

Manual de APACHE2

Revisado: 20/02/2021

Índice

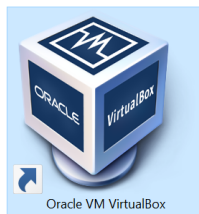
Requisitos	3
Introducción	4
Importar máquina virtual.	5
Configuración de Red Del VirtualBox	6
Servidor	7
Configuración de red	9
Máquina Virtual	9
Maquina real	10
Máquina Virtual	13
Instalación del BIND9	14
Maquina Real	14
Instalacion del APACHE2	24
Maquina Real	24
Prácticas con APACHE2	27
PRACTICA 1	28
PRÁCTICA 2	30
PRÁCTICA 3	33
PRÁCTICA 4,5,6,7	36
Comprobaciones	45

Requisitos

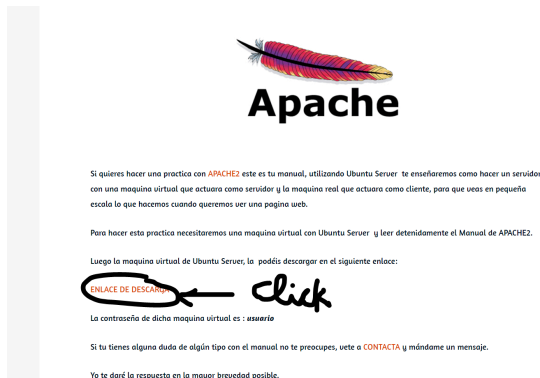
1. Tener instalada la última versión de VirtualBox.



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.



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:

← Importar servicio virtualizado

?

×

Preferencias de servicio

Estas son las máquinas virtuales contenidas en el servicio y las preferencias sugeridas de las máquinas virtuales importadas de VirtualBox. Puede cambiar varias de las propiedades mostradas haciendo doble clic en los elementos y deshabilitar otras usando las casillas de verificación de abajo.

Sistema virtual 1	
Nombre	Ubuntu-18-04-server
Tipo de SO invitado	Ubuntu (64-bit)
CPU	1
RAM	1024 MB
DVD	☒
Controlador USB	☒
Tarjeta de sonido	☒ ICH AC97

Carpeta base de máquina: D:\Maquinas_Virtuales

Política de dirección MAC: Incluir solo las direcciones NAT de adaptador de red

Opciones adicionales: ☒ Importar discos como VDI

Servicio virtualizado no firmado

Restaurar valores predeterminados Importar Cancelar

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

← Importar servicio virtualizado

Preferencias de servicio

Estas son las máquinas virtuales contenidas en el servicio y las preferencias sugeridas de las máquinas virtuales importadas de VirtualBox. Puede cambiar varias de las propiedades mostradas haciendo doble clic en los elementos y deshabilitar otras usando las casillas de verificación de abajo.

Sistema virtual 1	
Nombre	Server_Web
Tipo de SO invitado	Ubuntu (64-bit)
CPU	1
RAM	1024 MB
DVD	☒
Controlador USB	☒
Tarjeta de sonido	☒ ICH AC97

Carpeta base de máquina: D:\Maquinas_Virtuales

Política de dirección MAC: Generar nuevas direcciones MAC para todos los adaptadores de red

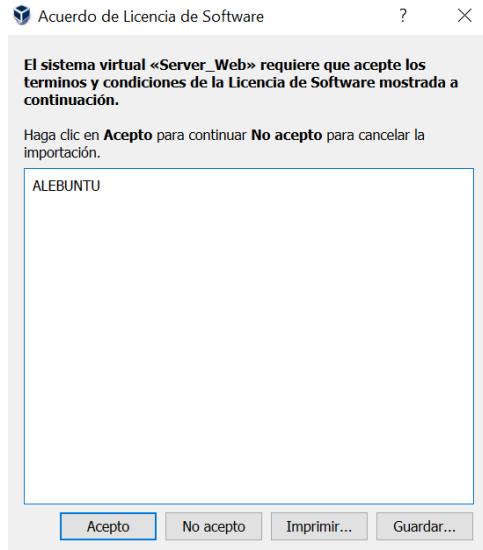
Opciones adicionales: ☒ Importar discos como VDI

Servicio virtualizado no firmado

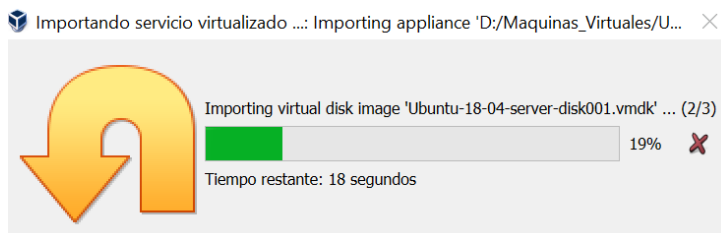
Restaurar valores predeterminados Importar Cancelar

Por último le damos a importar

Nos saldrá la siguiente imagen:

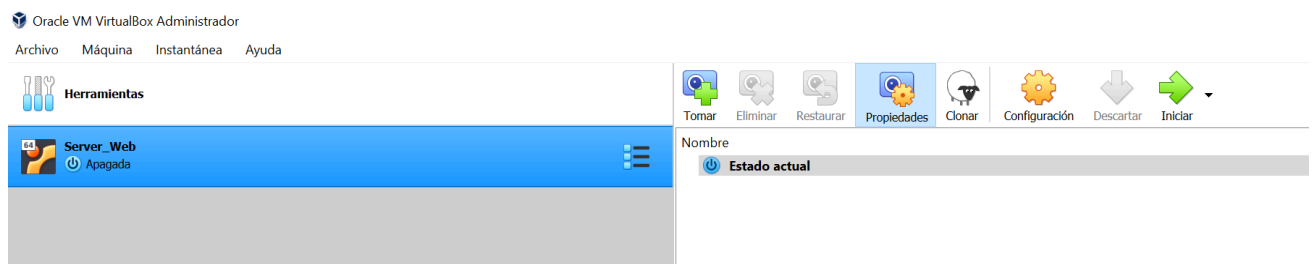


Aceptamos los acuerdos de licencia y saldrá una pantalla de carga:



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:



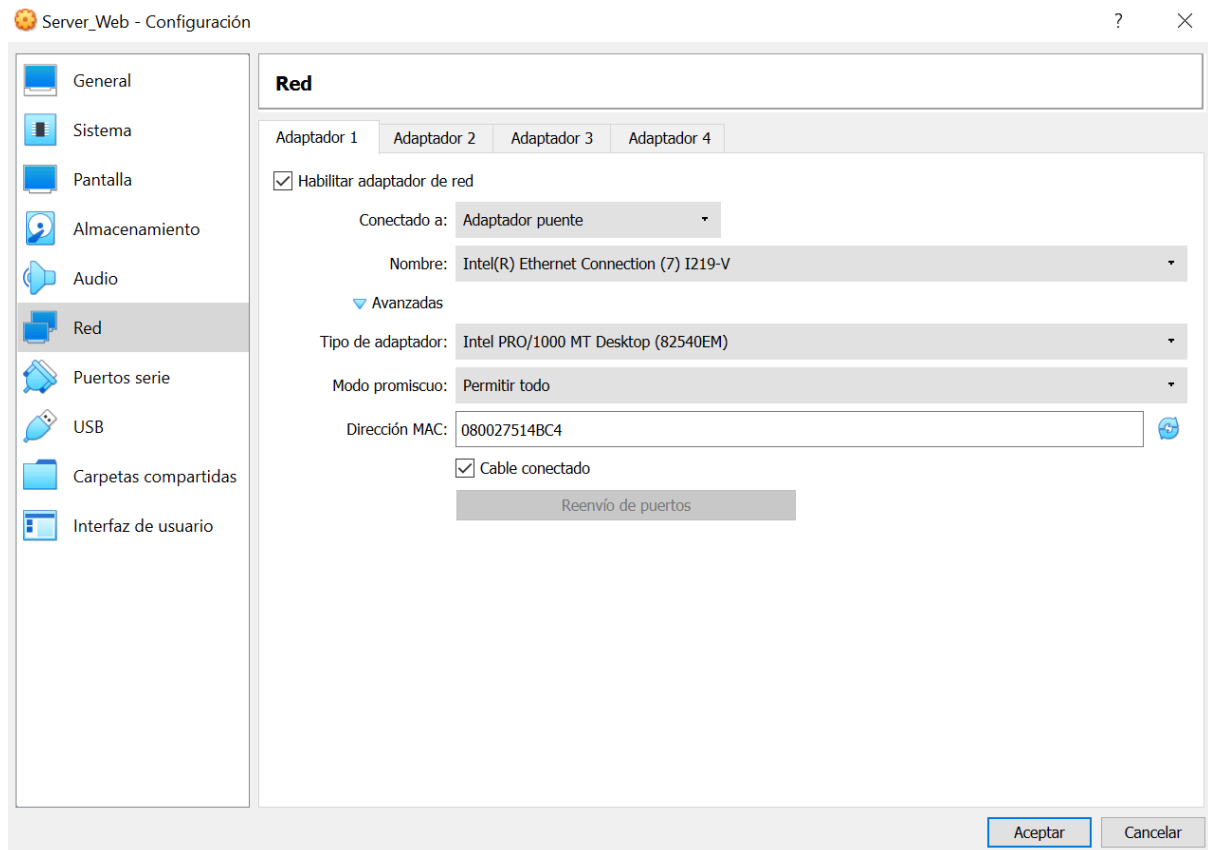
Configuración de Red Del VirtualBox

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

Servidor

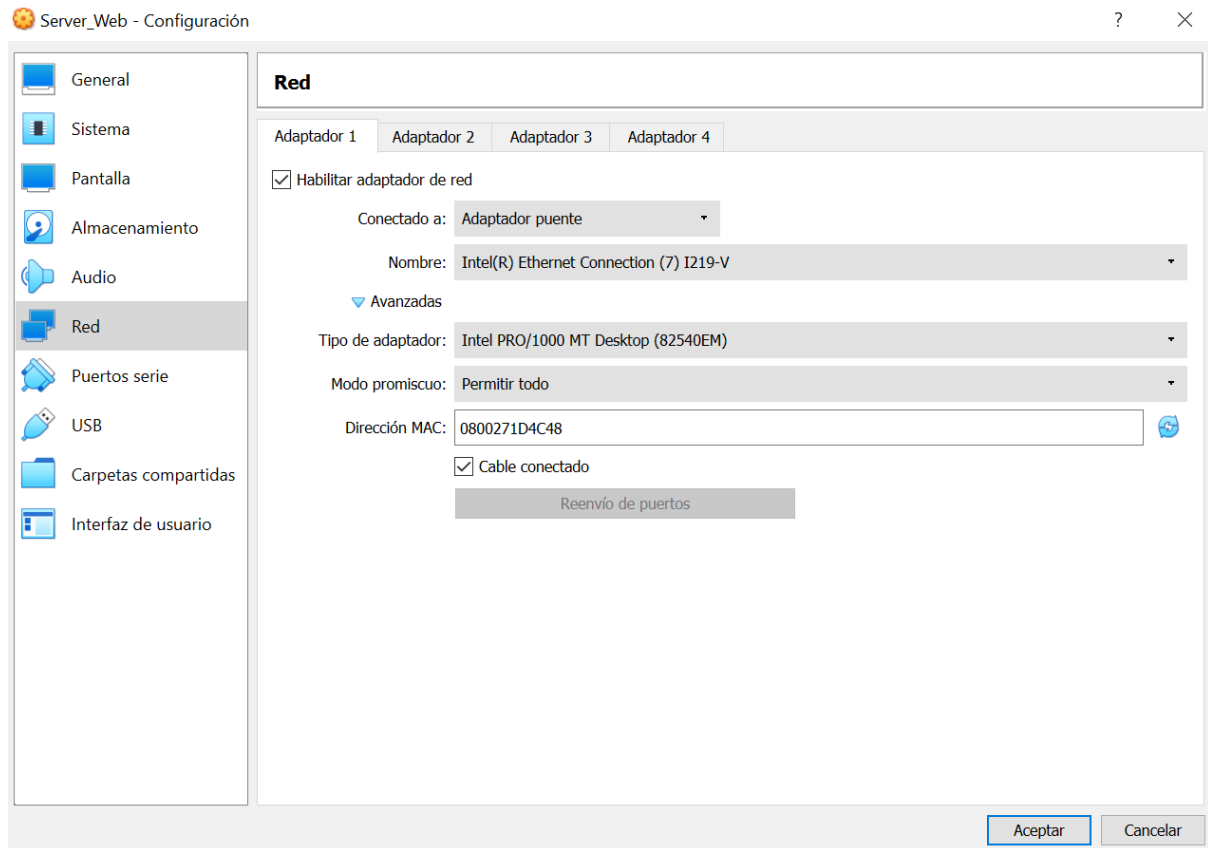
Servidor antes de iniciar la máquina virtual.

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*

 Server_Web [Corriendo] - Oracle VM VirtualBox

```
Ubuntu 18.04.5 LTS ubuntu18_04 tty1
ubuntu18_04 login: root
Password: _
```

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

 Server_Web [Corriendo] - Oracle VM VirtualBox

```
root@ubuntu18_04:~# ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.68 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::a00:27ff:fe1d:4c48 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:1d:4c:48 txqueuelen 1000 (Ethernet)
    RX packets 427 bytes 34409 (34.4 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 75 bytes 6756 (6.7 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 90 bytes 6750 (6.7 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 90 bytes 6750 (6.7 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@ubuntu18_04:~# _
```

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*.

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**

```
Microsoft Windows [Versión 10.0.19041.804]
(c) 2020 Microsoft Corporation. Todos los derechos reservados.

C:\Users\aleja>ssh usuario@192.168.1.68
The authenticity of host '192.168.1.68 (192.168.1.68)' can't be established.
ECDSA key fingerprint is SHA256:nBaTSBtgmVHfVDjOgVIPxBua9sCdkIwKNM43xvtOVI.
Are you sure you want to continue connecting (yes/no)?
```

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

```
usuario@ubuntu18_04 ~$
Memory usage: 14%          IP address for enp0s3: 192.168.1.68
Swap usage: 0%

* Introducing self-healing high availability clusters in MicroK8s.
  Simple, hardened, Kubernetes for production, from RaspberryPi to DC.

  https://microk8s.io/high-availability

0 packages can be updated.
0 of these updates are security updates.

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet
connection or proxy settings

Last login: Fri Feb 19 18:08:55 2021 from 192.168.1.170
usuario@ubuntu18_04:~$
```

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

```
root@ubuntu18_04: /home/usuario
usuario@ubuntu18_04:~$ su
Password:
root@ubuntu18_04: /home/usuario#
```

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

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

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

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

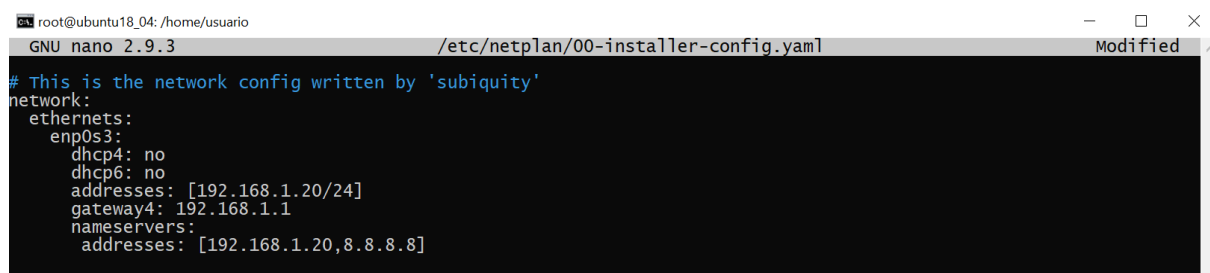
addresses: [192.168.1.20/24]

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):



```
root@ubuntu18_04: /home/usuario
GNU nano 2.9.3 /etc/netplan/00-installer-config.yaml Modified
# 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]
```

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.

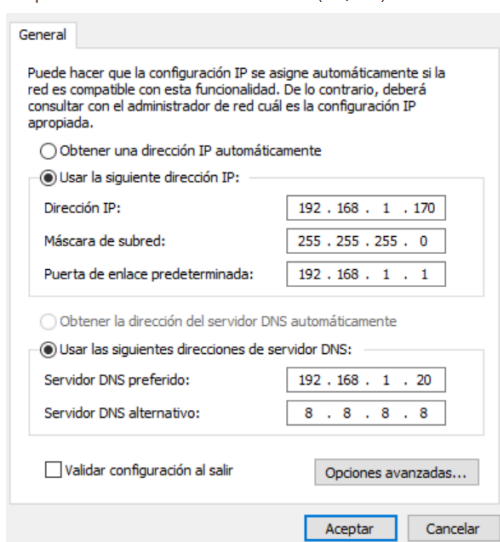
Símbolo del sistema

```
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:

Propiedades: Protocolo de Internet versión 4 (TCP/IPv4)



General

Puede hacer que la configuración IP se asigne automáticamente si la red es compatible con esta funcionalidad. De lo contrario, deberá consultar con el administrador de red cuál es la configuración IP apropiada.

☐ Obtener una dirección IP automáticamente

☒ Usar la siguiente dirección IP:

Dirección IP: 192 . 168 . 1 . 170

Máscara de subred: 255 . 255 . 255 . 0

Puerta de enlace predeterminada: 192 . 168 . 1 . 1

☐ Obtener la dirección del servidor DNS automáticamente

☒ Usar las siguientes direcciones de servidor DNS:

Servidor DNS preferido: 192 . 168 . 1 . 20

Servidor DNS alternativo: 8 . 8 . 8 . 8

☐ Validar configuración al salir

Opciones avanzadas...

Aceptar Cancelar

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**



Server_Web [Corriendo] - Oracle VM VirtualBox

```
root@ubuntu18_04:~# init 6
```

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

```
usuario@ubuntu18_04: ~  
C:\Users\aleja>ssh usuario@192.168.1.20  
usuario@192.168.1.20's password:  
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.15.0-135-generic x86_64)Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.15.0-135-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/advantage  
  
System information as of Sat Feb 20 11:05:41 UTC 2021  
  
System load:  0.68      Processes:            93  
Usage of /:   39.6% of 8.79GB   Users logged in:     0  
Memory usage: 13%          IP address for enp0s3: 192.168.1.20  
Swap usage:   0%  
  
* Introducing self-healing high availability clusters in MicroK8s.  
  Simple, hardened, kubernetes for production, from RaspberryPi to DC.  
    https://microk8s.io/high-availability  
  
0 packages can be updated.  
0 of these updates are security updates.  
  
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings  
  
Last login: Sat Feb 20 11:02:34 2021  
usuario@ubuntu18_04:~$
```

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

Comando para ser root: **su**

```
root@ubuntu18_04: /home/usuario  
usuario@ubuntu18_04:~$ su  
Password:  
root@ubuntu18_04: /home/usuario#
```

Comando de instalación: **apt install bind9**

```
root@ubuntu18_04: /home/usuario
```

```
root@ubuntu18_04:/home/usuario# apt install bind9
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  bind9utils python3-ply
Suggested packages:
  bind9-doc resolvconf python-ply-doc
The following NEW packages will be installed:
  bind9 bind9utils python3-ply
0 upgraded, 3 newly installed, 0 to remove and 0 not upgraded.
Need to get 659 kB of archives.
After this operation, 3554 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Pulsamos Y y después ENTER.

Una vez terminada la instalación nos quedará una cosa así:

```
Do you want to continue? [Y/n] Y
Get:1 http://es.archive.ubuntu.com/ubuntu bionic/main amd64 python3-ply all 3.11-1 [46.6 kB]
Get:2 http://es.archive.ubuntu.com/ubuntu bionic-updates/main amd64 bind9utils amd64 1:9.11.3+dfsg-1ubuntu1.14 [216 kB]
Get:3 http://es.archive.ubuntu.com/ubuntu bionic-updates/main amd64 bind9 amd64 1:9.11.3+dfsg-1ubuntu1.14 [397 kB]
Fetched 659 kB in 1s (646 kB/s)
Preconfiguring packages ...
Selecting previously unselected package python3-ply.
(Reading database ... 67128 files and directories currently installed.)
Preparing to unpack .../python3-ply_3.11-1_all.deb ...
Unpacking python3-ply (3.11-1) ...
Selecting previously unselected package bind9utils.
Preparing to unpack .../bind9utils_1%3a9.11.3+dfsg-1ubuntu1.14_amd64.deb ...
Unpacking bind9utils (1:9.11.3+dfsg-1ubuntu1.14) ...
Selecting previously unselected package bind9.
Preparing to unpack .../bind9_1%3a9.11.3+dfsg-1ubuntu1.14_amd64.deb ...
Unpacking bind9 (1:9.11.3+dfsg-1ubuntu1.14) ...
Setting up python3-ply (3.11-1) ...
Setting up bind9utils (1:9.11.3+dfsg-1ubuntu1.14) ...
Setting up bind9 (1:9.11.3+dfsg-1ubuntu1.14) ...
Adding group 'bind' (GID 113) ...
Done.
Adding system user 'bind' (UID 111) ...
Adding new user 'bind' (UID 111) with group 'bind' ...
Not creating home directory '/var/cache/bind'.
wrote key file "/etc/bind/rndc.key"
Created symlink /etc/systemd/system/multi-user.target.wants/bind9.service -> /lib/systemd/system/bind9.service.
bind9-pkcs11.service is a disabled or a static unit, not starting it.
bind9-resolvconf.service is a disabled or a static unit, not starting it.
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for ufw (0.36-0ubuntu0.18.04.1) ...
Processing triggers for ureadahead (0.100.0-21) ...
Processing triggers for systemd (237-3ubuntu10.44) ...
root@ubuntu18_04:/home/usuario#
```

Comando para introducir la zona directa:

nano /etc/bind/db.directa

```
root@ubuntu18_04: /home/usuario
```

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

```
root@ubuntu18_04: /home/usuario
GNU nano 2.9.3 /etc/bind/db.directa Modified
$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.
```

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

```
root@ubuntu18_04: /home/usuario
```

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

```
root@ubuntu18_04: /home/usuario
GNU nano 2.9.3 /etc/bind/db.inversa Modified
$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.
```

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

```
root@ubuntu18_04: /home/usuario
```

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

```
root@ubuntu18_04: /home/usuario  
GNU nano 2.9.3 /etc/bind/named.conf.local Modified  
//  
// Do any local configuration here  
//  
// Consider adding the 1918 zones here, if they are not used in your  
// organization  
//include "/etc/bind/zones.rfc1918";  
zone "roman.edu" {  
    type master;  
    file "/etc/bind/db.directa";  
};  
  
zone "1.168.192.in-addr.arpa" {  
    type master;  
    file "/etc/bind/db.inversa";  
};
```

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

```
root@ubuntu18_04: /home/usuario
```

```
root@ubuntu18_04:/home/usuario# /etc/init.d/bind9 restart  
[ ok ] Restarting bind9 (via systemctl): bind9.service.  
root@ubuntu18_04:/home/usuario#
```

Si hacemos la comprobación veremos que esta fallida:

nslookup www.roman.edu

```
root@ubuntu18_04: /home/usuario
```

```
root@ubuntu18_04:/home/usuario# nslookup www.roman.edu
Server:                127.0.0.53
Address:                127.0.0.53#53

** server can't find www.roman.edu: NXDOMAIN

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

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:

```
root@ubuntu18_04: /home/usuario
```

```
GNU nano 2.9.3 /etc/netplan/00-installer-config.yaml

# 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]
```

Después:

```
root@ubuntu18_04: /home/usuario
```

```
GNU nano 2.9.3 /etc/netplan/00-installer-config.yaml Modified

# 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]
```

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**

```
root@ubuntu18_04: /home/usuario
```

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

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

```
root@ubuntu18_04: /home/usuario
```

```
root@ubuntu18_04:/home/usuario# nslookup www.roman.edu
Server:           127.0.0.53
Address:          127.0.0.53#53

Non-authoritative answer:
Name:   www.roman.edu
Address: 192.168.1.20

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

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

```
root@ubuntu18_04: /home/usuario
```

```
root@ubuntu18_04:/home/usuario# nslookup www.google.es
Server:           127.0.0.53
Address:          127.0.0.53#53

Non-authoritative answer:
Name:   www.google.es
Address: 216.58.201.131
Name:   www.google.es
Address: 2a00:1450:4003:804::2003

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

Para terminar hacemos una comprobación con la máquina real:

 Símbolo del sistema

```
Microsoft Windows [Versión 10.0.19041.804]
(c) 2020 Microsoft Corporation. Todos los derechos reservados.

C:\Users\aleja>nslookup www.roman.edu
Servidor:  openldap.roman.edu
Address:  192.168.1.20

Nombre:  www.roman.edu
Address:  192.168.1.20

C:\Users\aleja>
```

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



root@ubuntu18_04: /home/usuario

```
root@ubuntu18_04:/home/usuario# init 6
```

Nos quedará una cosa así:



Símbolo del sistema

```
root@ubuntu18_04:/home/usuario# init 6
Connection to 192.168.1.20 closed by remote host.
Connection to 192.168.1.20 closed.

C:\Users\aleja>
```

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

```
usuario@ubuntu18_04: ~  
C:\Users\aleja>ssh usuario@192.168.1.20  
usuario@192.168.1.20's password:  
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.15.0-135-generic x86_64)welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.15-135-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/advantage  
  
System information as of Sat Feb 20 11:38:06 UTC 2021  
  
System load:  0.05          Processes:           93  
Usage of /:   39.8% of 8.79GB    Users logged in:    0  
Memory usage: 16%           IP address for enp0s3: 192.168.1.20  
Swap usage:   0%  
  
* Introducing self-healing high availability clusters in MicroK8s.  
  Simple, hardened, Kubernetes for production, from RaspberryPi to DC.  
  
  https://microk8s.io/high-availability  
  
0 packages can be updated.  
0 of these updates are security updates.  
  
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings  
  
Last login: Sat Feb 20 11:05:42 2021 from 192.168.1.170  
usuario@ubuntu18_04:~$
```

Nos ponemos como root: **su**

```
root@ubuntu18_04: /home/usuario  
  
usuario@ubuntu18_04:~$ su  
Password:  
root@ubuntu18_04: /home/usuario#
```


Comando de instalación: **apt install apache2**

```
root@ubuntu18_04: /home/usuario
root@ubuntu18_04:/home/usuario# apt install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap
  liblua5.2-0 ssl-cert
Suggested packages:
  www-browser apache2-doc apache2-suexec-pristine | apache2-suexec-custom openssl-blacklist
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap
  liblua5.2-0 ssl-cert
0 upgraded, 10 newly installed, 0 to remove and 0 not upgraded.
Need to get 1729 kB of archives.
After this operation, 6985 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Pulsamos Y y después ENTER.

Una vez terminada la instalación nos quedará una cosa así:

```
root@ubuntu18_04: /home/usuario
Unpacking apache2 (2.4.29-1ubuntu4.14) ...
Selecting previously unselected package ssl-cert.
Preparing to unpack .../9-ssl-cert_1.0.39_all.deb ...
Unpacking ssl-cert (1.0.39) ...
Setting up libapr1:amd64 (1.6.3-2) ...
Setting up apache2-data (2.4.29-1ubuntu4.14) ...
Setting up ssl-cert (1.0.39) ...
Setting up libaprutil1:amd64 (1.6.1-2) ...
Setting up liblua5.2-0:amd64 (5.2.4-1.1build1) ...
Setting up libaprutil1-ldap:amd64 (1.6.1-2) ...
Setting up libaprutil1-dbd-sqlite3:amd64 (1.6.1-2) ...
Setting up apache2-utils (2.4.29-1ubuntu4.14) ...
Setting up apache2-bin (2.4.29-1ubuntu4.14) ...
Setting up apache2 (2.4.29-1ubuntu4.14) ...
Enabling module mpm_event.
Enabling module authz_core.
Enabling module authz_host.
Enabling module authn_core.
Enabling module auth_basic.
Enabling module access_compat.
Enabling module authn_file.
Enabling module authz_user.
Enabling module alias.
Enabling module dir.
Enabling module autoindex.
Enabling module env.
Enabling module mime.
Enabling module negotiation.
Enabling module setenvif.
Enabling module filter.
Enabling module deflate.
Enabling module status.
Enabling module reqtimeout.
Enabling conf charset.
Enabling conf localized-error-pages.
Enabling conf other-vhosts-access-log.
Enabling conf security.
Enabling conf serve-cgi-bin.
Enabling site 000-default.
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /lib/systemd/system/apache2.service.
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.service → /lib/systemd/system/apache-htcacheclean.service.
Processing triggers for libc-bin (2.27-3ubuntu1.4) ...
Processing triggers for systemd (237-3ubuntu10.44) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for ufw (0.36-0ubuntu0.18.04.1) ...
Processing triggers for ureadahead (0.100.0-21) ...
root@ubuntu18_04: /home/usuario#
```

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.



The screenshot shows a web browser window with the address bar displaying `www.roman.edu`. The page content is the "Apache2 Ubuntu Default Page". It features the Ubuntu logo and the text "It works!". Below this, there is a paragraph explaining that this is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It mentions that the configuration system is fully documented in `/usr/share/doc/apache2/README.Debian.gz`. A section titled "Configuration Overview" follows, providing a detailed explanation of the configuration files and their organization. A code block shows the contents of `/etc/apache2/apache2.conf`, which includes comments and directives for loading modules and enabling configuration files. Finally, a bulleted list summarizes the key configuration files and their roles.

Apache2 Ubuntu Default Page

ubuntu

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|   |-- mods-enabled
|       |-- *.load
|       |-- *.conf
|   |-- conf-enabled
|       |-- *.conf
|   |-- sites-enabled
|       |-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective `*-available/` counterparts. These should be managed by using our helpers `a2enmod`, `a2dismod`, `a2ensite`, `a2dissite`, and `a2enconf`, `a2disconf`. See their respective `man` pages for detailed information.
- The binary is called `apache2`. Due to the use of environment variables, in the default

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>

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/**

```
root@ubuntu18_04: /var/www/html
```

```
root@ubuntu18_04:/home/usuario# cd /var/www/html/  
root@ubuntu18_04:/var/www/html#
```

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

```
root@ubuntu18_04: /var/www/html
```

```
root@ubuntu18_04:/var/www/html# ls  
index.html  
root@ubuntu18_04:/var/www/html#
```

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

```
root@ubuntu18_04: /var/www/html
```

```
root@ubuntu18_04:/var/www/html# echo "" > index.html  
root@ubuntu18_04:/var/www/html#
```

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

```
root@ubuntu18_04: /var/www/html
```

```
root@ubuntu18_04:/var/www/html# 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:

```
root@ubuntu18_04: /var/www/html
GNU nano 2.9.3 index.html Modified
<html>
  <head>
    <title>ESTO ES UN HOLA</title>
  </head>
  <body>
    <h1>HOLA ME LLAMO ALEJANDRO</h1>
  </body>
</html>
```

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 ponemos:
<http://www.roman.edu>



Con esto habremos terminado la primera práctica.

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

```
root@ubuntu18_04: /var/www/html
root@ubuntu18_04:/var/www/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:

```
root@ubuntu18_04: /var/www/html
GNU nano 2.9.3 hola_mundo.html Modified
<html>
  <head>
    <title>PAGINA WEB DIFERENTE</title>
  </head>
  <body>
    <h1>HOLA MUNDO</h1>
  </body>
</html>
```

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:

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

```
root@ubuntu18_04: /var/www/html
root@ubuntu18_04:/var/www/html# /etc/init.d/apache2 restart
[ ok ] Restarting apache2 (via systemctl): apache2.service.
root@ubuntu18_04:/var/www/html#
```

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



PAGINA WEB DIFERENTE x +
← → ↻ 🏠 www.roman.edu ... 📄 ☆
HOLA MUNDO

Con esto habremos terminado la segunda práctica.

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-default.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**

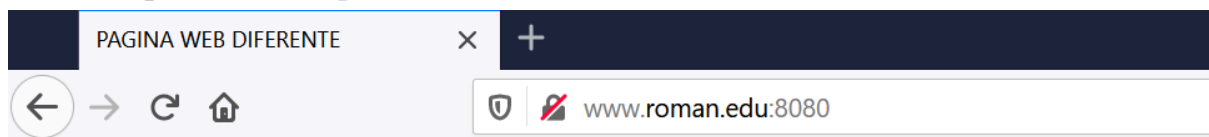
```
root@ubuntu18_04: /var/www/html
```

```
root@ubuntu18_04:/var/www/html# /etc/init.d/apache2 restart
[ ok ] Restarting apache2 (via systemctl): apache2.service.
root@ubuntu18_04:/var/www/html#
```

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



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



HOLA MUNDO

Con esto ya hemos finalizado la práctica 3.

PRÁCTICA 4,5,6,7

Para esta práctica crearemos dos usuarios virtuales para apache2.

Para ello crearemos los usuarios:

adduser admin

```
root@ubuntu18_04: /var/www/html
root@ubuntu18_04:/var/www/html# adduser admin
Adding user `admin' ...
Adding new group `admin' (1001) ...
Adding new user `admin' (1001) with group `admin' ...
Creating home directory `/home/admin' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for admin
Enter the new value, or press ENTER for the default
  Full Name []: Admin
   Room Number []:
   Work Phone []:
   Home Phone []:
    Other []:
Is the information correct? [Y/n] Y
root@ubuntu18_04:/var/www/html#
```

adduser visitante

```
root@ubuntu18_04: /var/www/html
root@ubuntu18_04:/var/www/html# adduser visitante
Adding user `visitante' ...
Adding new group `visitante' (1002) ...
Adding new user `visitante' (1002) with group `visitante' ...
Creating home directory `/home/visitante' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for visitante
Enter the new value, or press ENTER for the default
  Full Name []: Visitante
   Room Number []:
   Work Phone []:
   Home Phone []:
    Other []:
Is the information correct? [Y/n] Y
root@ubuntu18_04:/var/www/html#
```

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

a2dissite 000-default

```
root@ubuntu18_04: /var/www/html
```

```
root@ubuntu18_04:/var/www/html# a2dissite 000-default
Site 000-default disabled.
To activate the new configuration, you need to run:
  systemctl reload apache2
root@ubuntu18_04:/var/www/html#
```

Reiniciamos el servicio de APACHE2:

systemctl reload apache2

```
root@ubuntu18_04: /var/www/html
```

```
root@ubuntu18_04:/var/www/html# systemctl reload apache2
root@ubuntu18_04:/var/www/html#
```

Creamos nuestra propia configuración:

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

```
root@ubuntu18_04: /var/www/html
```

```
root@ubuntu18_04:/var/www/html# 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:

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

<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>
```

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**

```
admin@ubuntu18_04: /var/www/html
root@ubuntu18_04:/var/www/html# su admin
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
admin@ubuntu18_04:/var/www/html$
```

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

```
admin@ubuntu18_04: /var/www/html
admin@ubuntu18_04:/var/www/html$ cd /home/admin/
```

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

```
admin@ubuntu18_04: ~
```

```
admin@ubuntu18_04:~$ mkdir -p www/privado
admin@ubuntu18_04:~$
```

Creamos una página web en /www/privado/ :

nano www/privado/indice.html

```
admin@ubuntu18_04: ~
```

```
admin@ubuntu18_04:~$ 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>
```

```
admin@ubuntu18_04: ~
GNU nano 2.9.3      www/privado/indice.html      Modified
<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

```
admin@ubuntu18_04: ~
```

```
admin@ubuntu18_04:~$ htpasswd -c /home/admin/www/privado/.htpasswd admin
New password:
Re-type new password:
Adding password for user admin
admin@ubuntu18_04:~$
```

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

Iniciamos como usuario visitante: **su visitante**

```
visitante@ubuntu18_04: /home/admin
```

```
admin@ubuntu18_04:~$ su visitante
Password:
visitante@ubuntu18_04:/home/admin$
```

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

```
visitante@ubuntu18_04: ~
```

```
visitante@ubuntu18_04:/home/admin$ cd
visitante@ubuntu18_04:~$
```

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

```
visitante@ubuntu18_04: ~
```

```
visitante@ubuntu18_04:~$ mkdir www  
visitante@ubuntu18_04:~$
```

Creamos una página web en /www/privado/ :

nano www/indice.html

```
visitante@ubuntu18_04: ~
```

```
visitante@ubuntu18_04:~$ nano www/indice.html
```

Ponemos lo siguiente:

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



```
visitante@ubuntu18_04: ~  
GNU nano 2.9.3 www/indice.html Modified  
<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**

```
root@ubuntu18_04: /home/visitante
```

```
visitante@ubuntu18_04:~$ su
Password:
root@ubuntu18_04:/home/visitante#
```

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

```
root@ubuntu18_04: /home/visitante
```

```
root@ubuntu18_04:/home/visitante# nano /etc/apache2/ports.conf
```

Antes:

```
root@ubuntu18_04: /home/visitante
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 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:

```
root@ubuntu18_04: /home/visitante
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
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 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`

```
root@ubuntu18_04: /etc/apache2/sites-available
```

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

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

`a2ensite my_configuration.conf`

```
root@ubuntu18_04: /etc/apache2/sites-available
```

```
root@ubuntu18_04:/etc/apache2/sites-available# a2ensite my_configuration.conf
Enabling site my_configuration.
To activate the new configuration, you need to run:
  systemctl reload apache2
root@ubuntu18_04:/etc/apache2/sites-available#
```

Por último reiniciamos el servicio con el siguiente comando:

`systemctl reload apache2`

```
root@ubuntu18_04: /etc/apache2/sites-available
```

```
root@ubuntu18_04:/etc/apache2/sites-available# systemctl reload apache2
root@ubuntu18_04:/etc/apache2/sites-available#
```

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**

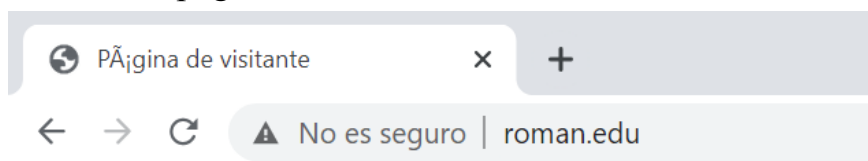
```
root@ubuntu18_04: /etc/apache2/mods-available
root@ubuntu18_04:/etc/apache2/mods-available# a2enmod auth_basic
Considering dependency authn_core for auth_basic:
Module authn_core already enabled
Module auth_basic already enabled
root@ubuntu18_04:/etc/apache2/mods-available#
```

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#
```

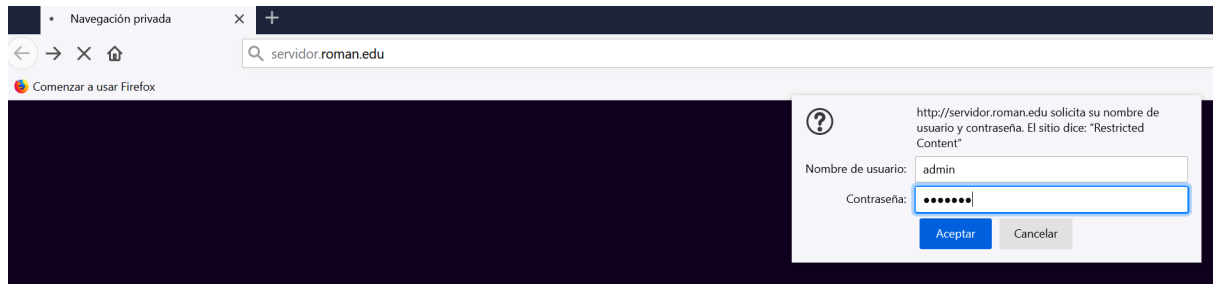
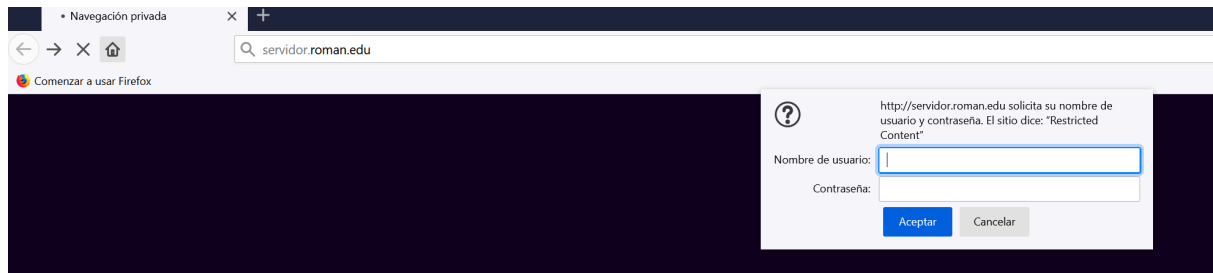
Comprobaciones

Nos sale la página del usuario visitante:

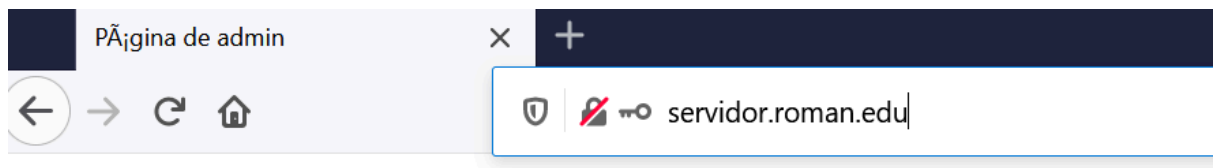


Esto es de visitante

Nos pide loguearnos para ver la página de admin:

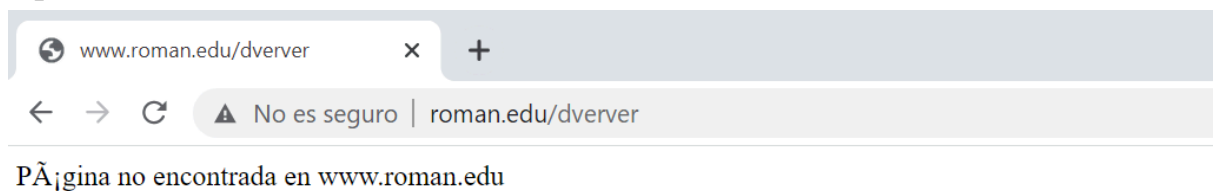


Y luego la pagina que sale le pertenece al usuario admin:

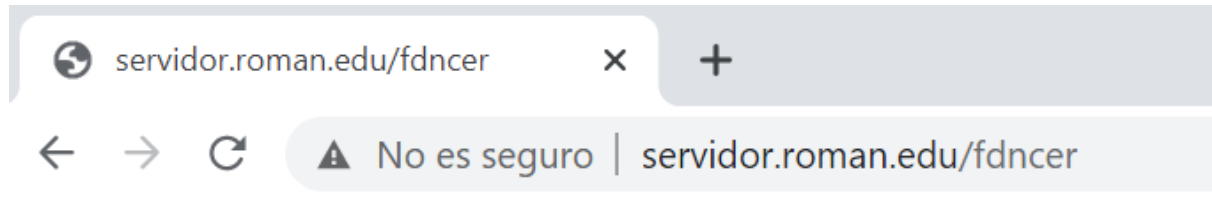


Esto es de admin

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:



Página no encontrada en servidor.roman.edu

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