

## PRÁCTICA 1. DNS en Ubuntu 22/24



Si imaginamos que trabajamos en una empresa necesitamos implantar un servicio de resolución que permita traducir nombres a direcciones IP. Queremos facilitar el acceso remoto de los trabajadores a los distintos equipos de la red de la empresa. Los trabajadores no tienen estudios informáticos, utilizan los ordenadores a nivel de usuario, por lo que les cuesta mucho memorizar las direcciones y relacionarlas con los servicios que ofrecen los equipos a los que quieren acceder.

Después de estudiar la red, el trabajo de los empleados y la perspectiva de crecimiento, proponéis que, además de utilizar el servicio DNS para navegar por Internet, se utilice también para resolver los nombres de los ordenadores de la red de área local.

## 1. Instalación del servidor DNS bind

Para instalarlo utilizaremos la orden apt-get desde una consola de root:

```
# sudo -s apt-get install bind9
```

Si has configurado la red de Virtualbox como Red privada no tendras acceso a Internet por tanto no podras instalarlo. Por tanto hay que cambiar la configuracion de red. Para ello siguientes pasos:

a) Modifica la red de VirtualBox a Modo Puente

b) Accede a la carpeta `/etc/netplan`

```
cd /etc/netplan
```

c) Haz una copia del fichero 50-cloud-init.yaml como 50-cloud-init\_copia.yaml

```
sudo cp 50-cloud-init.yaml 50-cloud-init_copia.yaml
```

d) Entra en el archivo de configuración anterior 50-cloud-init.yaml (en `/etc/netplan`) , cambia el valor del dhcp a truey añade un # (comenta ) las líneas que se han añadido:

```
GNU nano 6.2                               50-cloud-init.yaml
# This file is generated from information provided by the datasource. Changes
# to it will not persist across an instance reboot. To disable cloud-init's
# network configuration capabilities, write a file
# /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg with the following:
# network: {config: disabled}
network:
  ethernets:
    enp0s3:
      dhcp4: true
      #addresses: [192.168.1.100/24]
      #nameservers:
        # addresses: [8.8.8.8]
version: 2
```

e) Vuelve a guardar el archivo

f) Ejecuta el comando `sudo netplan apply`

g) Comprueba con `ifconfig` si te ha dado una dirección IP diferente a la fija que tenias

h) Si no te la ha dado, ver a la interfaz gráfica y desde la configuración de red, busca la opción de DHCP automático y Aplica cambios.

i) Vuelve a comprobar si te ha cambiado la IP. Si lo ha hecho, entonces ya tienes internet.

j) Descarga el servidor bind

```
# sudo -s apt-get install bind9
```

- k) Prueba que se ha instalado con el comando:

```
# sudo systemctl status bind9
```

En la línea Active, ha de salir en verde la palabra “active (running)”

```
alumnoadmin@ubuntuserver22:/etc$ systemctl status bind9
● named.service - BIND Domain Name Server
   Loaded: loaded (/lib/systemd/system/named.service; enabled; vendor preset:>)
   Active: active (running) since Sun 2024-10-13 18:08:46 UTC; 2min 4s ago
     Docs: man:named(8)
   Process: 18727 ExecStart=/usr/sbin/named $OPTIONS (code=exited, status=0/SU>
 Main PID: 18728 (named)
    Tasks: 5 (limit: 4557)
      Memory: 7.0M
        CPU: 63ms
       CGroup: /system.slice/named.service
               └─18728 /usr/sbin/named -u bind

oct 13 18:08:46 ubuntuserver22 named[18728]: network unreachable resolving './D>
oct 13 18:08:46 ubuntuserver22 named[18728]: network unreachable resolving './N>
oct 13 18:08:46 ubuntuserver22 named[18728]: network unreachable resolving './D>
oct 13 18:08:46 ubuntuserver22 named[18728]: network unreachable resolving './N>
oct 13 18:08:46 ubuntuserver22 named[18728]: network unreachable resolving './D>
oct 13 18:08:46 ubuntuserver22 named[18728]: network unreachable resolving './N>
oct 13 18:08:46 ubuntuserver22 named[18728]: network unreachable resolving './D>
oct 13 18:08:46 ubuntuserver22 named[18728]: network unreachable resolving './N>
oct 13 18:08:46 ubuntuserver22 named[18728]: managed-keys-zone: Initializing au>
oct 13 18:08:46 ubuntuserver22 named[18728]: resolver priming query complete: s>
Lines 1-22/22 (END)

alumnoadmin@ubuntuserver22:/etc$
```

- l) Vuelve a editar el archivo de configuración anterior (50-cloud-init.yaml) y vuelve a poner el valor de dhcp a false y quita los comentarios de las líneas:

```
GNU nano 6.2          50-cloud-init.yaml *
# This file is generated from information provided by the datasource. Changes
# to it will not persist across an instance reboot. To disable cloud-init's
# network configuration capabilities, write a file
# /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg with the following:
# network: {config: disabled}
network:
  ethernets:
    enp0s3:
      dhcp4: false
      addresses: [192.168.1.100/24]
      nameservers:
        addresses: [8.8.8.8]
  version: 2
```

- m) Ejecuta de nuevo: sudo netplan apply  
n) Y comprueba de nuevo el estado de bind (Comprueba que tienes el servicio activo) con el comando:

```
systemctl status bind9
```

En la línea Active, ha de salir en verde la palabra “active (running)”

```
alumnoadmin@ubuntuserver22:/etc$ systemctl status bind9
● named.service - BIND Domain Name Server
  Loaded: loaded (/lib/systemd/system/named.service; enabled; vendor preset:>)
  Active: active (running) since Sun 2024-10-13 18:08:46 UTC; 2min 4s ago
    Docs: man:named(8)
 Process: 18727 ExecStart=/usr/sbin/named $OPTIONS (code=exited, status=0/SU>
 Main PID: 18728 (named)
   Tasks: 5 (limit: 4557)
  Memory: 7.0M
    CPU: 63ms
   CGroup: /system.slice/named.service
           └─18728 /usr/sbin/named -u bind

oct 13 18:08:46 ubuntuserver22 named[18728]: network unreachable resolving './D>
oct 13 18:08:46 ubuntuserver22 named[18728]: network unreachable resolving './N>
oct 13 18:08:46 ubuntuserver22 named[18728]: network unreachable resolving './D>
oct 13 18:08:46 ubuntuserver22 named[18728]: network unreachable resolving './N>
oct 13 18:08:46 ubuntuserver22 named[18728]: network unreachable resolving './D>
oct 13 18:08:46 ubuntuserver22 named[18728]: network unreachable resolving './N>
oct 13 18:08:46 ubuntuserver22 named[18728]: managed-keys-zone: Initializing au>
oct 13 18:08:46 ubuntuserver22 named[18728]: resolver priming query complete: s>
Lines 1-22/22 (END)

alumnoadmin@ubuntuserver22:/etc$
```

Para salir del comando presiona Ctrl+C y Enter

Otra forma de saber si se ha instalado Bind, es comprobando si está la carpeta /bind en /etc. Accede, por tanto, a la ruta en la que se habrá instalado bind9 y comprueba que hay una serie de archivos.

```
# cd /etc/bind
```

**Antes de empezar**, vamos a establecer algunos conceptos para la práctica:

- El dominio que vais a creáros se va a llamar "*misiniciales.com*" (se refiere a tus iniciales, si te llamas Pere López Garrigues, sería "plg.com")
- El primer host que vais a crear se llama "[www.misiniciales.com](http://www.misiniciales.com)" y va a redireccionar al propio servidor.
- El archivo de configuración de vuestro servidor DNS se va a llamar "db.*misiniciales*" que es el que va a contener las referencias a nombres de dominios y nombres de máquina.
- La IP de la máquina [www.misiniciales.com](http://www.misiniciales.com) que tiene como dominio misiniciales.com es la IP de vuestra máquina servidor.

### Ejercicio 1

Crea un documento de memoria de la práctica llamado: P1\_DNS\_tunombre. Para este ejercicio, muestra una captura de pantalla mostrando que tienes bind9 instalado en tu servidor Ubuntu.

## 2. Configuración del servidor maestro DNS bind

### 2.1. Creación de una zona de búsqueda directa.

**La zona de búsqueda directa** resuelve los nombres de host en direcciones IP y albergan los registros de recursos comunes: A, CNAME, SRV, MX, SOA, TXT y NS.

#### 1. Editamos el archivo “db.local”

Teniendo como referencia el archivo db.local que se crea al instalar el servicio bind, vamos a crear nuestro propio archivo de configuración de dns. Dentro de /etc/bind. Para ello, haz una copia del db.local a db.misiniciales.com. En mi caso será db.geg.com

```
# cp db.local db.misiniciales.com
```

Editamos el archivo que acabamos de crear: 'db.misiniciales.com' y sustituimos todas las palabras "localhost" por el nombre del dominio que vamos a crear. Cambia la dirección IP por tu dirección del servidor.

También se debería cambiar la cláusula *Serial* cada vez que editemos nuestros ficheros de zona, esto es una forma de llevar un control de versiones. Por ejemplo, en este caso podríamos añadir una numeración del tipo YYMMDDVV (año mes día versión).

En este ejemplo quedará de este modo:

```
GNU nano 4.8                                     db.geg.com
;
; BIND data file for local loopback interface
;

$TTL    604800
@      IN      SOA   geg.com. root.geg.com. (
                      2           ; Serial
                      604800     ; Refresh
                      86400      ; Retry
                     2419200    ; Expire
                      604800 )   ; Negative Cache TTL
;
@      IN      NS    geg.com.
@      IN      A     192.168.1.100
@      IN      AAAA  ::1
dns   IN      A     192.168.1.100
```

**NOTA:** Ten mucho cuidado con los puntos detrás de cada nombre de domino. Aunque parezca extraño, el dominio acaba con un punto en la configuración.

- **NS** nos identifica el NAME SERVER o nombre del servidor.
- **@ IN A 192.168.1.100** dice que el dominio *misiniciales.com* llevará la dirección IP 192.168.1.100

**Ejercicio 2**

Muestra una captura de pantalla de tu fichero db.misiniciales.com.

## **2. Editamos el archivo “named.conf.local” para crear la ZONA DIRECTA**

En /etc/bind se encuentra el archivo “named.conf.local”. En él definiremos las zonas, directa e inversa, para el dominio *misiniciales.com* haciendo uso de la cláusula *zone*. Cada bloque de zona incluirá de que tipo será la zona y el fichero en el que estarán definidas, para cada zona, las características, propiedades y entidades del dominio.

```
GNU nano 6.2                               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";

//zona directa geg.com
zone "geg.com"{
    type master;
    file "/etc/bind/db.geg.com";
};■
```

Para crear la zona directa de nuestro dominio (*misiniciales.com*) hay que editar el archivo “**/etc/bind/named.conf.local**”. Es importante poner el nombre del fichero indicando el nombre de la ruta completa.

**Ejercicio 3**

Muestra una captura de pantalla de tu zona directa creada en el fichero “*named.conf.local*”.

## **3. Reiniciamos bind**

Una vez que hayamos configurado bind, lo reiniciamos:

```
sudo service bind9 restart
O bien
sudo systemctl restart bind9
```

Podemos comprobar que sintácticamente nuestros ficheros de configuración son correctos ejecutando las siguientes órdenes:

- `named-checkconf` : analiza el fichero `named.conf.local`

En nuestro ejemplo:

```
named-checkconf named.conf.local
o bien
sudo named-checkconf /etc/bind/named.conf.local
```

Si no contesta nada, es que todo está correcto.

- `named-checkzone nomDominio nomFichero`: analiza el fichero de zona en cuestión.

En nuestro ejemplo:

```
named-checkzone 192.168.1.100 /etc/bind/db.geg.com
o bien
sudo named-checkzone 192.168.1.100 /etc/bind/db.geg.com
```

Si en este caso contesta con :

```
alumnoadmin@ubuntuserver22:/etc/bind$ named-checkzone 192.168.1.100 /etc/bind/db.geg.com
zone 192.168.1.100/IN: loaded serial 2
OK
```

Es que todo ha ido bien.

#### Ejercicio 4

Muestra una captura de pantalla de los resultados de tus comandos.

## 4. Configuración fichero `named.conf.options`

Este fichero define la caché del DNS y la configuración genérica del servidor, como transferencia de zonas, forwarders (reenviadores), etc.

Con la configuración por defecto el servidor DNS Bind ya se podría resolver nombres de Internet. Pero si quisieramos que esta resolución se hiciera a través de otros servidores a los que conecte el nuestro (lo que se denomina **forwarding**), habría que editar el archivo `named.conf.options`:

Por tanto, **el siguiente paso sería** editar este fichero `/etc/bind/named.conf.options` que permite crear una lista de acceso para restringir quien puede realizar las consultas a nuestro servidor DNS. También permite indicar servidores *forwarders* sobre los que delegará nuestro servidor DNS local cuando no pueda resolver alguna consulta.

En nuestro caso vamos a configurar tan solo los forwarders (reenviadores), ya que son necesarios para que cuando un cliente realice una consulta DNS y no encuentre la respuesta en local, la respuesta se pueda encontrar en Internet, como puede ser el

nombre de un servidor, un sitio web, un servidor de correo, etc. Por lo general son los servidores DNS de nuestro ISP por ejemplo Telefónica, etc. Aunque se pueden poner también, por ejemplo, los DNS de Google: 8.8.8.8

```
GNU nano 6.2                               named.conf.options
options {
    directory "/var/cache/bind";

    // If there is a firewall between you and nameservers you want
    // to talk to, you may need to fix the firewall to allow multiple
    // ports to talk. See http://www.kb.cert.org/vuls/id/800113

    // If your ISP provided one or more IP addresses for stable
    // nameservers, you probably want to use them as forwarders.
    // Uncomment the following block, and insert the addresses replacing
    // the all-0's placeholder.

    forwarders {
        8.8.8.8;
    };

    =====
    // If BIND logs error messages about the root key being expired,
    // you will need to update your keys. See https://www.isc.org/bound-keys
    =====
}
```

Después de esto ya tendremos configurado el servidor DNS. Solo queda reiniciar el servicio bind9 y comprobar que esté ejecutándose correctamente.

```
# sudo systemctl restart bind9
O bien
#service bind9 restart
```

Reinicia el servicio BIND9

```
# sudo service bind9 status
```

```
gema@ubuntuserver:/etc$ sudo service bind9 status
sudo: unable to resolve host ubuntuserver: Temporary failure in name resolution
● named.service - BIND Domain Name Server
  Loaded: loaded (/lib/systemd/system/named.service; enabled; vendor preset:enabled)
  Active: active (running) since Mon 2022-10-03 12:49:34 UTC; 5h 15min ago
    Docs: man:named(8)
   Main PID: 30013 (named)
     Tasks: 5 (limit: 9437)
    Memory: 15.0M
   CGroup: /system.slice/named.service
           └─30013 /usr/sbin/named -f -u bind

oct 03 12:49:34 ubuntuserver named[30013]: network unreachable resolving './NS/>
oct 03 12:49:34 ubuntuserver named[30013]: network unreachable resolving './NS/>
oct 03 12:49:34 ubuntuserver named[30013]: network unreachable resolving './NS/>
```

### Ejercicio 4B

Muestra una captura de pantalla del estado del servicio bind9.

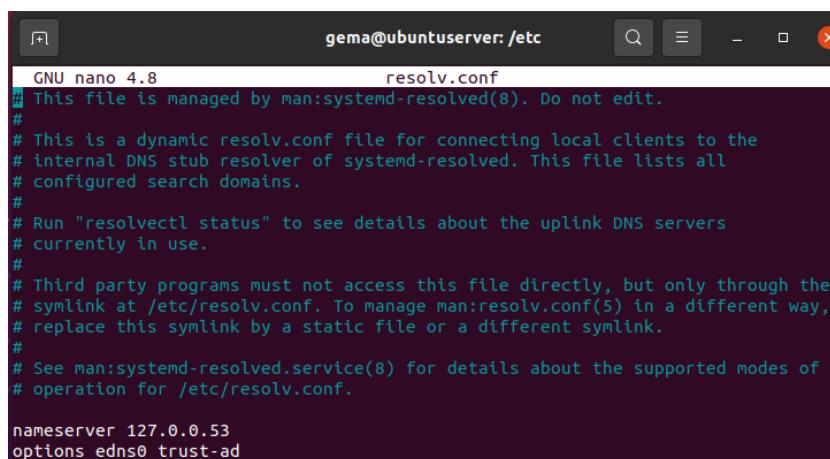
## 5. Comprobación de la resolución directa

Para probar el servidor DNS debemos indicar que la resolución de nombres de nuestro equipo va a realizarla nuestro propio servidor.

En Ubuntu Server las IPs de los servidores DNS se encuentran en el fichero **/etc/resolv.conf**. En principio únicamente deberíamos modificar su contenido de manera que aparezca nuestra propia IP de la interfaz de red.

Al configurar **resolv.conf** nos permitirá indicar cuales son los servidores DNS, el dominio de nuestro servidor y dónde es necesario realizar la búsqueda.

Pero... Si consultamos el fichero **/etc/resolv.conf**, donde debería aparecer como **nameserver** la *IP de nuestro servidor* (localhost), veremos que nos aparece la dirección de **loopback 127.0.0.53**. Además, si le hacemos un **ls -l**, podemos observar que se trata de un enlace simbólico a otro fichero de configuración.



```
GNU nano 4.8             resolv.conf
# This file is managed by man:systemd-resolved(8). Do not edit.
#
# This is a dynamic resolv.conf file for connecting local clients to the
# internal DNS stub resolver of systemd-resolved. This file lists all
# configured search domains.
#
# Run "resolvectl status" to see details about the uplink DNS servers
# currently in use.
#
# Third party programs must not access this file directly, but only through the
# symlink at /etc/resolv.conf. To manage man:resolv.conf(5) in a different way,
# replace this symlink by a static file or a different symlink.
#
# See man:systemd-resolved.service(8) for details about the supported modes of
# operation for /etc/resolv.conf.

nameserver 127.0.0.53
options edns0 trust-ad
```

Por defecto, las aplicaciones hacen uso del resolver de **systemd-resolved** que escucha en la interfaz de loopback en la dirección 127.0.0.53.

Entonces, **¿Qué está ocurriendo?** Bien, resulta que **Systemd** tiene su propia implementación de resolución de nombres: **systemd-resolved**. Además de implementar un resolver, añade características adicionales tales como caché DNS y validación DNSSEC.

Entonces..., **¿Cómo podemos desactivar systemd-resolved y que las aplicaciones consulten nuestro resolver directamente?** Existen 4 modos de implementación de systemd-resolved, no vamos a ver cómo funciona cada uno, tan solo nos centraremos en una.

Antes hemos comentado que **/etc/resolv.conf** era un enlace simbólico a otro fichero de configuración. Pues bien, existen 4 ficheros de configuración, 1 por cada modo de implementación de systemd-resolved.

## En UBUNTU 20

- Para que todo funcione como antes de systemd y las aplicaciones consulten nuestro resolver directamente, debemos cambiar el enlace simbólico de `/etc/resolv.conf` y apuntarlo al siguiente fichero `/run/systemd/resolve/resolv.conf` con el comando que se muestra en la siguiente imagen:

```
gema@ubuntuserver:/etc$ sudo ln -sf /run/systemd/resolve/resolv.conf /etc/resolv.conf
gema@ubuntuserver:/etc$ ls -l /etc/resolv.conf
lrwxrwxrwx 1 root root 32 oct 3 17:03 /etc/resolv.conf -> /run/systemd/resolve/resolv.conf
gema@ubuntuserver:/etc$
```

- Una vez modificado el enlace simbólico pasemos a comprobar el nuevo contenido de `/etc/resolv.conf` y veremos cómo, ahora sí, aparece como **nameserver** la IP de nuestro servidor.

```
# cat /etc/resolv.conf
```

- bien puedes también comprobarlo con:

```
# nano /etc/resolv.conf
```

Sino aparece la información que se muestra en la imagen de abajo

```
GNU nano 4.8          resolv.conf
# This file is managed by man:systemd-resolved(8). Do not edit.
#
# This is a dynamic resolv.conf file for connecting local clients to the
# internal DNS stub resolver of systemd-resolved. This file lists all
# configured search domains.
#
# Run "resolvectl status" to see details about the uplink DNS servers
# currently in use.
#
# Third party programs must not access this file directly, but only through the
# symlink at /etc/resolv.conf. To manage man:resolv.conf(5) in a different way,
# replace this symlink by a static file or a different symlink.
#
# See man:systemd-resolved.service(8) for details about the supported modes of
# operation for /etc/resolv.conf.

#nameserver 127.0.0.53
domain geg.com
nameserver 192.168.1.100
options edns0 trust-ad
```

Entonces entra en el archivo `/run/systemd/resolve/resolv.conf`

Y modifícalo de manera que se le parezca a la imagen anterior.

## En UBUNTU 22/24

En este caso sigue los siguientes pasos:

- Para que todo funcione como antes de systemd y las aplicaciones consulten

nuestro resolver directamente, debemos cambiar el enlace simbólico de `/etc/resolv.conf` y apuntarlo al siguiente fichero `/run/systemd/resolve/resolv.conf` con el comando que se muestra en la siguiente imagen:

```
gema@ubuntuserver:/etc$ sudo ln -sf /run/systemd/resolve/resolv.conf /etc/resolv.conf
gema@ubuntuserver:/etc$ ls -l /etc/resolv.conf
lrwxrwxrwx 1 root root 32 oct  3 17:03 /etc/resolv.conf -> /run/systemd/resolve/resolv.conf
gema@ubuntuserver:/etc$
```

Ahora edita el archivo `/etc/systemd/resolved.conf`

Y modifícalo de manera que quede como en la imagen siguiente:

- Descomenta la línea que pone DNS
- y añade **DNS=192.168.1.100**

```
root@servidorc: /home/admini
GNU nano 6.2
/etc/systemd/resolved.conf
# This file is part of systemd.
#
# systemd is free software; you can redistribute it and/or modify it under the
# terms of the GNU Lesser General Public License as published by the Free
# Software Foundation; either version 2.1 of the License, or (at your option)
# any later version.
#
# Entries in this file show the compile time defaults. Local configuration
# should be created by either modifying this file, or by creating "drop-ins" in
# the resolved.conf.d/ subdirectory. The latter is generally recommended.
# Defaults can be restored by simply deleting this file and all drop-ins.
#
# Use 'systemd-analyze cat-config systemd/resolved.conf' to display the full config.
#
# See resolved.conf(5) for details.

[Resolve]
# Some examples of DNS servers which may be used for DNS= and FallbackDNS=:
# Cloudflare: 1.1.1.1#cloudflare-dns.com 1.0.0.1#cloudflare-dns.com 2606:4700:4700::1111#cloudflare-dns.com
# Google: 8.8.8#dns.google 8.8.4.4#dns.google 2001:4860:4860::8888#dns.google 2001:4860:4860::8888#dns.google
# Quad9: 9.9.9.9#dns.quad9.net 149.112.112.112#dns.quad9.net 2620:fe::fe#dns.quad9.net 2620:fe::fe#dns.quad9.net
DNS=192.168.1.100
#FallbackDNS=
#Domains=
```

Una vez modificado hay que reiniciar el servicio con el comando:  
**systemctl restart systemd-resolved**

### Ejercicio 5

Muestra una captura de pantalla de tu archivo.

- Cambia ahora la configuración de la zona (`db.misiniciales.com`) de manera que **www** sea un alias para la máquina, y no un registro **A**.

Observa la siguiente imagen

```
GNU nano 4.8                               db.geg.com
;
; BIND data file for local loopback interface
;
$TTL    604800
@      IN      SOA     geg.com. root.geg.com. (
                      2           ; Serial
                      604800      ; Refresh
                      86400       ; Retry
                     2419200     ; Expire
                     604800 )    ; Negative Cache TTL
;
@      IN      NS      geg.com.
@      IN      A       192.168.1.100
@      IN      AAAA    ::1
dns   IN      A       192.168.1.100
ubuntuserver IN      A       192.168.1.100
www   IN      CNAME   ubuntuserver
```

Vuelve a comprobar que no hay errores en la configuración del fichero de zona, con el **named-checkzone** visto anteriormente.

- **Ejercicio 6**  
Muestra una captura de pantalla de tu alias.

## 2.2. Creación de una zona de búsqueda inversa

La zona de búsqueda inversa resuelve una dirección IP a un nombre de dominio, y albergan los registros: SOA, NS y PTR.

Una vez configurado el **dominio de búsqueda** directo, hay que configurar el **inverso**.

Para ello, debemos añadir las siguientes líneas al archivo

"/etc/bind/named.conf.local":

```
GNU nano 4.8                               named.conf.local
//                                         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";

//zona directa geg.com
zone "geg.com"{
    type master;
    file "/etc/bind/db.geg.com";
};

//zona busqueda inversa
zone "168.192.in-addr.arpa" {
    type master;
    file "/etc/bind/db.168.192";
};■
```

donde 168.192 hace referencia al principio de nuestra dirección ip y "/etc/bind/db.168.192" al archivo de configuración de resolución inversa.

Es importante que pongamos el nombre del fichero correctamente porque esta versión de bind9 busca directamente en el dominio 168.192.in-addr.arpa a la hora de hacer la resolución inversa, y si ponemos cualquier otro nombre de zona o de fichero nos dará error.

- Ejercicio 7**

Muestra una captura de pantalla de tu archivo "named.conf.local" donde se vea la zona inversa que has creado.

Como hemos hecho referencia al archivo, obviamente habrá que crearlo. Como en el caso de la zona directa, vamos a partir de la copia de un fichero que viene creado por defecto en bind9.

```
# cp db.127 db.168.192
```

Una vez creado, lo modificamos, sustituyendo todas las palabras "localhost" por nuestro dominio.

Tengamos en cuenta que en la parte más baja, habrá que escribir la dirección IP de nuestra máquina, pero al revés y omitiendo "192.168". Por ejemplo, si la IP de nuestra máquina es 192.168.1.45 habrá que escribir: 45.1

En este ejemplo quedaría de este modo:

```
GNU nano 4.8                               db.168.192
;
; BIND reverse data file for local loopback interface
;
$TTL    604800
@      IN      SOA     geg.com. root.geg.com. (
                      1           ; Serial
                      604800      ; Refresh
                      86400       ; Retry
                     2419200     ; Expire
                      604800 )     ; Negative Cache TTL
;
@      IN      NS      geg.com.
100.1  IN      PTR     geg.com.
```

Es importante resaltar que **deberá existir un registro PTR por cada fichero A que tengamos en el fichero de zona directa.**

Para comprobar que el archivo no tiene errores tienes que ejecutar los comandos mencionados en el punto de búsqueda directa: named-checkconf y named-checkzone.

**ii CUIDADO** cuando se haga checkzone de tu fichero db.168.192. El nombre de dominio es 168.192.in-addr.arpa!!

- **Ejercicio 8**  
Muestra una captura de pantalla de tu archivo db.168.192
- **Ejercicio 9**  
Muestra una captura de pantalla de tu resultado de los comandos named-.

Como siempre, comprobamos que funciona correctamente después de reiniciar el servicio.

```
# sudo service bind9 restart
```

Ahora para probar que funciona nuestro dominio tenemos que probar la resolución directa y la inversa de este modo:

Resolución directa:

(El comando Host es una utilidad fácil de usar para realizar búsquedas DNS que traducen nombres de dominio a direcciones IP y viceversa.)

Prueba los siguientes comandos pero para tu configuración.

```
# host www.geg.com
```

También podemos probar:

```
# host geg.com
```

Resolución inversa:

```
# host 192.168.1.100
```

Si todo ha ido bien ha de salirte algo similar a lo que aparece en la siguiente imagen.

```
gema@ubuntuserver:/etc/bind$ sudo host www.geg.com
www.geg.com is an alias for ubuntuserver.geg.com.
ubuntuserver.geg.com has address 192.168.1.100
gema@ubuntuserver:/etc/bind$ sudo host geg.com
geg.com has address 192.168.1.100
geg.com has IPv6 address ::1
gema@ubuntuserver:/etc/bind$
```

En este momento contamos con un servidor DNS muy sencillo en el que únicamente nos encontramos con la resolución de nombres del equipo servidor.

- **Ejercicio 10**

Muestra una captura de pantalla donde se vea que funciona tu DNS con resolución directa y resolución inversa.

Otros comandos posibles para comprobar el funcionamiento del DNS.

- **Ejercicio 11**

Busca que hace o para qué sirve el comando dig y muestra un ejemplo de su uso, con la captura de pantalla de su resultado.

El resultado ha de ser similar a la de la siguiente imagen:

```
gema@ubuntuserver:/etc/bind$ dig geg.com
; <>> DiG 9.16.1-Ubuntu <>> geg.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 49462
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
;;
;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 4096
;; COOKIE: faad572c6cd4e71701000000633c79b4edf8e6d12d529641 (good)
;; QUESTION SECTION:
;geg.com.           IN      A
;;
;; ANSWER SECTION:
geg.com.        604800  IN      A       192.168.1.100
;;
;; Query time: 4 msec
;; SERVER: 192.168.1.100#53(192.168.1.100)
;; WHEN: mar oct 04 18:21:40 UTC 2022
;; MSG SIZE rcvd: 80
```

- **Ejercicio 12**

Utiliza el comando nslookup para comprobar la funcionalidad del DNS. Muestra una captura de pantalla de su uso.

```
gema@ubuntuserver:/etc/bind$ nslookup google.com
Server:     8.8.8.8
Address:   8.8.8.8#53

Non-authoritative answer:
Name:   google.com
Address: 142.250.200.110
Name:   google.com
Address: 2a00:1450:4003:808::200e

gema@ubuntuserver:/etc/bind$ nslookup geg.com
Server:     192.168.1.100
Address:   192.168.1.100#53

Name:   geg.com
Address: 192.168.1.100
Name:   geg.com
Address: ::1
```