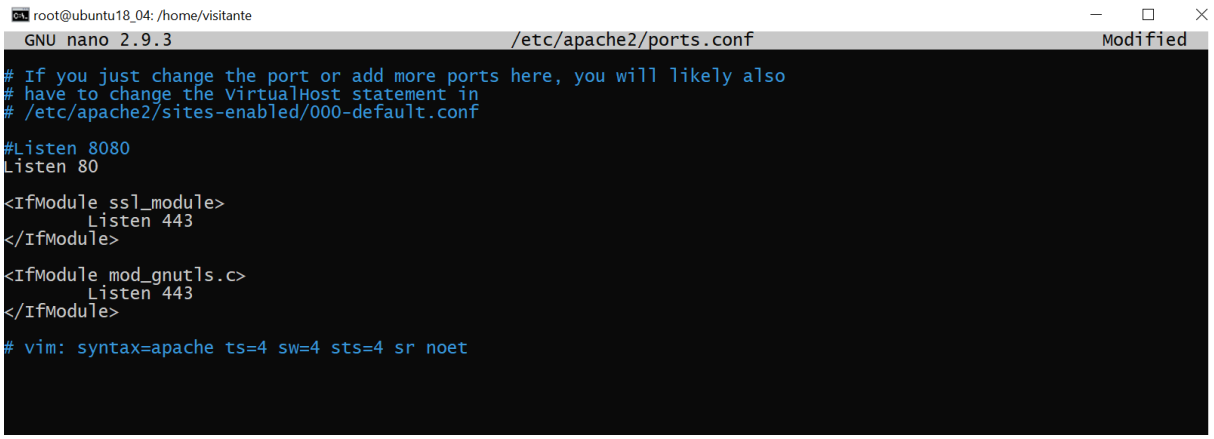
Editamos el ports.conf de APACHE2: **nano /etc/apache2/ports.conf**



Antes:



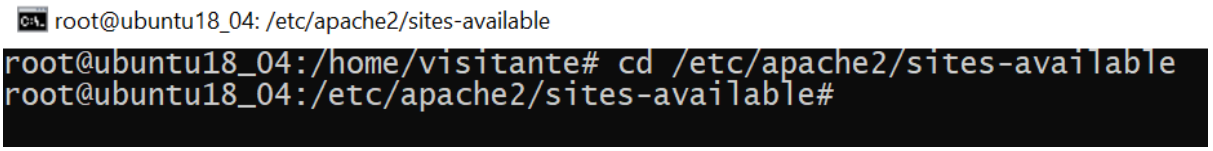
Después:



Después pulsamos CONTROL + X , pulsamos Y y después ENTER y se habrán guardado los cambios en ese archivo de configuración.

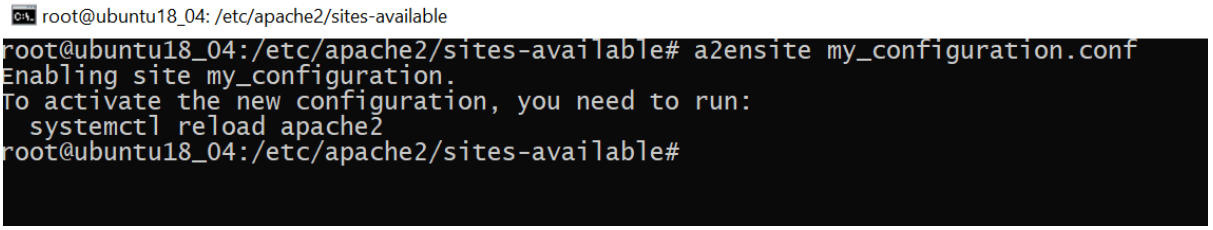
Primero nos vamos a la ruta /etc/apache2/sites-available/ :

cd /etc/apache2/sites-available



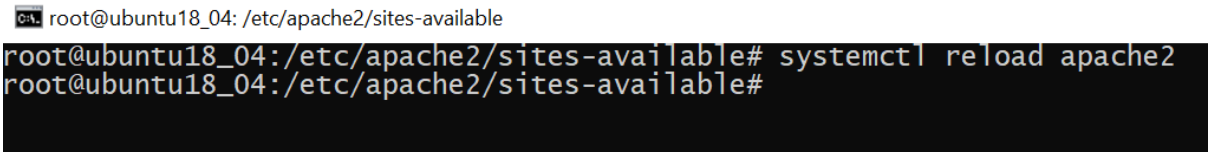
Ahora habilitamos el archivo de configuración que creamos anteriormente con el siguiente comando:

a2ensite my_configuration.conf



Por último reiniciamos el servicio con el siguiente comando:

systemctl reload apache2



Además nos faltaría activar el módulo de autenticación de APACHE2 para poder ver las páginas web de admin.

Nos vamos a donde se encuentran los módulos disponibles:
cd /etc/apache2/mods-available/

```
root@ubuntu18_04: /etc/apache2/mods-available
root@ubuntu18_04:/etc/apache2/sites-available# cd /etc/apache2/mods-available/
root@ubuntu18_04:/etc/apache2/mods-available#
```

Activamos el módulo: **a2enmod auth_basic**

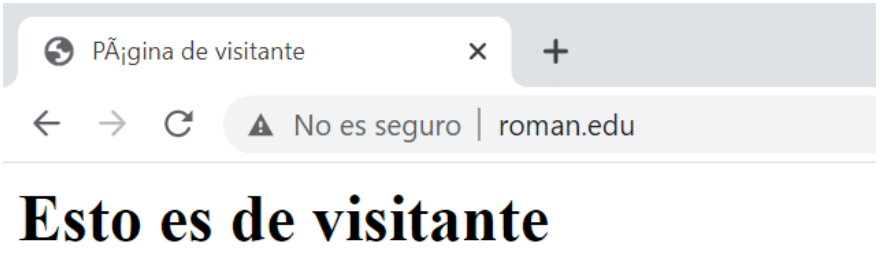
```
root@ubuntu18_04: /etc/apache2/mods-available
root@ubuntu18_04:/etc/apache2/mods-available# a2enmod auth_basic
Considering dependency authn_core for auth_basic:
Module authn_core already enabled
Module auth_basic already enabled
root@ubuntu18_04:/etc/apache2/mods-available#
```

Por último reiniciamos el servicio de APACHE2: **/etc/init.d/apache2 restart**

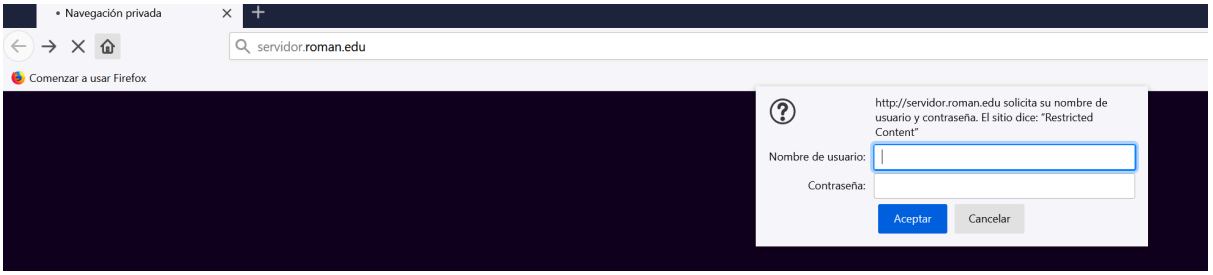
```
root@ubuntu18_04: /etc/apache2/mods-available
root@ubuntu18_04:/etc/apache2/mods-available# /etc/init.d/apache2 restart
[ ok ] Restarting apache2 (via systemctl): apache2.service.
root@ubuntu18_04:/etc/apache2/mods-available#
```

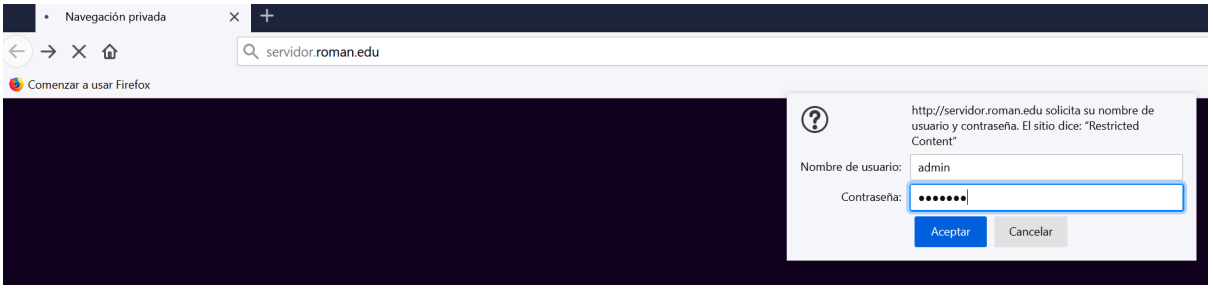
Comprobaciones

Nos sale la página del usuario visitante:

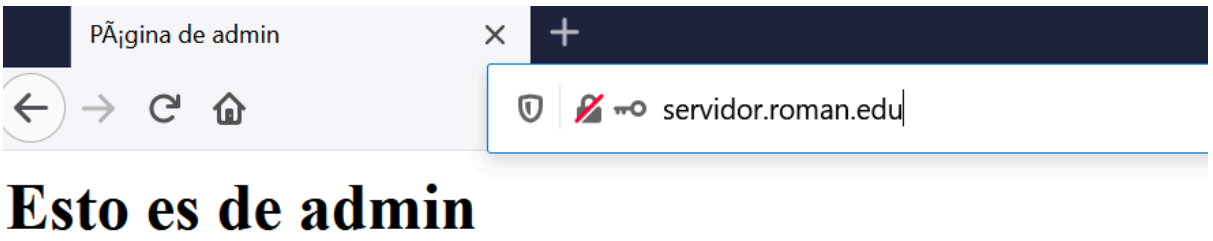


Nos pide loguearnos para ver la página de admin:

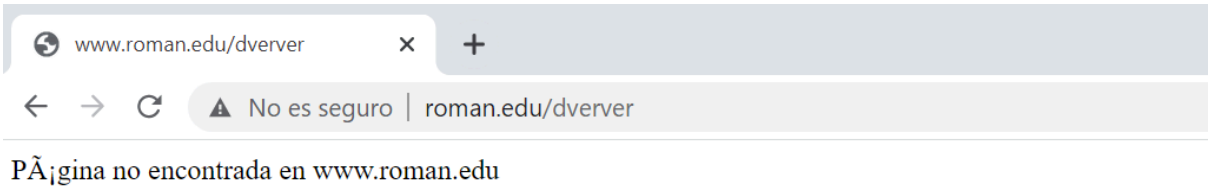




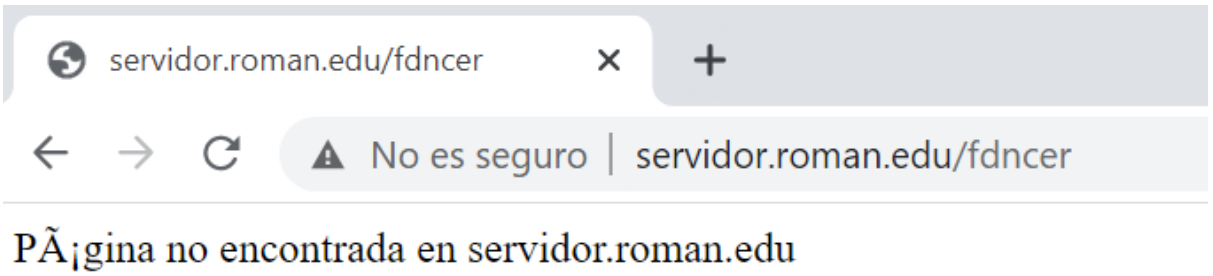
Y luego la pagina que sale le pertenece al usuario admin:



Las rutas de url que no existen de página web visitante, saldrá un mensaje específico:



Las rutas de url que no existen de página web admin, saldrá un mensaje específico:



Con esto hemos terminado las prácticas de APACHE2 , espero que os haya gustado.