

## Práctica 9.3. DNS

### Configurar un servidor DNS

#### 1. Objetivos general

- 1.1. Instalar, configurar y administrar un servicio DNS. Para ello necesitaremos conocer los fundamentos de los nombres de dominios y de los componentes que intervienen un sistema DNS.

#### 2. Objetivos específicos

- 2.1. Conocer el servicio DNS
- 2.2. Configurar el servicio DNS en un SO Linux
- 2.3. Crear un escenario virtual con VirtualBox

#### 3. Materiales

- 3.1. Guión de la práctica

#### 4. Recursos

- 4.1. [Curso CEP Lora del Rio. DNS](#)
- 4.2. [Curso Educación a Distancia JA. Ficheros "ServidorDNS.pdf" y "Presentación.pdf"](#)
- 4.3. [Curso CICA. Fichero "Entrega 2".pdf](#)
- 4.4. [Video DNS \(inglés\)](#)
- 4.5. [Red Hat Enterprise Linux 4: Manual de referencia. Capítulo 12. Berkeley Internet Name Domain \(BIND\)](#)
- 4.6. [INTEF. Servidor DNS bind9](#)
- 4.7. [Instalación de BIND versión 9: \(IES Mar de Cádiz\)](#)
- 4.8. [Debian wiki](#)

#### Otros Generales

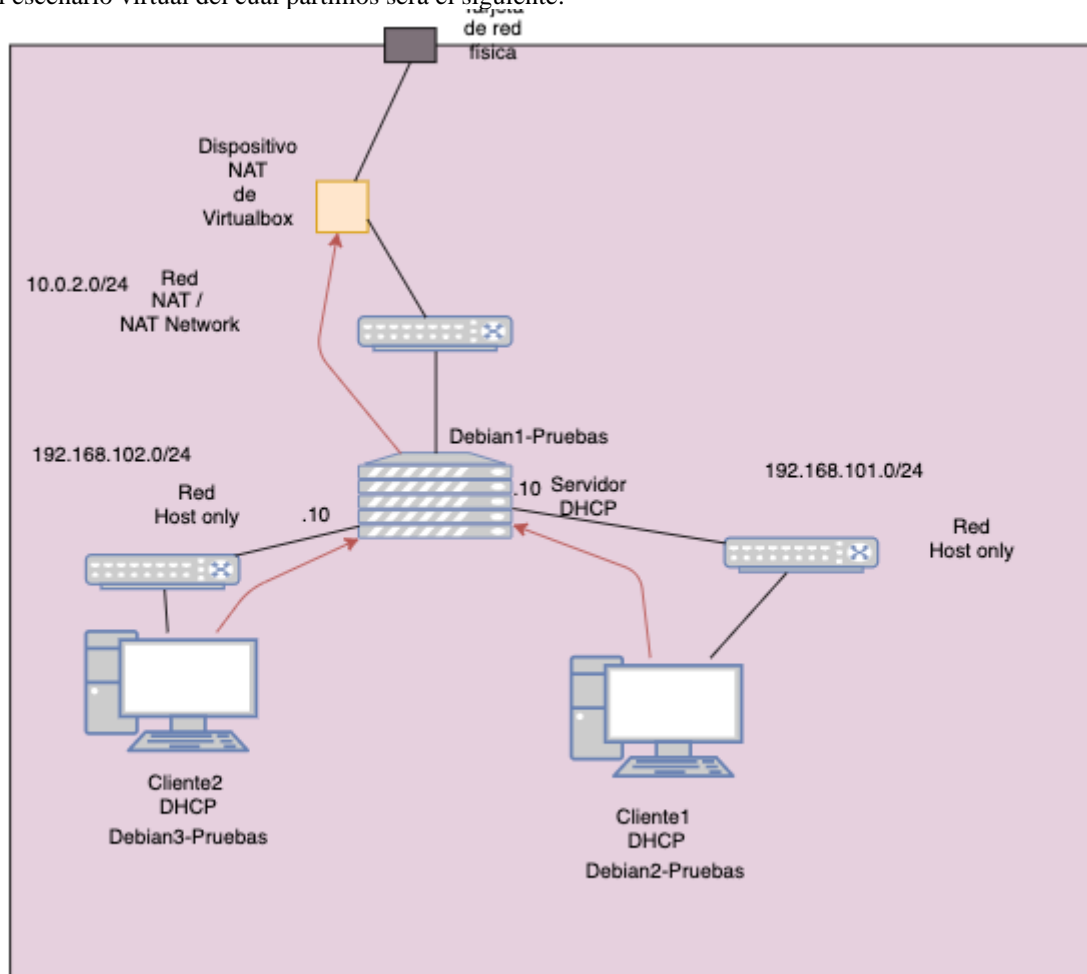
- 4.9. [IP Comando en linux. Artículo de ayuda muy interesante \(inglés\)](#)
- 4.10. [IP comando linux: Chuleta](#) : [IP comando linux chuleta 2](#)
- 4.11. [Linux comandos Referencia](#)
- 4.12.

## Actividad 1:

### Resolución de nombres. Resolución local o resolución centralizada

El acceso a un equipo puede realizarse usando su dirección IP, o a través de un nombre que lo identifica. Así, podríamos si usamos la utilidad ping podríamos lanzar “ping 192.168.1.10” o bien “ping PC10” o “ping PC10.2smr.iesvi”. Si accedemos a un equipo por su nombre, éste deberá ser resuelto a una dirección IP. Lo mismo ocurre con algunas aplicaciones que necesitan obtener el nombre de un equipo dado su dirección IP. En primer caso hablamos de una resolución directa (dado un nombre se obtiene la IP asociada), y de resolución inversa en el segundo caso (se obtiene el nombre dada la dirección IP asociada).

El escenario virtual del cual partimos será el siguiente:



### Esquema general de funcionamiento

El esquema general de funcionamiento es (en distros Linux):

1. Se fija el orden de resolución en un fichero de nombre **/etc/nsswitch.conf** a través de la directiva **host**.
2. El orden seguido para la resolución puede variar. La base de datos con la correspondencia entre nombre y dirección IP puede estar ubicada en:
  - a. un fichero local (**/etc/hosts**)
  - b. o bien a través de un **servidor DNS**

En el primer caso se habla de resolución local, ya que cada equipo encuentra la respuesta en un

fichero ubicada en el propio equipo, y en el segundo caso hablamos de una resolución centralizada de forma que no tenemos que tener repetida la base de datos de nombres y direcciones IP en cada equipo; si usamos la resolución local, cada equipo de la red debe tener configurado este fichero, mientras que en la resolución de servidores DNS el binomio nombre-dirección IP se encuentra centralizado en un servidor. En los comienzos de las redes TCP/IP, había pocos equipos y se usaba sólo la resolución local, pero cuando empezaron a haber más dispositivos fue necesario ubicarlos en una base de datos única para no tener que repetirla en cada equipo, dando lugar a los servidores DNS.

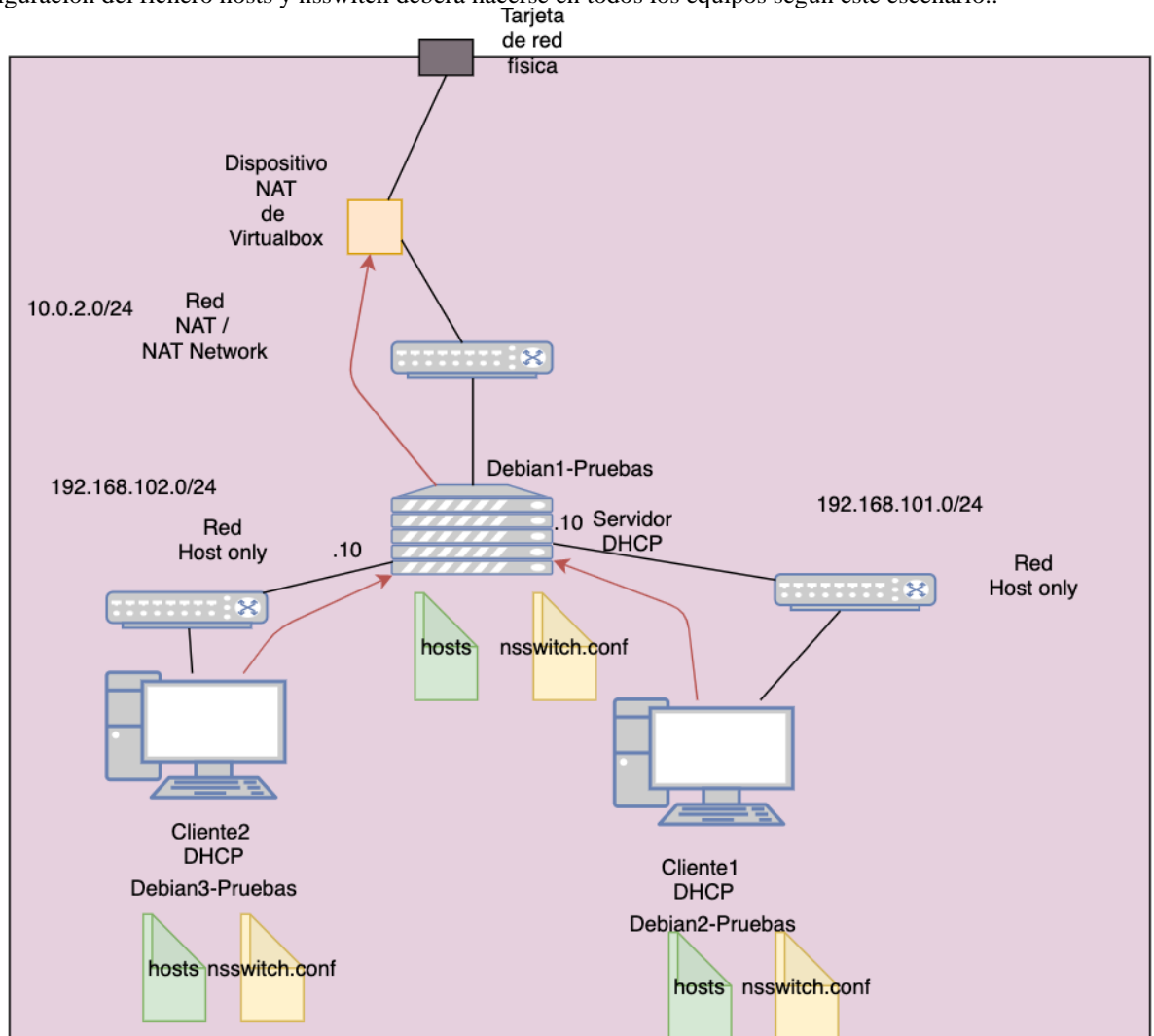
3. Para resolución local debe usarse el valor “**files**”, mientras que indicamos “**dns**” estamos configurando el acceso a un servidor DNS.
4. El orden natural de resolución DNS es:
  - a. Buscar en la caché de nombres DNS resueltos (esta caché está ubicada en la memoria RAM de cada equipo). Si un equipo ha realizado una consulta DNS y ha sido resuelta positivamente, dicha respuesta (nombre+IP) se guarda temporalmente en memoria caché del equipo. De esta forma, si se vuelve a acceder al mismo equipo en ese plazo de tiempo, no será necesario resolver acudiendo al fichero hosts o a un servidor DNS.
  - b. Si el valor a resolver no se encontrase en la caché, se buscaría en el fichero local (/etc/hosts). Este paso supone hacer una consulta en un fichero ubicado en el disco duro del propio equipo. En caso de tener éxito, se guardará en caché para próximas resoluciones.
  - c. Si el nombre buscado no se encuentra en el fichero, se buscaría en el servidor DNS. De igual forma, una consulta con respuesta positiva conlleva a ser guardada en memoria caché.
5. El contenido del fichero /etc/nsswitch.conf para la directiva hosts sería (para seguir el orden natural antes nombrado):

`hosts: files dns`
6. Si nos interesa realizar primero la búsqueda en los servidores DNS y en caso de fallo recurrir al fichero debe configurarse:

`hosts: dns files`
7. Recuerda que si cambiamos estos 2 ficheros, no es necesario reiniciar la red, los cambios son automáticos.

## Resolución local

La configuración del fichero hosts y nsswitch deberá hacerse en todos los equipos según este escenario..



1. Ejemplo: (casa/clase. desde un linux, si tenéis pc físico o con un virtual) añade un nombre DNS público en el fichero /etc/hosts y asígnale una dirección IP de tu red local. Por ejemplo añade al fichero /etc/hosts esta línea:

```
"192.168.0.1  www.google.es"
```

- a. Configura /etc/nsswitch de este modo y comprueba que al hacer "ping www.google.es" responde la IP "192.168.0.1".

```
hosts:          files dns
```

```

GNU nano 3.2 /etc/hosts
127.0.0.1 localhost
127.0.1.1 debian.debian-pruebas
192.168.0.1 www.google.es

# The following lines are desirable for IPv6 capable hosts
::1 localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters

GNU nano 3.2 /etc/nsswitch.conf
#
# Example configuration of GNU Name Service Switch functionality.
# If you have the 'libc-dns-resolver' and 'info' packages installed, try:
# 'info libc "Name Service Switch"' for information about this file.
#
passwd:      files system
group:       files system
shadow:      files
gshadow:     files
hosts:       files dns
networks:    files

protocols:   db files
services:    db files
ethers:      db files
rpc:         db files
  
```

Observa que al hacer “ping [www.google.es](http://www.google.es)” nos aparece entre paréntesis la IP 192.168.0.1; es una señal de que el nombre DNS [www.google.es](http://www.google.es) se ha resuelto en 192.168.0.1, y por tanto, está cogiendo el valor configurado en el fichero /etc/hosts. Acude primero al fichero /etc/hosts que al servidor DNS porque en el fichero /etc/nsswitch se ha configurado la directiva “hosts: files dns”.

- b. Si cambiamos el orden, debes obtener la IP propia de Google.

```

GNU nano 3.2 /etc/hosts
127.0.0.1 localhost
127.0.1.1 debian.debian-pruebas
192.168.0.1 www.google.es

# The following lines are desirable for IPv6 capable hosts
::1 localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters

GNU nano 3.2 /etc/nsswitch.conf
#
# Example configuration of GNU Name Service Switch functionality.
# If you have the 'libc-dns-resolver' and 'info' packages installed, try:
# 'info libc "Name Service Switch"' for information about this file.
#
passwd:      files system
group:       files system
shadow:      files
gshadow:     files
hosts:       dns files
networks:    files

protocols:   db files
services:    db files
ethers:      db files
rpc:         db files
  
```

Observa que ahora la IP que aparece entre paréntesis no es la configurada en /etc/hosts sino la IP propia asociada a este nombre [www.google.es](http://www.google.es), y ello se debe a que la directiva “hosts” de /etc/nsswitch se ha configurado “dns files”, es decir, primero consulta en los servidores DNS y si no hubiera respuesta positiva buscaría en el fichero /etc/hosts.

## 2. Ejercicio:

En este ejercicio vamos a asignar un nombre DNS a cada equipo en el aula.

Para ello, modificar el fichero /etc/hosts (en linux) y C:\Windows\System32\Drivers\etc\hosts (en windows).

Modifica el fichero /etc/hosts agregando la lista de equipos del aula (nombre + IP). Deja el orden de resolución “files dns”. A continuación haz pruebas con “ping nombre\_equipo” para comprobar que se hace la resolución de nombre. Ejemplo: si añades “172.16.5.100 profe”, cuando hagas “ping profe” debes observar que aparece la IP del equipo “profe”. PENDIENTE CAPTURA CUANDO LO

TRABAJEMOS EN EL AULA.

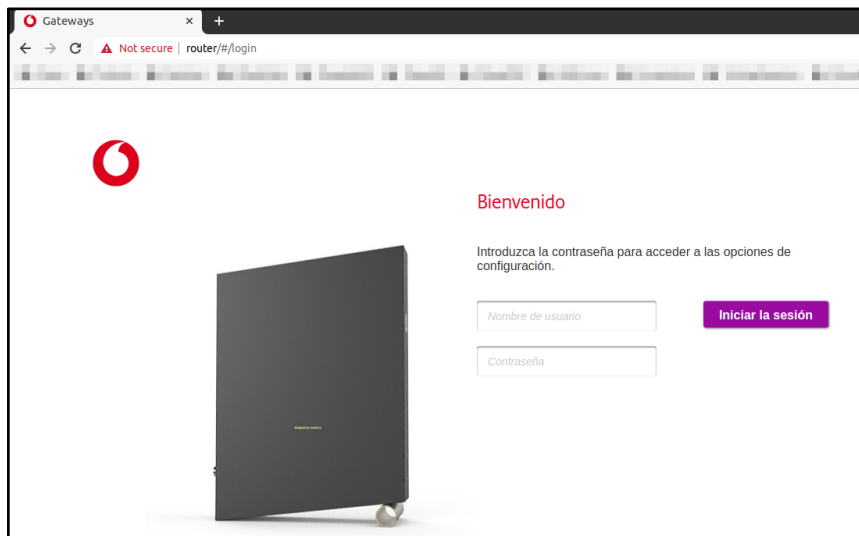
3. Ejercicio: (casa: desde un linux/windows)

haz prueba en casa agregando nombres de equipos singulares de la red de tu casa que tengan configurada una dirección IP como, por ejemplo, ordenadores, smartTV, móviles, impresoras, etc.

```
127.0.0.1    localhost
127.0.1.1    debian Debian1-Pruebas
# 192.168.0.1 www.google.es pepitogrillo.iesvi
192.168.0.10 impresora
192.168.0.11 tv
192.168.0.12 alexa
192.168.0.13 decosatelite
192.168.0.1  router router.casa

# The following lines are desirable for IPv6 capable hosts
::1         localhost ip6-localhost ip6-loopback
ff02::1     ip6-allnodes
ff02::2     ip6-allrouters
```

De esta forma podremos conectar con los dispositivos indicados usando su nombre en vez de la dirección IP. Por ejemplo, si queremos configurar el router a través del servidor web que trae embebido:



4. Ejercicio: seguramente habrás notado que cuando cambiamos el nombre de las máquinas virtuales (/etc/hostname), el sistema se vuelve lento al lanzar los comandos.

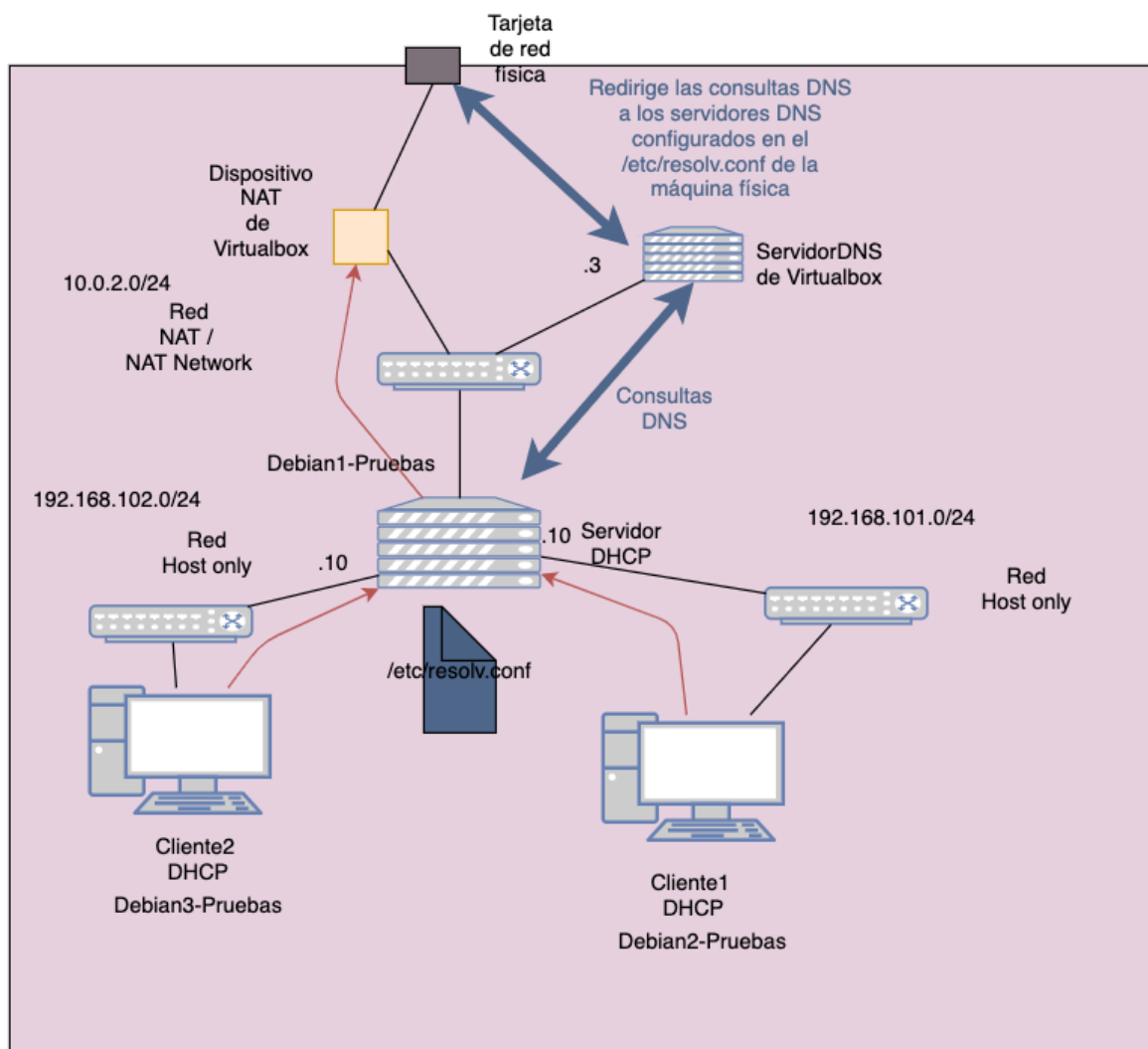
Una causa de esta situación es que el sistema realiza resoluciones de nombres y busca el nombre DNS de la máquina en el fichero /etc/hosts. Si recuerdas cambiamos el nombre de la máquina en el fichero “etc/hostname”, para cambiar el nombre genérico “debian” al nombre real de la máquina “Debian1-Pruebas”. Esto provoca que la resolución de este nombre no funcione bien, por lo que de alguna forma provoca un tiempo de espera. Para evitar esto, la solución es indicar a nivel de resolución local, que el nombre “Debian{X}-pruebas” está asociado a la propia máquina, es decir, a la dirección 127.0.0.1

```
127.0.0.1 localhost
127.0.0.1 debian1-Pruebas.iesvi debian1-Pruebas
```

*Nota: donde pone debian1-Pruebas reemplaza por el nombre correcto del equipo.*

Realiza esta configuración en todas las máquinas virtuales que uses.

## Resolución a través de servidores DNS



Partiendo del escenario anterior, en Debian1-Pruebas podemos comprobar en el fichero `/etc/resolv.conf` como el “nameserver” (servidor dns) es la 10.2.3, la dirección del servidor dns de virtualbox, como puedes observar en la imagen del escenario.

El fichero [/etc/resolv.conf](#) nos va a permitir definir los servidores DNS usados por el equipo en caso de tener que recurrir a ellos.

Veamos la configuración de este fichero en Debian1-Pruebas.

```

usuario@Debian1-Pruebas: ~ 63x37
domain technicolor.net
search technicolor.net
nameserver 10.0.2.3
/etc/resolv.conf (END)

```

Como se puede ver, Virtualbox ha configurado la dirección IP 10.0.2.3 como servidor DNS. Para comprobar que Debian1-Pruebas usa un servidor DNS cuya IP es 10.0.2.3 podemos usar la utilidad `nslookup`. En Debian, puede ser que tengamos que instalar el paquete ‘`dnsutils`’ (`sudo apt install dnsutils`).

```
usuario@Debian1-Pruebas:~$ nslookup www.google.es
Server:      10.0.2.3
Address:     10.0.2.3#53

Non-authoritative answer:
Name:   www.google.es
Address: 216.58.211.35
Name:   www.google.es
Address: 2a00:1450:4003:80b::2003

usuario@Debian1-Pruebas:~$ █
```

Server y Address nos indica el servidor DNS que ha realizado la consulta.

Ejercicio: configura la MV Debian1-Pruebas para usar como servidor DNS el propio servidor DNS embebido en el router de casa/aula. (La imagen es demostrativa).

```
usuario@Debian1-Pruebas:~$ nslookup www.google.es
Server:      192.168.1.1
Address:     192.168.1.1#53

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

usuario@Debian1-Pruebas:~$ █
```

Lee los apartados siguientes de [Recursos-4.2] (ServidorDNS.pdf):

1. Servicios de nombres de dominio
2. Sistema de nombres planos y jerárquicos.
3. El servicio DNS

## ENTREGA:

DOCUMENTACIÓN: Captura de pantallas y explicación para los ejercicios

Instrumento de Evaluación: prueba



Sergio:

```

GNU nano 3.2 nsswitch.conf Modified
# /etc/nsswitch.conf
#
# Example configuration of GNU Name Service Switch functionality.
# If you have the 'glibc-doc-reference' and 'info' packages installed, try:
# info libc "Name Service Switch" for information about this file.
passwd:      files systemd
group:        files systemd
shadow:       files
gshadow:      files
networks:     files
hosts:        files dns
protocols:    db files
services:     db files
ethers:       db files
rpc:          db files
netgroup:     nis
  
```

```

hosts
1 # Copyright (c) 1993-2009 Microsoft Corp.
2
3 # This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
4
5 # This file contains the mappings of IP addresses to host names. Each
6 # entry should be kept on an individual line. The IP address should
7 # be placed in the first column followed by the corresponding host name.
8 # The IP address and the host name should be separated by at least one
9 # space.
10
11 # Additionally, comments (such as these) may be inserted on individual
12 # lines or following the machine name denoted by a '#' symbol.
13
14 # For example:
15
16 # 102.54.94.97 rhino.acme.com # source server
17 # 38.25.63.10 x.acme.com # x client host
18
19 # localhost name resolution is handled within DNS itself.
20 #   127.0.0.1 localhost
21 #   ::1 localhost
22
23 172.16.0.181 profe.com
24 127.0.0.1 www.partitionwizard.com
25 127.0.0.1 www.minitool.com
26 127.0.0.1 www.powerdatarecovery.com
27 127.0.0.1 pas2.partitionwizard.com
28 127.0.0.1 pas2.minitool.com
29 127.0.0.1 pas2.eofsoft.com
30 127.0.0.1 tracking.minitool.com
31
32 127.0.0.1 localhost
33 127.0.0.1 localhost
34 127.0.0.1 localhost::1 localhost
35 ::1 localhost
36
  
```

Windows Explorer view of `BOOTCAMP (C:) > Windows > System32 > drivers > etc`. The file list shows `hosts` (28/04/2021 13:20, Archivo). Below the list, the command prompt shows the output of `ping profe.com`, which is `172.16.0.181`. The output indicates successful ping results.

```

hosts
1 # Copyright (c) 1993-2009 Microsoft Corp.
2
3 # This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
4
5 # This file contains the mappings of IP addresses to host names. Each
6 # entry should be kept on an individual line. The IP address should
7 # be placed in the first column followed by the corresponding host name.
8 # The IP address and the host name should be separated by at least one
9 # space.
10
11 # Additionally, comments (such as these) may be inserted on individual
12 # lines or following the machine name denoted by a '#' symbol.
13
14 # For example:
15
16 # 102.54.94.97 rhino.acme.com # source server
17 # 38.25.63.10 x.acme.com # x client host
18
19 # localhost name resolution is handled within DNS itself.
20 #   127.0.0.1 localhost
21 #   ::1 localhost
22
23 172.16.0.181 profe.com
24 172.16.1.1 router
25 127.0.0.1 www.partitionwizard.com
26 127.0.0.1 www.minitool.com
27 127.0.0.1 www.powerdatarecovery.com
28 127.0.0.1 pas2.partitionwizard.com
29 127.0.0.1 pas2.minitool.com
30 127.0.0.1 pas2.eofsoft.com
31 127.0.0.1 tracking.minitool.com
32
33 127.0.0.1 localhost
34 127.0.0.1 localhost
35 127.0.0.1 localhost::1 localhost
36 ::1 localhost
37
  
```

```

Windows > System32 > drivers > etc
Nombre                               Fecha de modificación             Tipo
hosts                               28/04/2021 13:24                  Archivo
hosts.ics                           15/02/2021 19:23                  Archivo iCalendar
lmhosts.sam                          07/12/2019 10:12                  Archivo SAM
networks                             07/12/2019 10:12                  Archivo
protocol                             07/12/2019 10:12                  Archivo
services                             07/12/2019 10:12                  Archivo

Cmder
λ ipconfig

Configuración IP de Windows

Adaptador de Ethernet Ethernet:

    Sufixo DNS específico para la conexión. . . :
    Vínculo: dirección IPv6 local. . . . . : fe80::bdef:89d6:d9ce:c469%14
    Dirección IPv4. . . . . : 172.16.0.104
    Máscara de subred. . . . . : 255.255.224.0
    Puerta de enlace predeterminada. . . . . : 172.16.1.1
  
```

# SISTEMAS INFORMÁTICOS. 1º DAM. UD9. Práctica 3. DNS

```
Cmder
Sufijo DNS específico para la conexión. . :
Adaptador de LAN inalámbrica Wi-Fi:
Estado de los medios. . . . . : medios desconectados
Sufijo DNS específico para la conexión. . : 41701109.41.andared.ced.junta-andalucia.es
Adaptador de Ethernet Conexión de red Bluetooth:
Estado de los medios. . . . . : medios desconectados
Sufijo DNS específico para la conexión. . :
C:\Users\Sergio DC\Desktop
λ ping router
Haciendo ping a router [172.16.1.1] con 32 bytes de datos:
Respuesta desde 172.16.1.1: bytes=32 tiempo=1ms TTL=64
Tiempo de espera agotado para esta solicitud.
Respuesta desde 172.16.1.1: bytes=32 tiempo=1ms TTL=64
Respuesta desde 172.16.1.1: bytes=32 tiempo<1m TTL=64
Estadísticas de ping para 172.16.1.1:
Paquetes: enviados = 4, recibidos = 3, perdidos = 1
(25% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
Mínimo = 0ms, Máximo = 1ms, Media = 0ms
C:\Users\Sergio DC\Desktop
λ
```

```
SSINF-Debian-Base-1-PRUEBAS [Corriend...
Archivo Máquina Ver Entrada Dispositivos Ayuda
GNU nano 3.2 resolv.conf
nameserver 172.16.1.1
```

```
SSINF-Debian-Base-1-PRUEBAS [Corriend...
Archivo Máquina Ver Entrada Dispositivos Ayuda
root@debian1-Pruebas:/etc# ls
adduser.conf      discover.conf.d   id.so.conf        pam.d             shadow
adjtime           discover-modprobe.conf id.so.conf.d      passud            shadow
alternatives      dpkg             libaudit.conf     passwd            shells
apm               emacs            locale.alias      perl              skel
apparmor          environment      locale.gen        profile           ssh
apparmor.d        fstab            localtime        profile.d         ssl
apt               gai.conf         logcheck          protocols         subgid
bash.bashrc       groff            login.defs        python            subuid
bash_completion   group            logrotate.conf    python2.7         subuid
bash_completion.d group-            logrotate.d        python3            sudoers
bindresvport.blacklist grub.d            machine-id         python3.7          sudoers.d
binfmt.d          gshadow          magic              rc0.d             sysctl.conf
c                 gshadow          magic.mime         rc1.d             sysctl.d
ca-certificates   gss              mailcap            rc2.d             systemd
ca-certificates.conf hdparm.conf      mailcap.order     rc3.d             terminfo
calendar          host.conf        manpath.config    rc4.d             timezone
console-setup     hostname         mime.types        rc5.d             tmpfiles.d
cron.d            hosts            mke2fs.conf       rc6.d             ucf.conf
cron.daily        hosts.allow      modprobe.d         rc6.d             ufw
cron.hourly       hosts.deny       modules-load.d     rcs.d             update-motd.d
cron.monthly      init.d           mtd               rpc               vim
cronontab         inputrc          mtab              rsyslog.conf      X11
cron.weekly       iproute2         network           rsyslog.d         xattr.conf
debconf.conf      issue            networks          security          xdg
debian_version    kernel           nsuitch.conf      selinux           services
default           kernel-lmg.conf os-release         sestatus.conf
deluser.conf      ldapp            pam.conf
dhcp              idmap            id.so.cache
dictionaries-common
root@debian1-Pruebas:/etc#
```

```
SSINF-Debian-Base-1-PRUEBAS [Corriend...
Archivo Máquina Ver Entrada Dispositivos Ayuda
GNU nano 3.2 resolv.conf
nameserver 172.16.1.1
nameserver 8.8.8.8
nameserver 8.8.4.4
```

```
SSINF-Debian-Base-1-PRUEBAS [Corriend...
Archivo Máquina Ver Entrada Dispositivos Ayuda
root@debian1-Pruebas:/etc# sudo apt install dnsutils
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libnss3
Suggested packages:
  rblcheck
The following NEW packages will be installed:
  dnsutils libnss3
0 upgraded, 2 newly installed, 0 to remove and 26 not upgraded.
Need to get 602 kB of archives.
After this operation, 1,027 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://deb.debian.org/debian buster/main amd64 libnss3 amd64 1:9.11.5.P4+dfsg-5.1+deb10u3 [237 kB]
Get:2 http://deb.debian.org/debian buster/main amd64 dnsutils amd64 1:9.11.5.P4+dfsg-5.1+deb10u3 [365 kB]
Fetched 602 kB in 0s (3,667 kB/s)
Selecting previously unselected package libnss3:amd64.
(Reading database ... 28749 files and directories currently installed.)
Preparing to unpack .../libnss3_1:9.11.5.P4+dfsg-5.1+deb10u3_amd64.deb ...
Unpacking libnss3:amd64 (1:9.11.5.P4+dfsg-5.1+deb10u3) ...
Selecting previously unselected package dnsutils.
Preparing to unpack .../dnsutils_1:9.11.5.P4+dfsg-5.1+deb10u3_amd64.deb ...
Unpacking dnsutils (1:9.11.5.P4+dfsg-5.1+deb10u3) ...
Setting up libnss3:amd64 (1:9.11.5.P4+dfsg-5.1+deb10u3) ...
Setting up dnsutils (1:9.11.5.P4+dfsg-5.1+deb10u3) ...
Processing triggers for libc-bin (2.28-10) ...
Processing triggers for man-db (2.8.5-2) ...
root@debian1-Pruebas:/etc#
```

```
SSINF-Debian-Base-1-PRUEBAS [Corriend...
Archivo Máquina Ver Entrada Dispositivos Ayuda
root@debian1-Pruebas:/etc# nslookup www.elpais.es
Server:      172.16.1.1
Address:     172.16.1.1#53

Non-authoritative answer:
www.elpais.es canonical name = lb-redireccionesweb-pro-407952733.eu-west-1.elb.amazonaws.com.
Name:   lb-redireccionesweb-pro-407952733.eu-west-1.elb.amazonaws.com
Address: 34.246.117.165
Name:   lb-redireccionesweb-pro-407952733.eu-west-1.elb.amazonaws.com
Address: 52.209.191.15
root@debian1-Pruebas:/etc#
```

```

SSINF-Debian-Base-1-PRUEBAS [Corriend...
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
GNU nano 3.2                                resolv.conf  Modified
nameserver 172.16.1.1
Get Help  Write Out  Where Is  Cut Text  Justify  Cur Pos  Undo
Exit      Read File  Replace  Uncut Text  To Spell  Go To Line  Redo
CTRL DERECHA
  
```

```

SSINF-Debian-Base-1-PRUEBAS [Corriend...
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
root@debian1-Pruebas:/etc# nslookup www.elpais.es
Server:
172.16.1.1
Address:
172.16.1.1#53

Non-authoritative answer:
www.elpais.es canonical name = lb-redireccionesweb-pro-407952733.eu-west-1.elb.amazonaws.com.
Name:   lb-redireccionesweb-pro-407952733.eu-west-1.elb.amazonaws.com
Address: 34.246.117.165
Name:   lb-redireccionesweb-pro-407952733.eu-west-1.elb.amazonaws.com
Address: 52.209.191.15

root@debian1-Pruebas:/etc# _
  
```

```

SSINF-Debian-Base-1-PRUEBAS [Corriend...
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
##### Actividad 1 - Ejercicio 3 #####
192.168.1.43  xiaomiSergio

# The following lines are desirable for IPv6 capable hosts
::1 localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters

root@debian1-Pruebas:/etc# ping xiaomiSergio
PING xiaomiSergio (192.168.1.43) 56(84) bytes of data:
64 bytes from xiaomiSergio (192.168.1.43): icmp_seq=13 ttl=63 time=48.4 ms
64 bytes from xiaomiSergio (192.168.1.43): icmp_seq=14 ttl=63 time=12.7 ms
64 bytes from xiaomiSergio (192.168.1.43): icmp_seq=15 ttl=63 time=10.9 ms
  
```

```

root@debian1-Pruebas:/etc# ping xiaomiSergio
PING xiaomiSergio (192.168.1.43) 56(84) bytes of data:
64 bytes from xiaomiSergio (192.168.1.43): icmp_seq=1 ttl=63 time=243 ms
64 bytes from xiaomiSergio (192.168.1.43): icmp_seq=2 ttl=63 time=265 ms
64 bytes from xiaomiSergio (192.168.1.43): icmp_seq=3 ttl=63 time=357 ms
64 bytes from xiaomiSergio (192.168.1.43): icmp_seq=4 ttl=63 time=106 ms
64 bytes from xiaomiSergio (192.168.1.43): icmp_seq=5 ttl=63 time=129 ms
64 bytes from xiaomiSergio (192.168.1.43): icmp_seq=6 ttl=63 time=151 ms
64 bytes from xiaomiSergio (192.168.1.43): icmp_seq=7 ttl=63 time=167 ms
64 bytes from xiaomiSergio (192.168.1.43): icmp_seq=8 ttl=63 time=195 ms
64 bytes from xiaomiSergio (192.168.1.43): icmp_seq=9 ttl=63 time=495 ms
64 bytes from xiaomiSergio (192.168.1.43): icmp_seq=10 ttl=63 time=271 ms
64 bytes from xiaomiSergio (192.168.1.43): icmp_seq=11 ttl=63 time=268 ms
64 bytes from xiaomiSergio (192.168.1.43): icmp_seq=12 ttl=63 time=286 ms
64 bytes from xiaomiSergio (192.168.1.43): icmp_seq=13 ttl=63 time=578 ms
64 bytes from xiaomiSergio (192.168.1.43): icmp_seq=14 ttl=63 time=1130 ms
64 bytes from xiaomiSergio (192.168.1.43): icmp_seq=15 ttl=63 time=114 ms
64 bytes from xiaomiSergio (192.168.1.43): icmp_seq=16 ttl=63 time=549 ms
64 bytes from xiaomiSergio (192.168.1.43): icmp_seq=17 ttl=63 time=1010 ms
64 bytes from xiaomiSergio (192.168.1.43): icmp_seq=18 ttl=63 time=199 ms
64 bytes from xiaomiSergio (192.168.1.43): icmp_seq=19 ttl=63 time=244 ms
64 bytes from xiaomiSergio (192.168.1.43): icmp_seq=20 ttl=63 time=245 ms
64 bytes from xiaomiSergio (192.168.1.43): icmp_seq=21 ttl=63 time=274 ms
64 bytes from xiaomiSergio (192.168.1.43): icmp_seq=22 ttl=63 time=289 ms
64 bytes from xiaomiSergio (192.168.1.43): icmp_seq=23 ttl=63 time=101 ms
64 bytes from xiaomiSergio (192.168.1.43): icmp_seq=24 ttl=63 time=128 ms
64 bytes from xiaomiSergio (192.168.1.43): icmp_seq=25 ttl=63 time=151 ms
^C
--- xiaomiSergio ping statistics ---
25 packets transmitted, 25 received, 0% packet loss, time 77ms
rtt min/avg/max/mdev = 101.117/317.847/1129.633/255.519 ms, pipe 2
root@debian1-Pruebas:/etc#
  
```

Actividad 2 – Ejercicio 3

Actividad 2 – Ejercicio 3

```

##### Actividad 1 - Ejercicio 3 #####
192.168.1.43 xiaomiSergio
192.168.1.34 samsungS6papa

# The following lines are desirable for IPv6 capable hosts
::1 localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters

root@debian1-Pruebas:/etc# ping samsungS6papa
PING samsungS6papa (192.168.1.34) 56(84) bytes of data.
From 192.168.1.40 (192.168.1.40) icmp_seq=3 Destination Host Unreachable
From 192.168.1.40 (192.168.1.40) icmp_seq=6 Destination Host Unreachable
From 192.168.1.40 (192.168.1.40) icmp_seq=9 Destination Host Unreachable

```

Actividad 2 – Ejercicio 3... error UNREACHABLE

```

root@debian1-Pruebas:/etc# ping samsungS6papa
PING samsungS6papa (192.168.1.34) 56(84) bytes of data.
From 192.168.1.40 (192.168.1.40) icmp_seq=3 Destination Host Unreachable
From 192.168.1.40 (192.168.1.40) icmp_seq=6 Destination Host Unreachable
From 192.168.1.40 (192.168.1.40) icmp_seq=9 Destination Host Unreachable
^C
--- samsungS6papa ping statistics ---
38 packets transmitted, 0 received, +3 errors, 100% packet loss, time 812ms
root@debian1-Pruebas:/etc#

```

Actividad 2 – Ejercicio 3... error UNREACHABLE

```

##### Actividad 1 - Ejercicio 3 #####
192.168.1.43 xiaomiSergio
192.168.1.34 samsungS6papa
192.168.1.40 wifiCasa

# The following lines are desirable for IPv6 capable hosts
::1 localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters

root@debian1-Pruebas:/etc# ping wifiCasa
PING wifiCasa (192.168.1.40) 56(84) bytes of data.
64 bytes from wifiCasa (192.168.1.40): icmp_seq=1 ttl=127 time=0.851 ms
64 bytes from wifiCasa (192.168.1.40): icmp_seq=2 ttl=127 time=1.04 ms
64 bytes from wifiCasa (192.168.1.40): icmp_seq=3 ttl=127 time=1.56 ms

```

Actividad 2 – Ejercicio 3

```

root@debian1-Pruebas:/etc# ping wifiCasa
PING wifiCasa (192.168.1.40) 56(84) bytes of data.
64 bytes from wifiCasa (192.168.1.40): icmp_seq=1 ttl=127 time=0.851 ms
64 bytes from wifiCasa (192.168.1.40): icmp_seq=2 ttl=127 time=1.04 ms
64 bytes from wifiCasa (192.168.1.40): icmp_seq=3 ttl=127 time=1.56 ms
64 bytes from wifiCasa (192.168.1.40): icmp_seq=4 ttl=127 time=1.78 ms
64 bytes from wifiCasa (192.168.1.40): icmp_seq=5 ttl=127 time=2.02 ms
64 bytes from wifiCasa (192.168.1.40): icmp_seq=6 ttl=127 time=1.75 ms
64 bytes from wifiCasa (192.168.1.40): icmp_seq=7 ttl=127 time=0.796 ms
64 bytes from wifiCasa (192.168.1.40): icmp_seq=8 ttl=127 time=1.16 ms
64 bytes from wifiCasa (192.168.1.40): icmp_seq=9 ttl=127 time=0.743 ms
64 bytes from wifiCasa (192.168.1.40): icmp_seq=10 ttl=127 time=1.35 ms
64 bytes from wifiCasa (192.168.1.40): icmp_seq=11 ttl=127 time=2.15 ms
64 bytes from wifiCasa (192.168.1.40): icmp_seq=12 ttl=127 time=3.32 ms
64 bytes from wifiCasa (192.168.1.40): icmp_seq=13 ttl=127 time=1.31 ms
64 bytes from wifiCasa (192.168.1.40): icmp_seq=14 ttl=127 time=1.06 ms
64 bytes from wifiCasa (192.168.1.40): icmp_seq=15 ttl=127 time=1.32 ms
64 bytes from wifiCasa (192.168.1.40): icmp_seq=16 ttl=127 time=1.61 ms
64 bytes from wifiCasa (192.168.1.40): icmp_seq=17 ttl=127 time=1.19 ms
64 bytes from wifiCasa (192.168.1.40): icmp_seq=18 ttl=127 time=2.85 ms
^C
--- wifiCasa ping statistics ---
18 packets transmitted, 18 received, 0% packet loss, time 48ms
rtt min/avg/max/mdev = 0.743/1.546/3.318/0.674 ms
root@debian1-Pruebas:/etc#

```

Actividad 2 – Ejercicio 3

## Actividad 2. Fundamentos del servicio DNS

En esta actividad vamos a aprender el funcionamiento del servicio DNS con objeto de montar nuestro propio servidor. En la actividad 1, hemos usado el servicio DNS desde el lado del cliente a través de resoluciones locales o centralizadas. Ha llegado el momento de desplegar nuestro propio servicio.

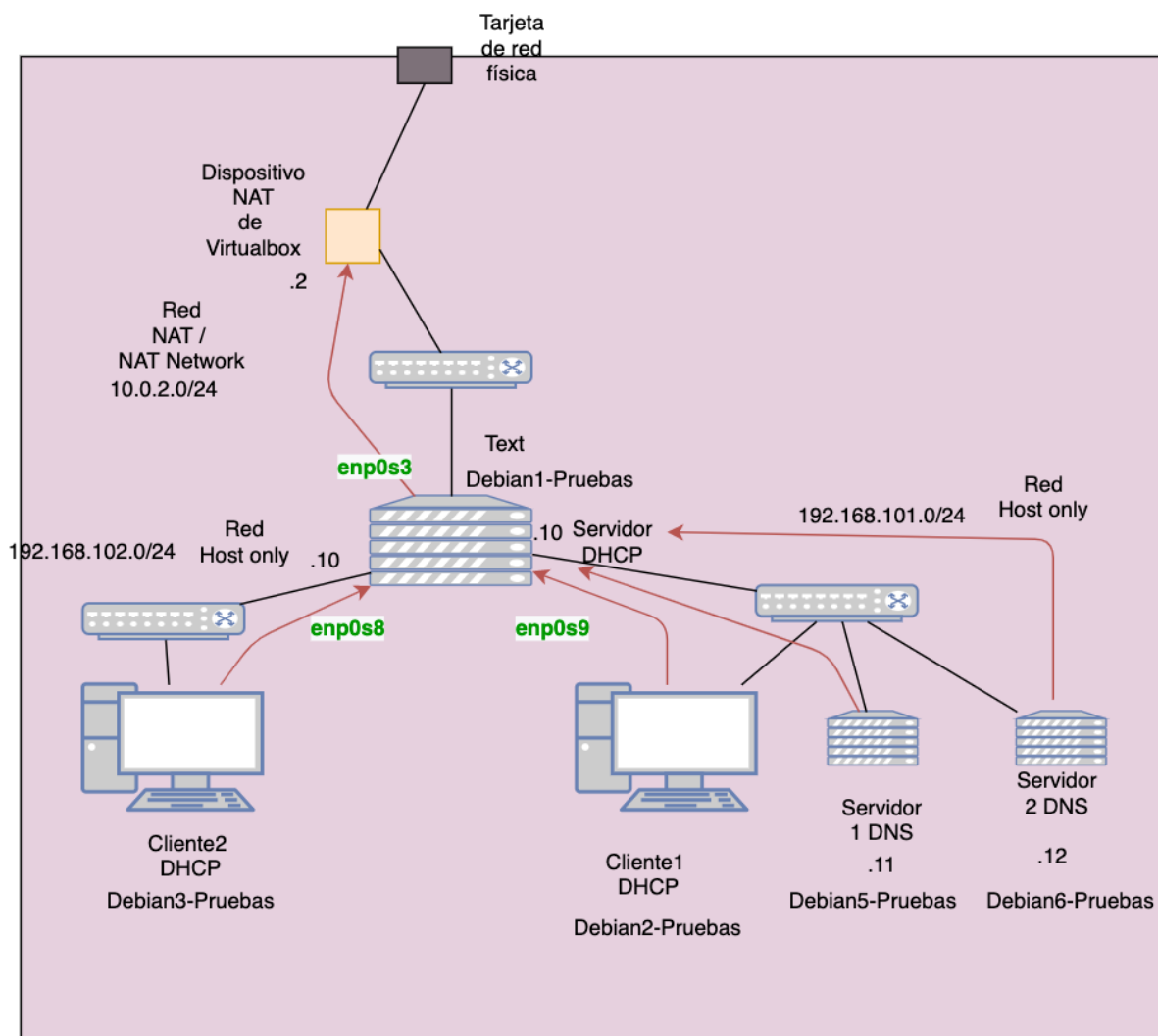
Para ello, vamos a realizar los siguientes ejercicios:

1. Ver un [vídeo](#) sobre DNS [Recursos-4.4]
2. Leer varios apartados de [Recursos 4.2] (Usa los ficheros ServidorDNS.pdf como PresentacionDNS.pdf):
  - a. Funcionamiento del servicio DNS
  - b. El espacio de nombres de dominio
  - c. Tipos de dominio
  - d. Dominios de primer nivel
  - e. Servidores raíz y de dominios de primer nivel y sucesivos.
  - f. Resolutores de nombres
  - g. Proceso de resolución de un nombre de dominio
  - h. Resoluciones directas y resoluciones inversas
  - i. Zonas. Primarias y secundarias
  - j. Tipos de servidores de DNS: primarios, secundarios, caché, reenviador
3. El siguiente videotutorial [TCP Y UDP: los protocolos de transporte de las aplicaciones. Demo en Packet Tracer](#) reproduce una comunicación TCP/IP mediante el simulador de red Packet Tracer. Se usa el laboratorio “[ServidoresDNS+WEB+DHCP.pkt](#)” (anexo a la tarea) que contiene 3 servidores para ofrecer los servicios DNS, Web y DHCP. A este escenario acompaña un equipo cliente que configura sus parámetros IP de forma dinámica y accede al servidor web previa resolución del nombre DNS del servidor web; el acceso a este servidor web por su nombre será posible por resolución DNS mediante el servidor DNS.

## ENTREGA

- [Documentación a entregar:](#) nada
- [Instrumento de evaluación:](#) prueba

### Actividad 3. Instalación y configuración del servicio. Zonas, y registros. Clientes DNS: pruebas.



Este es el escenario a utilizar para esta actividad.

Antes simplemente asignábamos las IPs en la configuración del DHCP, pero esos servidores aún no existían.

Ahora vamos a crearlos realmente, y serán los servidores que nos permitan realizar las resoluciones DNS.

Como se vé en la imagen, serán los servidores (Debian5-Pruebas y Debian6-Pruebas) y se usarán para crear nuestro propio espacio de nombres DNS, por un lado, y centralizar la resolución de nombres DNS externos a nuestra organización.

El primer paso que debes realizar es la configuración IP y el nombre de ambos servidores DNS:

- Direccionamiento IP estático (*IP+máscara+gateway* según aparece en el escenario). Así para Debian5-Pruebas:

```
auto enp0s3
iface enp0s3 inet static
    address 192.168.101.11
    netmask 255.255.255.0
    gateway 192.168.101.10
```

- Nombres según hemos indicado anteriormente, es decir, *Debian5-Pruebas* y *Debian6-Pruebas* (deben aparecer en el prompt de cada equipo). Modifica tanto el fichero */etc/hostname* como */etc/hosts* agregando el nuevo nombre.

```

usuario@Debian5-Pruebas:~$ ip r
default via 192.168.101.10 dev enp0s3 onlink
169.254.0.0/16 dev enp0s3 scope link metric 1000
192.168.101.0/24 dev enp0s3 proto kernel scope link src 192.168.101.11
usuario@Debian5-Pruebas:~$

```

La imagen la tabla de rutas con la configuración IP indicada. Observar el prompt que aparece el nombre Debian5-Pruebas.

No te olvides de:

1. Vamos a realizar **control de versiones** de la configuración. Realiza un commit de los directorios de configuración de DNS (**git add \*** + **git commit -m "servidor DHCP instalado"**).
2. Antes de realizar ninguna instalación, haz un **snapshot** de cada máquina virtual que hayas usado y vayas a configurar DNS (realiza todos los que necesitarás durante la tarea; se recomienda realizarlo cada vez que se tenga una configuración operativa)
  - Debian1-Pruebas: para tener una foto de esta máquina antes de realizar cambios de esta tarea. Para ello usa la utilidad que ofrece VirtualBox (barra de menús → Snapshot). Escribe un nombre y una descripción.
  - Debian5-Pruebas y Debian6-Pruebas → snapshot del instante inicial antes de realizar ninguna configuración.

### Ejercicio 0. Habilitando el acceso a Internet a las máquinas virtuales de las 2 redes host-only.

Si intentamos instalar el paquete bind9 (paquete para disponer de un servidor DNS), observarás que falla. La razón se encuentra en que Debian5-Pruebas no tiene conectividad IP a Internet. ¿Por qué? ¿Cómo podemos solucionarlo? Analicemos la situación actual cuando Debian5-Pruebas intenta acceder a Internet sigue los siguientes pasos:

1. La IP de destino no está en su tabla de rutas ni pertenece a una red que esté en la tabla de rutas. Por tanto envía el paquete a su puerta de enlace, es decir, Debian1-Pruebas.
2. Debian1-Pruebas recibe el paquete y por la misma razón de antes lo envía a su puerta de enlace, en este caso el dispositivo NAT (10.0.2.15). Analicemos el paquete:
  - a. IP-origen = Debian5-Pruebas = 192.168.101.11
  - b. IP-destino = IP de algún equipo de Internet
3. El dispositivo NAT cambia (translate = traduce) la IP origen por la IP de la tarjeta de la máquina física. Si la red NAT estuviera conectada a la tarjeta eth0 (o cualquier otro nombre que tuviera como enp0s3 o similar) de la máquina física saldría de este modo:
 

IP-origen = IP-MF = 172.16.4.25 (supongamos que es ésta la IP de la tarjeta física de la máquina física).

IP-destino = IP de algún equipo de Internet
4. El dispositivo NAT no sólo cambia la IP origen del paquete sino que deja registrada dicha traducción para cuando venga la respuesta del equipo de destino.
5. Si nuestra máquina física tiene bien configurada la red tendrá Internet por lo que podrá obtener respuesta del paquete anterior. En este caso el paquete de vuelta que recibirá la máquina física es:
 

IP-origen = IP de algún equipo de Internet

IP-destino = 172.16.4.25
6. Cómo el dispositivo NAT registró una traducción para el paquete de envío, vuelve a traducir a sus valores iniciales quedando:
 

IP-origen = IP de algún equipo de Internet

IP-destino = IP-Debian5-Pruebas = 192.168.101.11
7. **Aquí es donde tenemos el problema, el dispositivo NAT NO SABE LLEGAR A Debian5-Pruebas;** en tu tabla de rutas NO HAY RUTAS para llegar a Debian5-pruebas. Podría comunicar con cualquier equipo de su red NAT (Debian1-Pruebas, por ejemplo -gracias a ello Debian1-Pruebas SÍ tiene conectividad a Internet).



8. **Solución:** cuando un paquete llegue a Debian1-Pruebas con destino una red que no sea las que están conectadas a sus interfaces (las 2 redes host-only y la red NAT) le vamos a cambiar la IP de origen por la suya propia. Es decir, antes de pasar por el dispositivo NAT vamos a hacerle otro NAT pero en Debian1-Pruebas; de este forma los paquetes de Debian5-Pruebas “llevarán” la “identidad” de Debian1-Pruebas, es decir, tendrán como IP-origen la IP de Debian1-Pruebas, máquina que SÍ tiene acceso a Internet. Para ello vamos a usar la utilidad “**iptables**” la cual es un potente cortafuegos de los sistemas Linux y también ofrece funcionalidades como la que vamos a explotar. Le vamos a aplicar las reglas siguientes a las 203.10 tarjetas de Debian1-Pruebas que conectan con las 2 redes host-only.

```
up iptables -t nat -A POSTROUTING -o enp0s3 -s 192.168.101.0/24 -j MASQUERADE
down iptables -t nat -D POSTROUTING -o enp0s3 -s 192.168.101.0/24 -j MASQUERADE
```

9. Así la configuración IP de Debian1-Pruebas queda (/etc/network/interfaces):

```
auto enp0s3
iface enp0s3 inet dhcp

auto enp0s8
iface enp0s8 inet static
    address 192.168.101.10
    netmask 255.255.255.0
    up iptables -t nat -A POSTROUTING -o enp0s3 -s 192.168.101.0/24 -j MASQUERADE
    down iptables -t nat -D POSTROUTING -o enp0s3 -s 192.168.101.0/24 -j MASQUERADE

auto enp0s9
iface enp0s9 inet static
    address 192.168.102.10
    netmask 255.255.255.0
    up iptables -t nat -A POSTROUTING -o enp0s3 -s 192.168.102.0/24 -j MASQUERADE
    down iptables -t nat -D POSTROUTING -o enp0s3 -s 192.168.102.0/24 -j MASQUERADE
```

No te olvides de reiniciar el servicio de red (sudo service networking restart). Nota: si saliera algún error, acuérdate de monitorizar con:

- sudo service networking status
- o ver el fichero /var/log/syslog
- prueba también a reiniciar si ves que has depurado todos los errores y sigue dando problemas.

10. ¿Qué significan las 2 reglas anteriores?

- Cuando se levante la tarjeta enp0s8 (“up”), se lanza el comando “*iptables -t nat -A POSTROUTING -o enp0s3 -s 192.168.101.0/24 -j MASQUERADE*”. ¿Qué hace este comando? Añade una regla de postrouting de forma que los paquetes que vengan de la red 192.168.101.0/24 (-s 192.168.101.0/24) y que salgan por la interfaz enp0s3 (-o enp0s3) se les hará un cambio de IP, en concreto a la IP que tuviera la tarjeta enp3s0 (MASQUERADE). Perfecto, ya hemos conseguido que Debian1-Pruebas haga NAT. La regla “down ...” lo que hace es eliminar dicha regla cuando la tarjeta se echa abajo.
- Para la otra la explicación es exactamente igual salvo que deberemos cambiar la dirección IP de la red de 192.168.101.0 a 192.168.102.0.

11. Si has realizado correctamente la configuración, al lanzar el comando “sudo iptables -L -t nat” te saldrá:

```
usuario@Debian1-Pruebas:~$ sudo iptables -L -t nat
Chain PREROUTING (policy ACCEPT)
target     prot opt source                destination

Chain INPUT (policy ACCEPT)
target     prot opt source                destination

Chain POSTROUTING (policy ACCEPT)
target     prot opt source                destination
MASQUERADE all  -- 192.168.101.0/24        anywhere
MASQUERADE all  -- 192.168.102.0/24        anywhere

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination
usuario@Debian1-Pruebas:~$
```



12. Importante es no olvidarte de ENRUTAR en Debian1-Pruebas; los paquetes que provienen de las 2 redes host-only deben ser reenviados por Debian1-Pruebas. ¿Te acuerdas cómo activar el enrutamiento en equipos Linux?

```
GNU nano 3.2 /etc/sysctl.conf
#net.ipv4.tcp_syncookies=1

# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1
```

No te olvides de reiniciar el equipo (sudo init 6).

13. Prueba ahora a conectar a Internet desde Debian5-Pruebas:

```
usuario@Debian5-Pruebas:~$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=61 time=15.5 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=61 time=16.5 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=61 time=17.3 ms
^C
--- 8.8.8.8 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 4ms
rtt min/avg/max/mdev = 15.490/16.414/17.255/0.730 ms
usuario@Debian5-Pruebas:~$
```

Tenemos conectividad IP!!

```
usuario@Debian5-Pruebas:~$ host www.google.es
www.google.es has address 216.58.201.131
www.google.es has IPv6 address 2a00:1450:4003:80b::2003
usuario@Debian5-Pruebas:~$
```

Tenemos resolución DNS!!

14. El proceso completo queda:
- Debian5-Pruebas envía un paquete IP:
    - IP-origen = IP-Debian5-Pruebas = 192.168.101.11
    - IP-destino = IP de algún equipo de Internet.
  - El paquete de Debian5-Pruebas llega a Debian1-Pruebas y Debian1-Pruebas lo reenvía y realiza una conversión de dirección IP origen (NAT) como:
    - IP-origen = IP-Debian1-Pruebas = 10.0.2.15 (esta IP puede variar ya que la obtiene de forma dinámica en su red NA)..
    - IP-destino = IP de algún equipo de Internet.
  - Cuando el paquete llega al dispositivo NAT, éste lo reenvía como:
    - IP-origen = IP-MF = 172.16.4.25 (la que tuviera la tarjeta física de la máquina física).
    - IP-destino = IP de algún equipo de Internet
  - Cuando el paquete vuelve a la máquina física se deshace la traducción:
    - IP-origen = IP de algún equipo de Internet
    - IP-destino = IP-Debian1-Pruebas = 10.0.2.15
  - De esta forma llega a Debian1-Pruebas, quien a su vez deshace su propia traducción NAT:
    - IP-origen = IP de algún equipo de Internet
    - IP-destino = IP-Debian5-Pruebas = 192.168.101.11

### Ejercicio 1. Instalación y ficheros de configuración

Vamos a instalar el servicio **bind9**.

- `sudo apt update`
- `sudo apt install bind9`

Comprobamos que el servicio está operativo:

```

usuario@Debian5-Pruebas:~$ sudo service bind9 status
● bind9.service - BIND Domain Name Server
   Loaded: loaded (/lib/systemd/system/bind9.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2020-12-19 13:52:13 GMT; 34s ago
     Docs: man:named(8)
  Main PID: 1013 (named)
    Tasks: 4 (limit: 1150)
   Memory: 12.8M
    CGroup: /system.slice/bind9.service
            └─1013 /usr/sbin/named -u bind

Dec 19 13:52:13 Debian5-Pruebas named[1013]: FORMERR resolving './NS/IN': 192.112.36.4#53
Dec 19 13:52:13 Debian5-Pruebas named[1013]: DNS format error from 192.203.230.10#53 resolving ./NS: non-improving referral
Dec 19 13:52:13 Debian5-Pruebas named[1013]: FORMERR resolving './NS/IN': 192.203.230.10#53
Dec 19 13:52:13 Debian5-Pruebas named[1013]: DNS format error from 192.36.148.17#53 resolving ./NS: non-improving referral
Dec 19 13:52:13 Debian5-Pruebas named[1013]: FORMERR resolving './NS/IN': 192.36.148.17#53
Dec 19 13:52:13 Debian5-Pruebas named[1013]: DNS format error from 199.7.83.42#53 resolving ./NS: non-improving referral
Dec 19 13:52:13 Debian5-Pruebas named[1013]: FORMERR resolving './NS/IN': 199.7.83.42#53
Dec 19 13:52:13 Debian5-Pruebas named[1013]: DNS format error from 198.41.0.4#53 resolving ./NS: non-improving referral
Dec 19 13:52:13 Debian5-Pruebas named[1013]: FORMERR resolving './NS/IN': 198.41.0.4#53
Dec 19 13:52:13 Debian5-Pruebas named[1013]: resolver printing query complete
usuario@Debian5-Pruebas:~$

```

Otra utilidad es **rndc**:

```

usuario@Debian5-Pruebas:~$ sudo rndc status
[sudo] password for usuario:
version: BIND 9.11.5-P4-5.1+deb10u2-Debian (Extended Support Version) <id:998753c>
running on Debian5-Pruebas: Linux x86_64 4.19.0-13-amd64 #1 SMP Debian 4.19.160-2 (2020-11-28)
boot time: Sun, 03 Jan 2021 08:41:33 GMT
last configured: Sun, 03 Jan 2021 08:41:34 GMT
configuration file: /etc/bind/named.conf
CPUs found: 1
worker threads: 1
UDP listeners per interface: 1
number of zones: 105 (97 automatic)
debug level: 0
xfers running: 0
xfers deferred: 0
soa queries in progress: 0
query logging is OFF
recursive clients: 0/900/1000
tcp clients: 3/150
server is up and running
usuario@Debian5-Pruebas:~$

```

Antes de iniciar (o reiniciar) el servicio, podemos comprobar la sintaxis de la configuración con:

```

usuario@Debian5-Pruebas:~$ sudo named-checkconf
usuario@Debian5-Pruebas:~$

```

Si no hay errores, NO SE MOSTRARÁ nada, como se ve en la imagen anterior.

Lee la siguiente documentación para reforzar sobre otras alternativas a BIND y opciones de administración del servicio:

- [2] Fichero ServidorDNS.pdf - Apartado “Instalación, configuración y administración de servidores DNS” (página 26 y 27). Aprovecha la lectura para conocer otras soluciones alternativas a bind9. Revisa cómo administrar el servicio (detenerlo, reiniciarlo, comprobar su estado, iniciarlo)
- En las otras referencias bibliográficas puedes contrastar la información anterior.

Si analizamos la instalación del servidor bind, podemos observar los siguientes ficheros:

```

usuario@Debian5-Pruebas:/etc/bind$ ls -l
total 48
-rw-r--r-- 1 root root 2761 Aug 25 09:10 bind.keys
-rw-r--r-- 1 root root 237 Aug 25 09:10 db.0
-rw-r--r-- 1 root root 271 Aug 25 09:10 db.127
-rw-r--r-- 1 root root 237 Aug 25 09:10 db.255
-rw-r--r-- 1 root root 353 Aug 25 09:10 db.empty
-rw-r--r-- 1 root root 270 Aug 25 09:10 db.local
-rw-r--r-- 1 root bind 463 Aug 25 09:10 named.conf
-rw-r--r-- 1 root bind 498 Aug 25 09:10 named.conf.default-zones
-rw-r--r-- 1 root bind 165 Aug 25 09:10 named.conf.local
-rw-r--r-- 1 root bind 846 Aug 25 09:10 named.conf.options
-rw-r----- 1 bind bind 77 Dec 19 13:52 rndc.key
-rw-r--r-- 1 root root 1317 Aug 25 09:10 zones.rfc1918
usuario@Debian5-Pruebas:/etc/bind$

```

- **named.conf**: archivo principal de configuración del servicio. En la versión actual no tiene ninguna directiva propia de configuración sino que remite a otros ficheros mediante “include”.

```
include "/etc/bind/named.conf.options";
include "/etc/bind/named.conf.local";
include "/etc/bind/named.conf.default-zones";
```

- **named.conf.options:** contiene opciones de configuración del servicio, que iremos analizando posteriormente en diferentes ejercicios (forwarders, forward-only, listen-on, allow-query, allow-recursion ...).
- **named.conf.default-zones:** Contiene información sobre las zonas que se crean por defecto al instalar bind y en las que, por tanto, tiene autoridad el servidor. De cada zona se tiene su nombre, su tipo y el nombre y ubicación del archivo de zona. De todas ellas debe destacarse la zona con los servidores de nombre raíz:

```
// prime the server with knowledge of the root servers
zone "." {
    type hint;
    file "/usr/share/dns/root.hints";
};
```

Si exploramos el contenido del fichero **root.hints**, podemos ver el listado de los 13 servidores de nombres raíz que actualmente tenemos mundialmente. De esta forma, nuestro servidor DNS conoce sus direcciones IP y por tanto puede acudir a ellos para el caso que se recurra a una consulta iterativa (recuerda el vídeo sobre DNS).

```
.                3600000      NS      A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET. 3600000      A       198.41.0.4
A.ROOT-SERVERS.NET. 3600000      AAAA    2001:503:ba3e::2:30
;
; FORMERLY NS1.ISI.EDU
;
.                3600000      NS      B.ROOT-SERVERS.NET.
B.ROOT-SERVERS.NET. 3600000      A       199.9.14.201
B.ROOT-SERVERS.NET. 3600000      AAAA    2001:500:200::b
;
; FORMERLY C.PSI.NET
;
.                3600000      NS      C.ROOT-SERVERS.NET.
C.ROOT-SERVERS.NET. 3600000      A       192.33.4.12
C.ROOT-SERVERS.NET. 3600000      AAAA    2001:500:2::c
;
; FORMERLY TERP.UMD.EDU
;
.                3600000      NS      D.ROOT-SERVERS.NET.
D.ROOT-SERVERS.NET. 3600000      A       199.7.91.13
D.ROOT-SERVERS.NET. 3600000      AAAA    2001:500:2d::d
;
; FORMERLY NS.NASA.GOV
;
```

- **named.conf.local:** es el fichero de configuración de las zonas que va a crear el administrador, es decir, en este fichero es donde vamos a definir las zonas de nuestra organización. De cada zona se debe tener su nombre, su tipo y el nombre y ubicación del archivo de zona. Verás que su contenido inicial está vacío.

Puedes reforzar sobre los archivos de configuración en:

- [recursos-4.2] Fichero ServidorDNS.pdf- Apartado “Archivos de configuración” (pág 30 y 31).
- En las otras referencias bibliográficas puedes contrastar la información anterior.

## ENTREGA:

DOCUMENTACIÓN: Nada

Instrumento de Evaluación: prueba/evaluación en clase 100%.

**(Ampliar conocimientos): . Tipos de registros.**

Antes de realizar la creación de las zonas vamos a conocer los diferentes tipos de registros que puede tener un archivo de zona: SOA, NS, A, CNAME, MX, PTR, principalmente.

Tipo	Función
SOA	Especifica el inicio de autoridad sobre la zona, es decir, el servidor autoritario en la zona y para darle unos parámetros de configuración. Debe ser el primer registro de la zona.
NS	Sirve para especificar que un equipo es servidor de nombres en la zona.
A	Sirve para resolver el nombre de dominio de un equipo y obtener su dirección IPv4.
AAAA	Sirve para resolver el nombre de dominio de un equipo y obtener su dirección IPv6.
CNAME	Permite crear un alias o nombre alternativo para otro nombre DNS de equipo.
MX	Sirve para especificar un servidor de intercambio de correo para el dominio.
PTR	Permite realizar resoluciones inversas, es decir, obtener un nombre de equipo a partir de su nombre DNS.

Usa la siguiente documentación para reforzar lo anterior.

- [Recurso 4.2] Fichero ServidorDNS.pdf- Apartado “Registro de recursos” (páginas 32 y 33) y el apartado “Tipos de registros” (página 34).
- [Recurso 45] Apartado “12.3.2. Registros de recursos de archivos de zona”

Ejercicio 3. Configuración de una zona directa. Registros SOA, NS, A

Ya tenemos el servicio en funcionamiento y conocemos los tipos de registros principales que pueden configurarse en el servidor DNS. Vamos ahora a realizar una configuración de registros para un dominio ejemplo. El nombre será “**ejemplo.com**” y vamos a crear diversos registros (en este ejercicio vamos a usar los registros tipo SOA, NS y A). Así, vamos a definir:

Fichero /etc/bind/named.conf.local → definimos la zona “ejemplo.com” de tipo maestro y los registros están definidos en el fichero db.ejemplo.com ubicado en /var/lib/bind.

```
zone "ejemplo.com"
{
    type master;
    file "/var/lib/bind/db.ejemplo.com";
};
```

Fichero: /var/lib/bind/db.ejemplo.com → contenido de los registros de la zona.

```
$TTL 604800
ejemplo.com. IN SOA ns1 admin (
    2      ; Serial
    604800 ; Refresh
    86400  ; Retry
    2419200 ; Expire
    604800 ) ; Negative Cache TTL

; Servidores de nombre
@ IN NS ns1.ejemplo.com.

ns1.ejemplo.com. IN A 127.0.0.1
pc1 IN A 192.168.3.11
```

Explicuemos la configuración:

- Toda zona debe contener un registro SOA (marcado en azul). La sintaxis de un registro SOA es:

```
@ IN SOA <primary-name-server> <hostmaster-email> (
    <serial-number>
    <time-to-refresh>
    <time-to-retry>
    <time-to-expire>
    <minimum-TTL> )
```

- serial-number = Número de Serie: La versión de ese archivo. Aumenta cada vez que el archivo cambia.
- time-to-refresh = Actualizar. Es la frecuencia con la que los servidores de nombres secundarios verifican el servidor de nombres primarios para ver si se han realizado cambios en el archivo de zona de dominio. Es decir, es el tiempo que espera un servidor de nombres secundario para ver si el archivo ha cambiado, y por lo tanto pedir una **transferencia de zona**.
- time-to-retry = Volver a Intentar. Esto es el tiempo que un servidor secundario espera para recuperar una transferencia de zona fallida. Este tiempo suele ser menor al intervalo de actualización.
- time-to-expire = Expirar. Tiempo que el servidor de nombres secundario intentará descargar una zona. Cuando pase, se rechaza la información antigua.
  - En resumen, la directiva <time-to-retry> es un valor numérico usado por los servidores esclavo para determinar el intervalo de tiempo que tiene que esperar antes de emitir una petición de actualización de datos en caso de que el servidor de nombres maestro no

responda. Si el servidor maestro no ha respondido a una petición de actualización de datos antes que se acabe el intervalo de tiempo <time-to-expire>, los servidores esclavo paran de responder como una autoridad por peticiones relacionadas a ese espacio de nombres.

- minimum-TTL = Mínimo. cantidad de tiempo que otros servidores de nombres guardan en caché la información de zona.
- primary-name-server: Es el nombre del servidor DNS primario de la zona. Es el nombre del host del servidor de nombres que tiene autoridad para este dominio
- Hostmaster-email : Persona responsable; contiene la dirección de correo electrónico de la persona responsable de la zona. Ejemplos equivalentes:
  - Si se escribe admin → equivale a [admin@ejemplo.com](mailto:admin@ejemplo.com)
  - Si se escribe [admin.ejemplo.com](mailto:admin.ejemplo.com). (observa el punto final, y también que el primer punto equivale a la @) → equivale a [admin@ejemplo.com](mailto:admin@ejemplo.com)
- Podemos sustituir @ por el nombre del dominio. El símbolo @ coloca la directiva \$ORIGIN (o el nombre de la zona, si la directiva \$ORIGIN no está configurada) como el espacio de nombres que está siendo definido por este registro de recursos SOA. Así, también sería válido:

```
ejemplo.com.      IN      SOA      PC11  admin (
                  2          ; Serial
                  604800     ; Refresh
                  86400      ; Retry
                  2419200    ; Expire
                  604800 )   ; Negative Cache TTL
```

- NOTA: retomaremos este fichero en la actividad de transferencias de zonas.
- Hemos definido 2 registros de tipo A (marcado en verde):
  - ns1.ejemplo.com. (observa que el nombre DNS acaba en punto). Este registro se corresponde con el nombre “ns1.ejemplo.com” (ahora NO lleva punto).
  - pc1 (observa que no lleva punto, lo que se traduce que este registro se corresponde con el nombre “pc1.ejemplo.com”, es decir, si no lleva punto el nombre DNS real es el resultado de concatenar el nombre indicado y el nombre del dominio (marcado en azul negrita del registro SOA). El registro “ns1.ejemplo.com.” también podría haberse declarado como “ns1”.
- Se ha definido un registro de tipo NS (marcado en naranja):
  - ns1.ejemplo.com. → indica que este equipo es el SERVIDOR DE NOMBRES del dominio.
  - Otras sintaxis alternativas:
    - Podríamos eliminar @ dejándolo vacío, es decir:

```
IN      NS      ns1.ejemplo.com.
```

- O bien indicar el propio nombre de dominio:

```
ejemplo.com.  IN      NS      ns1.ejemplo.com.
```

Vamos a realizar pruebas de resolución en el propio Debian5-Pruebas; es decir, vamos a consultar nombres en Debian5-Pruebas usando el servidor DNS ubicado en Debian5-Pruebas. Para ello debemos configurar el servidor DNS que usará Debian5-Pruebas que va a ser él mismo. Así el fichero /etc/resolv.conf de Debian5-Pruebas tendrá la directiva “nameserver 127.0.0.1”. Una vez realizada esta pequeña configuración realicemos pruebas usando el cliente DNS ‘host’ (viene por defecto instalado)

1. Prueba 1: resolvamos el nombre pc1.ejemplo.com para el cual se creó un registro tipo A igual a 192.168.3.11.

```
usuario@Debian5-Pruebas:~$ host pc1.ejemplo.com
pc1.ejemplo.com has address 192.168.3.11
usuario@Debian5-Pruebas:~$
```

2. Prueba 2: probemos ahora con ns1.ejemplo.com

```
usuario@Debian5-Pruebas:~$ host ns1.ejemplo.com
ns1.ejemplo.com has address 127.0.0.1
usuario@Debian5-Pruebas:~$
```

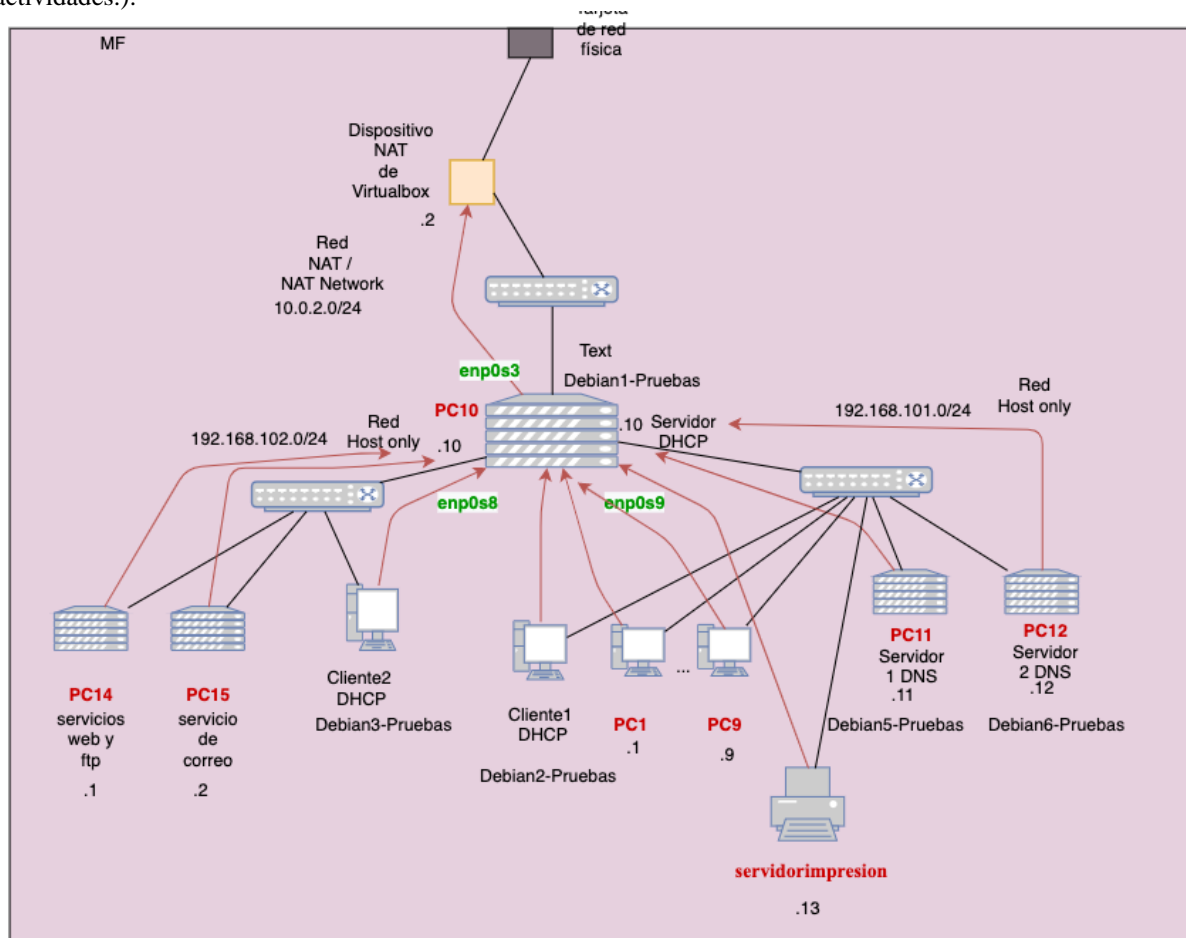
3. Prueba 3: y si le pasamos una dirección IP, ¿nos devolverá el nombre asociado?

```
usuario@Debian5-Pruebas:~$ host 192.168.3.11
Host 11.3.168.192.in-addr.arpa. not found: 3(NXDOMAIN)
usuario@Debian5-Pruebas:~$
```

Vemos que la respuesta no ha sido positiva. ¿Por qué, si tenemos registrados tanto el nombre DNS como su dirección IP? Encontrarás la respuesta en el siguiente ejercicio.

Vemos que la respuesta no ha sido positiva. ¿Por qué, si tenemos registrados tanto el nombre DNS como su dirección IP? Encontrarás la respuesta en el siguiente ejercicio.

1. Prueba 4. Adapta la configuración del servidor DNS para disponer de los siguientes registros según este el siguiente escenario (todos son de tipo A; la consulta de los registros SOA y NS lo veremos en próximas actividades.):



- a. Elimina los registros A asociados al ejemplo mostrado anteriormente (pc1 y ns1).

- b. La red 192.168.101.0/24 tiene 9 estaciones de trabajo con IP estáticas nombrados como *PCx.ejemplo.com* (x es un número comprendido entre 1 y 9, inclusivos). Estos equipos tienen la IP 192.168.101.x (el mismo número que contiene el nombre; así PC1.ejemplo.com es 192.168.101.1, y así con el resto). También hay 1 impresora nombrada como *servidorimpresion.ejemplo.com* con IP 192.168.101.13.
- c. Configura los registros NS como *PC11.ejemplo.com* (.11) y *PC12.ejemplo.com* (.12).
- d. La red 192.168.102.0/24 alberga varios servidores:
- Un servidor web y FTP: *PC14.ejemplo.com* (192.168.102.1) → el mismo servidor alberga los 2 servicios).
  - Un servidor de correo: *PC15.ejemplo.com*. (192.168.102.2).
- e. Debian1-Pruebas será accedido por el nombre *PC10.ejemplo.com*. Este equipo tiene varias direcciones IP (una por cada red host-only y la que obtiene por DHCP en la red NAT). Configura un par de registros de tipo A para este nombre.

```
$TTL 604800
ejemplo.com.      IN      SOA      PC11  admin (
                    2      ; Serial
                    604800 ; Refresh
                    86400  ; Retry
                    2419200; Expire
                    604800 )      ; Negative Cache TTL

; Servidores de nombre
@      IN      NS      PC11.ejemplo.com.
      IN      NS      PC12.ejemplo.com.

; Estaciones de trabajo de la red 192.168.101.0/24
PC1     IN      A       192.168.101.1
PC2     IN      A       192.168.101.2
PC3     IN      A       192.168.101.3
PC4     IN      A       192.168.101.4
PC5     IN      A       192.168.101.5
PC6     IN      A       192.168.101.6
PC7     IN      A       192.168.101.7
PC8     IN      A       192.168.101.8
PC9     IN      A       192.168.101.9

; Servidor DHCP + Router. Debian1-Pruebas
PC10    IN      A       192.168.101.10
      IN      A       192.168.102.10

; Servidores DNS
PC11    IN      A       192.168.101.11
PC12    IN      A       192.168.101.12

; Servidores de la red 192.168.102.0/24
PC14    IN      A       192.168.102.1
PC15    IN      A       192.168.102.1
```

Haz pruebas de resolución de nombres de estos registros.

```
usuario@Debian5-Pruebas:~$ host PC1.ejemplo.com
PC1.ejemplo.com has address 192.168.101.1
usuario@Debian5-Pruebas:~$ host PC10.ejemplo.com
PC10.ejemplo.com has address 192.168.102.10
PC10.ejemplo.com has address 192.168.101.10
usuario@Debian5-Pruebas:~$ host servidorimpresion.ejemplo.com
servidorimpresion.ejemplo.com has address 192.168.101.13
usuario@Debian5-Pruebas:~$ host pc14.ejemplo.com
PC14.ejemplo.com has address 192.168.102.1
usuario@Debian5-Pruebas:~$ host pc15.ejemplo.com
PC15.ejemplo.com has address 192.168.102.1
usuario@Debian5-Pruebas:~$
```

- Realicemos una prueba para verificar que el servidor DNS tiene bien configurado su registro SOA:

```
usuario@Debian5-Pruebas:~$ host -t SOA ejemplo.com
ejemplo.com has SOA record PC11.ejemplo.com. admin.ejemplo.com. 2 604800 86400 2419200
604800
usuario@Debian5-Pruebas:~$
```

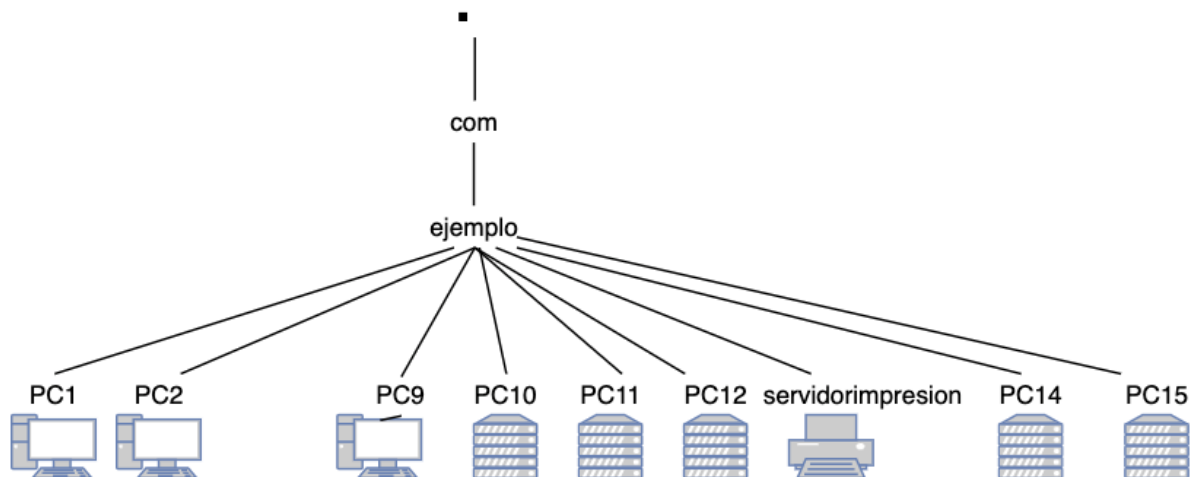
- Y los registros NS:



```

usuario@Debian5-Pruebas:~$ host -t NS ejemplo.com
ejemplo.com name server PC11.ejemplo.com.
ejemplo.com name server PC12.ejemplo.com.
usuario@Debian5-Pruebas:~$
    
```

Podemos ver los nombres que hemos configurado están organizados en este espacio de nombres (en la imagen). **IMPORTANTE: antes de realizar ninguna configuración en el-los fichero-os de zona hay que analizar los nombres DNS que queremos que tengan los equipos de la red y dibujarlos en forma de árbol formando la figura denominada.**



### Directivas \$INCLUDE, \$ORIGIN, \$TTL

Vamos a poner en escena estas directivas:

#### \$TTL

- El concepto tiempo de vida (TTL, Time To Live) de los RR (Registros de Recurso) significa el tiempo, medido en segundos, que un RR estará en la caché de los servidores de nombres antes de borrarse. Es decir, cuando un servidor de nombres remoto hace una consulta, dicha consulta se guarda en caché durante el tiempo especificado por esta directiva.
- Todos los RR pueden llevar opcionalmente un TTL cuando se crean, pero en caso de que no se especifique, su TTL vendrá dado por el valor de la directiva \$TTL. Es decir, cada recurso puede contener su propio valor TTL, el cual ignora esta directiva.
- Por lo tanto, la directiva \$TTL especifica el tiempo de vida por defecto de los RR y es obligatoria.

```

$TTL      604800
ejemplo.com.      IN      SOA      PC11  admin (
                    2      ; Serial
                    604800 ; Refresh
                    86400  ; Retry
                    2419200 ; Expire
                    604800 ) ; Negative Cache TTL

; Servidores de nombre
@          IN      NS      PC11.ejemplo.com.
          IN      NS      PC12.ejemplo.com.

; Estaciones de trabajo de la red 192.168.101.0/24
PC1        3600    IN      A      192.168.101.1
PC2        3600    IN      A      192.168.101.2
PC3        3600    IN      A      192.168.101.3
PC4        3600    IN      A      192.168.101.4
PC5        3600    IN      A      192.168.101.5
PC6        1h      IN      A      192.168.101.6
PC7        1h      IN      A      192.168.101.7
PC8        1h      IN      A      192.168.101.8
PC9        1h      IN      A      192.168.101.9

```

La imagen anterior muestra que los registros PC1 a PC9 tienen un TTL de 1 hora (observar que se ha puesto en formato de segundos y también en horas). El resto de registros, para los que no haya definido un valor TTL, usarán el valor general definido en la directiva \$TTL, el cual es 604800 segundos (también podríamos haber indicado \$TTL 7d o \$TTL 1w, es decir el valor de los RR sin declaración del TTL se guardarán durante una semana).

[Más información.](#)

### \$INCLUDE

La directiva \$INCLUDE permite que la configuración del fichero de zona pueda ser distribuida en varios ficheros, normalmente temáticos desde algún punto de vista, y luego incluirlos todos en el fichero de zona. Esta forma de trabajar es más importante cuanto mayor es el fichero de zona. Es una directiva opcional.

Supongamos que PC1 y PC2 son equipos del departamento de Recursos Humanos y PC3 a PC9 del departamento de Administración. Queremos definir estos registros en ficheros diferentes (en este caso no tiene mucho sentido ya que son pocos registros, pero imagina que hay 1000 registros en Recursos Humanos y otros tantos en Administración: en estas situaciones tiene sentido no tener todos los registros en un único fichero para hacerlo menos pesado). El resto de registros lo definiremos en el fichero principal de zona. La [imagen](#) ilustra la explicación anterior.

Así podemos redefinir la configuración como:

```

GNU nano 3.2 db.ejemplo.com
$TTL 604800
ejemplo.com. IN SOA PC11 admin (
                        2      ; Serial
                        604800 ; Refresh
                        86400  ; Retry
                        2419200 ; Expire
                        604800 ) ; Negative Cache TTL

; Servidores de nombre
@ IN NS PC11.ejemplo.com.
  IN NS PC12.ejemplo.com.

; Estaciones de trabajo de la red 192.168.101.0/24
$INCLUDE /var/lib/bind/db.ejemplorrh.com
$INCLUDE /var/lib/bind/db.ejemploadmin.com

; Servidor DHCP + Router.- Debian1-Pruebas
PC10 IN A 192.168.101.10
     IN A 192.168.102.10

; Servidores DNS
PC11 IN A 192.168.101.11
PC12 IN A 192.168.101.12

; Servidores de la red 192.168.102.0/24
PC14 IN A 192.168.102.1
PC15 IN A 192.168.102.2

```

```

GNU nano 3.2 db.ejemplorrh.com
; Estaciones de trabajo del dpto RRHH
PC1 3600 IN A 192.168.101.1
PC2 3600 IN A 192.168.101.2

```

```

GNU nano 3.2 db.ejemploadmin.com
; Estaciones de trabajo del dpto Administración
PC3 3600 IN A 192.168.101.3
PC4 3600 IN A 192.168.101.4
PC5 3600 IN A 192.168.101.5
PC6 1h IN A 192.168.101.6
PC7 1h IN A 192.168.101.7
PC8 1h IN A 192.168.101.8
PC9 1h IN A 192.168.101.9

```

Realiza pruebas de resolución de registros en los 2 ficheros nuevos y de registros que permanecen en el fichero principal.

Más [información](#)

**Nota:** No te olvides de realizar el commit.

\$ORIGIN

- La directiva \$ORIGIN define el nombre del dominio que será añadido al final de cualquier nombre que no acabe en punto (nombres relativos o no cualificados) en los RR, para así transformarlos en nombres FQDN (fully qualified domain name). Si un nombre acaba en punto, ya sería un nombre FQDN y no se utilizaría \$ORIGIN.
- La directiva \$ORIGIN no es obligatoria, y en el caso de que no esté presente se asumirá que su valor es el del dominio descrito por el fichero de zona, que en BIND será el nombre que se encuentra en la cláusula zone del fichero de configuración named.conf.local. Es aconsejable usarla siempre, al menos al principio del fichero de zona, de forma que este queda así más descriptivo y portable, pues puede que algún software DNS no admita su ausencia.

```
GNU nano 3.2 db.ejemplo.com
$TTL 604800
$ORIGIN ejemplo.com.
ejemplo.com. IN SOA PC11 admin (
                2      ; Serial
                604800 ; Refresh
                86400  ; Retry
                2419200 ; Expire
                604800 ) ; Negative Cache TTL

; Servidores de nombre
@ IN NS PC11.ejemplo.com.
  IN NS PC12.ejemplo.com.

; Estaciones de trabajo de la red 192.168.101.0/24
$INCLUDE /var/lib/bind/db.ejemplorrh.com
$INCLUDE /var/lib/bind/db.ejemploadmin.com

; Servidor DHCP + Router.- Debian1-Pruebas
PC10 IN A 192.168.101.10
     IN A 192.168.102.10

; Servidores DNS
PC11 IN A 192.168.101.11
PC12 IN A 192.168.101.12

; Servidores de la red 192.168.102.0/24
PC14.ejemplo.com. IN A 192.168.102.1
PC15 IN A 192.168.102.2
```

La imagen muestra:

- Se ha definido la directiva \$ORIGIN con el dominio ejemplo.com. (observa que se pone un punto al final). Es opcional ya que en el fichero /etc/bind/named.conf.local se definió la zona con este nombre, pero es recomendable ponerlo para mayor claridad.

```
zone "ejemplo.com"
{
    type master;
    file "/var/lib/bind/db.ejemplo.com";
};
```

- Todos los RR se definen usando como nombres relativos o no cualificados (PC1, PC2 ...) por lo que su FQDN se crea con este nombre y el nombre de la zona (ejemplo.com.), que lo coge o del fichero /etc/bind/named.conf.local o bien de la directiva \$ORIGIN ejemplo.com.

- Esta directiva puede ponerse en cualquier parte del fichero de zona y además, se pueden poner más de una, teniendo efecto cada directiva en el trozo del fichero de zona que va desde su aparición hasta la siguiente directiva \$ORIGIN o el final del fichero.

```
GNU nano 3.2 db.ejemploadmin.com

; Estaciones de trabajo del dpto Administración
PC3.recepcion      3600  IN      A      192.168.101.3
PC4.recepcion.ejemplo.com. 3600  IN      A      192.168.101.4

$ORIGIN contabilidad.ejemplo.com.
PC5      3600  IN      A      192.168.101.5
PC6      1h    IN      A      192.168.101.6

$ORIGIN secretaria.ejemplo.com.
PC7      1h    IN      A      192.168.101.7
PC8      1h    IN      A      192.168.101.8
PC9      1h    IN      A      192.168.101.9
```

En la imagen anterior se observa:

- PC3 no tienen asociada una declaración de \$ORIGIN por lo que cogen el valor definido en el fichero principal (\$ORIGIN ejemplo.com.). Al mismo tiempo se ha definido un subdominio (recepcion). Así el FQDN sería PC3.recepcion.ejemplo.com.
- PC4 está en el mismo subdominio que PC3 pero se ha declarado su FQDN.
- El resto de RR están en 2 subdominios diferentes, contabilidad y secretaria.

La configuración anterior responde a este [espacio de nombres](#).

Realiza pruebas de resolución con la nueva configuración de registros.

Más [información](#)

**Nota:** no te olvides de realizar commit

#### Ejercicio 4. Configuración de otros tipos de registro

Al comienzo de este ejercicio debes estar ya familiarizado con los archivos de zonas directa e inversa y haber configurado registros SOA, NS, A y PTR. Ahora vamos a poner en práctica otros 2 tipos de registros: CNAME y MX.

##### Registros CNAME

Los registros CNAME son alias de otros nombres DNS de equipo. ¿Qué significa esto? Podemos dar nombres más significativos a los dispositivos, como por ejemplo a PC10.ejemplo.com, que es nuestro router, podríamos denominarlo como router.ejemplo.com. La siguiente tabla asocia nombres DNS y alias:

Nombre real	Alias	Justificación
PC10.ejemplo.com	router.ejemplo.com	Indicado antes
PC11.ejemplo.com	dns1.ejemplo.com	Es uno de los servidores DNS de nuestra red.
PC12.ejemplo.com	dns2.ejemplo.com	Es el otro servidor DNS.
PC14.ejemplo.com	www.ejemplo.com	Este equipo está pensado para instalar un servidor web.

	ftp.ejemplo.com	Y también un servicio FTP.
PC15.ejemplo.com	smtp.ejemplo.com	El servicio de correo necesita de 2 componentes: uno para recepcionar el correo de los clientes que se corresponde con el servicio SMTP. Crearemos un registro para denotar tal servicio.
	pop.ejemplo.com	Y POP es el otro servicio que permite a los clientes descargarse los correos. Definiremos otro registro para este fin.

El espacio de nombre nuevo es [éste](#).

```

GNU nano 3.2 db.ejemplo.com

$TTL 604800
$ORIGIN ejemplo.com.
ejemplo.com. IN SOA PC11 admin (
                2      ; Serial
                604800 ; Refresh
                86400  ; Retry
                2419200 ; Expire
                604800 ) ; Negative Cache TTL

; Servidores de nombre
ejemplo.com. IN NS PC11.ejemplo.com.
              IN NS PC12.ejemplo.com.

; Estaciones de trabajo de la red 192.168.101.0/24
$INCLUDE /var/lib/bind/db.ejemplorrh.com
$INCLUDE /var/lib/bind/db.ejemploadmin.com

; Servidor DHCP + Router.- Debian1-Pruebas
PC10 IN A 192.168.101.10
      IN A 192.168.102.10

router IN CNAME PC10

; Servidores DNS
PC11 IN A 192.168.101.11
dns1 IN CNAME PC11
PC12 IN A 192.168.101.12
dns2 IN CNAME PC12

; Servidores de la red 192.168.102.0/24
PC14 ejemplo.com IN A 192.168.102.1
www IN CNAME PC14
ftp IN CNAME PC14

```

Realiza pruebas de resolución con la nueva configuración de registros. Por ejemplo:

```

usuario@Debian5-Pruebas:~$ host router.ejemplo.com
router.ejemplo.com is an alias for PC10.ejemplo.com.
PC10.ejemplo.com has address 192.168.101.10
PC10.ejemplo.com has address 192.168.102.10
usuario@Debian5-Pruebas:~$ host dns1.ejemplo.com
dns1.ejemplo.com is an alias for PC11.ejemplo.com.
PC11.ejemplo.com has address 192.168.101.11

```

Más información:

- [BIB-5] Apartado “12.3.2. Registros de recursos de archivos de zona” (pág 34) apartado “12.3.3. Ejemplo de archivo de zonas”.

**Nota:** no te olvides de realizar commit + sincronizar a los repositorios remotos.

### Registros MX

Por último veamos los registros tipo MX. Estos registros sirven para identificar a los servidores de correo de un dominio. Estos registros los entenderás mejor cuando hayas trabajado la unidad de correo electrónico. Lee los mismos documentos de antes (los indicados para CNAME).

Para que entiendas el sentido de estos registros piensa en el siguiente caso:

- Supongamos que tu cuenta es “[usuario@hotmail.com](mailto:usuario@hotmail.com)”. Cuando envías un correo a “[usuario@gmail.com](mailto:usuario@gmail.com)”, este mensaje debe ser dirigido a un servidor de correo del dominio gmail.com.
- Pasos:
  - Lo primero es que tu equipo envía el correo a tu servidor de correo, es decir, al servidor ubicado en “hotmail.com”.
  - En segundo lugar, ¿cómo sabe el servidor de correo de “hotmail.com” cuál es el servidor (o servidores) de gmail? La respuesta está en los registros MX; el servidor de correo en “hotmail.com” hace una consulta de registro MX del dominio gmail.com y devolverá la IP que se corresponde con el servidor de correo. Sencillo, ¿verdad? No te agobies, lo entenderás mejor en la unidad de correo electrónico.

Esta [figura](#) puede ayudarte.

En el ejemplo que estamos desarrollando, PC15.ejemplo.com, va a ser el servidor de correo de nuestra red. Sólo tenemos un servidor de correo en nuestra red, pero lo suyo es disponer de más de uno (por tolerancia a fallos y balanceo de carga). Imaginemos que hay otro equipo PC16.ejemplo.com que actúa de servidor de correo y definido con el alias smtp2.ejemplo.com.

```

GNU nano 3.2                                db.ejemplo.com
$TTL 604800
$ORIGIN ejemplo.com.
ejemplo.com.      IN      SOA      PC11  admin (
                        2      ; Serial
                        604800 ; Refresh
                        86400  ; Retry
                        2419200 ; Expire
                        604800 ) ; Negative Cache TTL

; Servidores de nombre
ejemplo.com.      IN      NS      PC11.ejemplo.com.
ejemplo.com.      IN      NS      PC12.ejemplo.com.

; Servidores de correo
ejemplo.com.      IN      MX      10  smtp.ejemplo.com.
ejemplo.com.      IN      MX      20  smtp2.ejemplo.com.

; Estaciones de trabajo de la red 192.168.101.0/24
$INCLUDE /var/lib/bind/db.ejemplorrh.com
$INCLUDE /var/lib/bind/db.ejemploadmin.com

; Servidor DHCP + Router.- Debian1-Pruebas
PC10      IN      A      192.168.101.10
PC10      IN      A      192.168.102.10

router     IN      CNAME   PC10

; Servidores DNS
PC11      IN      A      192.168.101.11
dns1      IN      CNAME   PC11
PC12      IN      A      192.168.101.12
dns2      IN      CNAME   PC12

; Servidores de la red 192.168.102.0/24
PC14.ejemplo.com. IN      A      192.168.102.1
www        IN      CNAME   PC14
ftp        IN      CNAME   PC14

PC15      IN      A      192.168.102.2
smtp      IN      CNAME   PC15
pop       IN      CNAME   PC15

PC16      IN      A      192.168.102.3
smtp2     IN      CNAME   PC16

```

Realiza pruebas de resolución con la nueva configuración de registros. Por ejemplo:

```

usuario@Debian5-Pruebas:~$ host -t MX ejemplo.com
ejemplo.com mail is handled by 20 smtp2.ejemplo.com.
ejemplo.com mail is handled by 10 smtp.ejemplo.com.

```

**Nota:** no te olvides de realizar commit

### Ejercicio 5. Clientes DNS. Probando los servidores DNS

Por último vamos a revisar y profundizar sobre los clientes DNS y cómo realizar consultas a los servidores DNS. Hasta ahora hemos estado usando la utilidad 'host'. Vamos a conocer otros como son nslookup y dig. Exploremos algunas posibilidades que nos ofrecen estos clientes DNS a través de ejemplos. La utilidad 'host' viene por defecto en sistemas Debian; debemos instalar el paquete 'dnsutils' para disponer de 'nslookup' y 'dig'.

#### nslookup

- Podemos realizar consultas en modo interactivo y no interactivo.
- Modo no interactivo: podemos realizar múltiples consultas sin necesidad de repetir el comando 'nslookup'. Ejemplos:

```

usuario@Debian5-Pruebas:~$ nslookup
> set type=SOA
> ejemplo.com
Server:      127.0.0.1
Address:     127.0.0.1#53

ejemplo.com
    origin = PC11.ejemplo.com
    mail addr = admin.ejemplo.com
    serial = 2
    refresh = 604800
    retry = 86400
    expire = 2419200
    minimum = 604800

> set type=mx
> ejemplo.com
Server:      127.0.0.1
Address:     127.0.0.1#53

ejemplo.com    mail exchanger = 10 smtp.ejemplo.com.
ejemplo.com    mail exchanger = 20 smtp2.ejemplo.com.

> set type=ns
> ejemplo.com
Server:      127.0.0.1
Address:     127.0.0.1#53

ejemplo.com    nameserver = PC11.ejemplo.com.
ejemplo.com    nameserver = PC12.ejemplo.com.

```

Entramos en modo interactivo al escribir 'nslookup'.

No1: consultamos el registro SOA de la zona ejemplo.com

No2: consultamos los servidores de correo de la zona.

No3: consultamos los servidores DNS de la zona.



```

> set type=A
ejemplo.com
Server:      127.0.0.1
Address:     127.0.0.1#53

*** Can't find ejemplo.com: No answer
> pc1.ejemplo.com
Server:      127.0.0.1
Address:     127.0.0.1#53

Name:  PC1.ejemplo.com
Address: 192.168.101.1
> set type=PTR
> 192.168.102.3
Server:      127.0.0.1
Address:     127.0.0.1#53

** server can't find 3.102.168.192.in-addr.arpa: NXDOMAIN
> 192.168.101.2
Server:      127.0.0.1
Address:     127.0.0.1#53

2.101.168.192.in-addr.arpa      name = PC2.ejemplo.com.

```

No4: consulta errónea de registro tipo A al indicar el nombre de la zona y no de un registro concreto como sí aparece en No5.

No6: Consulta de registro PTR.

- Modo no interactivo

- Consulta del registro SOA de una zona. Observa que una vez realizada la consulta se vuelve a mostrar el prompt del sistema (a diferencia de interactivo).

```

usuario@Debian5-Pruebas:~$ nslookup -type=soa ejemplo.com
Server:      127.0.0.1
Address:     127.0.0.1#53

ejemplo.com
    origin = PC11.ejemplo.com
    mail addr = admin.ejemplo.com
    serial = 2
    refresh = 604800
    retry = 86400
    expire = 2419200
    minimum = 604800

usuario@Debian5-Pruebas:~$ █

```

- Consulta de los servidores de correo:

```

usuario@Debian5-Pruebas:~$ nslookup -type=any ejemplo.com
Server:      127.0.0.1
Address:     127.0.0.1#53

ejemplo.com
    origin = PC11.ejemplo.com
    mail addr = admin.ejemplo.com
    serial = 2
    refresh = 604800
    retry = 86400
    expire = 2419200
    minimum = 604800
ejemplo.com    nameserver = PC11.ejemplo.com.
ejemplo.com    nameserver = PC12.ejemplo.com.
ejemplo.com    mail exchanger = 20 smtp2.ejemplo.com.
ejemplo.com    mail exchanger = 10 smtp.ejemplo.com.

usuario@Debian5-Pruebas:~$ nslookup -type=mx ejemplo.com
Server:      127.0.0.1
Address:     127.0.0.1#53

ejemplo.com    mail exchanger = 10 smtp.ejemplo.com.
ejemplo.com    mail exchanger = 20 smtp2.ejemplo.com.

usuario@Debian5-Pruebas:~$ █

```

- Consulta de todos los registros:
- Haz tus propias pruebas.

# SISTEMAS INFORMÁTICOS. 1º DAM. UD9. Práctica 3. DNS

○ Sergio:

The image displays four screenshots from the Oracle VM VirtualBox Administrator interface, illustrating the steps to create a new virtual machine and configure its network settings.

**Top Left Screenshot:** The "Clonar máquina virtual" (Clone virtual machine) dialog box is shown. The "Nuevo nombre de máquina y ruta" (New name and path) section is highlighted. The "Nombre" (Name) field is set to "SSINF-Debian-Base-4-PRUEBA", and the "Ruta" (Path) is set to "C:\Users\Sergio DC\VirtualBox VMs". The "Política de dirección MAC" (MAC address policy) is set to "Generar nuevas direcciones MAC para todos los adaptadores de red" (Generate new MAC addresses for all network adapters).

**Top Right Screenshot:** The "Clonar máquina virtual" dialog box is shown again, with the "Nombre" field set to "SSINF-Debian-Base-5-PRUEBA" and the "Ruta" field set to "C:\Users\Sergio DC\VirtualBox VMs". The "Política de dirección MAC" is set to "Generar nuevas direcciones MAC para todos los adaptadores de red".

**Bottom Left Screenshot:** The "Configuración" (Settings) window for the "SSINF-Debian-Base-5-PRUEBA" virtual machine is shown. The "Red" (Network) settings are highlighted. The "Adaptador 1" (Adapter 1) tab is selected. The "Habilitar adaptador de red" (Enable network adapter) checkbox is checked. The "Conectado a" (Connected to) dropdown is set to "Adaptador sólo-anfitrión" (Host-only adapter). The "Nombre" (Name) field is set to "VirtualBox Host-Only Ethernet Adapter".

**Bottom Right Screenshot:** The "SSINF-Debian-Base-5-PRUEBA [Corriendo]" (Running) window is shown. The terminal displays the output of the command "hostname", which is "debian5-Pruebas".

```
SSINF-Debian-Base-5-PRUEBA [Corriendo]...
Archivo Máquina Ver Entrada Dispositivos Ayuda

root@debian1-Pruebas:/etc# cd network
root@debian1-Pruebas:/etc/network# sudo nano interfaces_
```

```
SSINF-Debian-Base-5-PRUEBA [Corriendo]...
Archivo Máquina Ver Entrada Dispositivos Ayuda

GNU nano 3.2 interfaces Modified

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug enp0s3
iface enp0s3 inet static
```

```
SSINF-Debian-Base-5-PRUEBA [Corriendo]...
Archivo Máquina Ver Entrada Dispositivos Ayuda

GNU nano 3.2 interfaces Modified

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug enp0s3
iface enp0s3 inet static

    address 192.168.101.11
    netmask 255.255.255.0
    gateway 192.168.101.10
```

```
SSINF-Debian-Base-5-PRUEBA [Corriendo]...
Archivo Máquina Ver Entrada Dispositivos Ayuda

root@debian1-Pruebas:/etc/network# sudo service networking restart
root@debian1-Pruebas:/etc/network# ifconfig
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state DOWN group default qlen 1000
    link/ether 08:00:27:70:aa:28 brd ff:ff:ff:ff:ff:ff
root@debian1-Pruebas:/etc/network# reboot
```

```
SSINF-Debian-Base-5-PRUEBA [Corriendo]...
Archivo Máquina Ver Entrada Dispositivos Ayuda

root@debian5-Pruebas:/# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:70:aa:28 brd ff:ff:ff:ff:ff:ff
    inet 192.168.101.11/24 brd 192.168.101.255 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe70:aa28/64 scope link
        valid_lft forever preferred_lft forever
root@debian5-Pruebas:/# ping 8.8.8.8_
```

```
SSINF-Debian-Base-5-PRUEBA [Corriendo]...
Archivo Máquina Ver Entrada Dispositivos Ayuda

root@debian5-Pruebas:/# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:70:aa:28 brd ff:ff:ff:ff:ff:ff
    inet 192.168.101.11/24 brd 192.168.101.255 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe70:aa28/64 scope link
        valid_lft forever preferred_lft forever
root@debian5-Pruebas:/# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=113 time=12.7 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=113 time=12.10 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=113 time=13.10 ms
^C
--- 8.8.8.8 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 5ms
rtt min/avg/max/mdev = 12.683/13.218/13.982/0.569 ms
root@debian5-Pruebas:/#
```

```

SSINF-Debian-Base-5-PRUEBA [Corriendo]...
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda

Preconfiguring packages ...
Selecting previously unselected package python3-ply.
(Reading database ... 28475 files and directories currently installed.)
Preparing to unpack .../python3-ply_3.11-3_all.deb ...
Unpacking python3-ply (3.11-3) ...
Selecting previously unselected package bind9utils.
Preparing to unpack .../bind9utils_1:9.11.5.P4+dfsg-5.1+deb10u3_and64.deb ...
Unpacking bind9utils (1:9.11.5.P4+dfsg-5.1+deb10u3) ...
Selecting previously unselected package dns-root-data.
Preparing to unpack .../dns-root-data_2019031302_all.deb ...
Unpacking dns-root-data (2019031302) ...
Selecting previously unselected package net-tools.
Preparing to unpack .../net-tools_1.60+git20180626.aebd88e-1_and64.deb ...
Unpacking net-tools (1.60+git20180626.aebd88e-1) ...
Selecting previously unselected package bind9.
Preparing to unpack .../bind9_1:9.11.5.P4+dfsg-5.1+deb10u3_and64.deb ...
Unpacking bind9 (1:9.11.5.P4+dfsg-5.1+deb10u3) ...
Setting up net-tools (1.60+git20180626.aebd88e-1) ...
Setting up python3-ply (3.11-3) ...
Setting up dns-root-data (2019031302) ...
Setting up bind9utils (1:9.11.5.P4+dfsg-5.1+deb10u3) ...
Setting up bind9 (1:9.11.5.P4+dfsg-5.1+deb10u3) ...
Adding group 'bind' (GID 112) ...
Done.
Adding system user 'bind' (UID 106) ...
Adding new user 'bind' (UID 106) 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 (2.8.3-2) ...
Processing triggers for systemd (241-7+deb10u6) ...
root@debian5-Pruebas:/# sudo apt install bind9

```

```

SSINF-Debian-Base-5-PRUEBA [Corriendo]...
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda

root@debian5-Pruebas:/# sudo service bind9 status
sudo: unable to resolve host debian5-Pruebas: Name or service not known
bind9.service - BIND Domain Name Server
Loaded: loaded (/lib/systemd/system/bind9.service; enabled; vendor preset: enabled)
Active: active (running) since Fri 2021-04-30 14:15:59 CEST; 1min 9s ago
       Docs: man:bind(8)
Main PID: 660 (named)
Tasks: 4 (limit: 1149)
Memory: 13.9M
CGroup: /system.slice/bind9.service
        └─660 /usr/sbin/named -u bind

Apr 30 14:15:59 debian5-Pruebas named[660]: network unreachable resolving './DNSKEY/IN': 2001:7fe::5
Apr 30 14:15:59 debian5-Pruebas named[660]: network unreachable resolving './NS/IN': 2001:7fe::53#53
Apr 30 14:15:59 debian5-Pruebas named[660]: network unreachable resolving './DNSKEY/IN': 2001:500:20::b
Apr 30 14:15:59 debian5-Pruebas named[660]: network unreachable resolving './NS/IN': 2001:500:200::b
Apr 30 14:15:59 debian5-Pruebas named[660]: network unreachable resolving './DNSKEY/IN': 2001:500:2::c#5
Apr 30 14:15:59 debian5-Pruebas named[660]: network unreachable resolving './NS/IN': 2001:503:c2::c
Apr 30 14:15:59 debian5-Pruebas named[660]: network unreachable resolving './DNSKEY/IN': 2001:503:c27::2
Apr 30 14:15:59 debian5-Pruebas named[660]: managed-keys-zone: Key 20326 for zone . acceptance timer
Apr 30 14:15:59 debian5-Pruebas named[660]: resolver priming query complete

root@debian5-Pruebas:/#

```

```

SSINF-Debian-Base-5-PRUEBA [Corriendo]...
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda

root@debian5-Pruebas:/etc/bind# ls -la
total 56
drwxr-xr-x  2 root root 4096 Apr 30 14:15 .
drwxr-xr-x 74 root root 4096 Apr 30 14:15 ..
-rw-r--r-- 1 root root 2761 Feb 15 08:51 bind.keys
-rw-r--r-- 1 root root  237 Feb 15 08:51 db.0
-rw-r--r-- 1 root root  271 Feb 15 08:51 db.127
-rw-r--r-- 1 root root  237 Feb 15 08:51 db.255
-rw-r--r-- 1 root root  353 Feb 15 08:51 db.empty
-rw-r--r-- 1 root root  270 Feb 15 08:51 db.local
-rw-r--r-- 1 root root  463 Feb 15 08:51 named.conf
-rw-r--r-- 1 root root  498 Feb 15 08:51 named.conf.default-zones
-rw-r--r-- 1 root root  165 Feb 15 08:51 named.conf.local
-rw-r--r-- 1 root root  846 Feb 15 08:51 named.conf.options
-rw-r--r-- 1 bind bind  77 Apr 30 14:15 rndc.key
-rw-r--r-- 1 root root 1317 Feb 15 08:51 zones.rfc1918
root@debian5-Pruebas:/etc/bind#

```

```

SSINF-Debian-Base-5-PRUEBA [Corriendo]...
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda

root@debian5-Pruebas:/# named-checkconf
root@debian5-Pruebas:/#

```

```

SSINF-Debian-Base-5-PRUEBA [Corriendo]...
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda

-rw-r--r-- 1 root bind 498 Feb 15 08:51 named.conf.default-zones
-rw-r--r-- 1 root bind 165 Feb 15 08:51 named.conf.local
-rw-r--r-- 1 root bind 846 Feb 15 08:51 named.conf.options
-rw-r--r-- 1 bind bind 77 Apr 30 14:15 rndc.key
-rw-r--r-- 1 root root 1317 Feb 15 08:51 zones.rfc1918
root@debian5-Pruebas:/etc/bind# cat named.conf.default-zones
// prime the server with knowledge of the root servers
zone "." {
    type hint;
    file "/usr/share/dns/root.hints";
};

// be authoritative for the local forward and reverse zones, and for
// broadcast zones as per RFC 1912

zone "localhost" {
    type master;
    file "/etc/bind/db.local";
};

zone "127.0.0.1" {
    type master;
    file "/etc/bind/db.127";
};

zone "0.0.0.0" {
    type master;
    file "/etc/bind/db.0";
};

zone "255.0.0.0" {
    type master;
    file "/etc/bind/db.255";
};

root@debian5-Pruebas:/etc/bind#

```

```

SSINF-Debian-Base-5-PRUEBA [Corriendo]...
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda

root@debian5-Pruebas:/etc/bind# ls -la
total 56
drwxr-xr-x  2 root root 4096 Apr 30 14:15 .
drwxr-xr-x 74 root root 4096 Apr 30 14:15 ..
-rw-r--r-- 1 root root 2761 Feb 15 08:51 bind.keys
-rw-r--r-- 1 root root  237 Feb 15 08:51 db.0
-rw-r--r-- 1 root root  271 Feb 15 08:51 db.127
-rw-r--r-- 1 root root  237 Feb 15 08:51 db.255
-rw-r--r-- 1 root root  353 Feb 15 08:51 db.empty
-rw-r--r-- 1 root root  270 Feb 15 08:51 db.local
-rw-r--r-- 1 root root  463 Feb 15 08:51 named.conf
-rw-r--r-- 1 root bind 498 Feb 15 08:51 named.conf.default-zones
-rw-r--r-- 1 root bind 165 Feb 15 08:51 named.conf.local
-rw-r--r-- 1 root bind 846 Feb 15 08:51 named.conf.options
-rw-r--r-- 1 bind bind 77 Apr 30 14:15 rndc.key
-rw-r--r-- 1 root root 1317 Feb 15 08:51 zones.rfc1918
root@debian5-Pruebas:/etc/bind# cat named.conf.local
// Do any local configuration here
//
// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";

root@debian5-Pruebas:/etc/bind#

```

