



DNS

El servicio de resolución de nombre o "domain name system" es el responsable de traducir nombres de dominio como www.google.es a direcciones IP y viceversa.

En redes, los equipos se comunican entre ellos usando exclusivamente sus direcciones IP. En telefonía móvil pasa lo mismo, para llamar a un contacto debemos usar su número de teléfono. Para facilitar la comunicación al usuario, los teléfonos móviles incluyen una aplicación de agenda que es capaz de traducir un contacto a un número de teléfono cuando queremos emitir una llamada. Esta misma aplicación también es capaz de traducir un número de teléfono a un contacto de la agenda cuando recibimos una llamada.

El DNS funciona de la misma manera, es decir mantiene una agenda que relaciona IP y nombre.

Todos los sistemas operativos incluyen un archivo que se suele llamar "hosts" en el que se pueden guardar parejas de IP y nombre de manera local a ese equipo. Este archivo será el primero en ser consultado y tendrá prioridad sobre todos. Actúa como una caché local.

```

administrador@equipo: ~
GNU nano 6.2                               /etc/hosts *
127.0.0.1      localhost
127.0.1.1      equipo
127.0.2.1      www.google.es

# The following lines are desirable for IPv6 capable hosts
::1          ip6-localhost ip6-loopback
fe00::0       ip6-localnet
ff00::0       ip6-mcastprefix
ff02::1       ip6-allnodes
ff02::2       ip6-allrouters

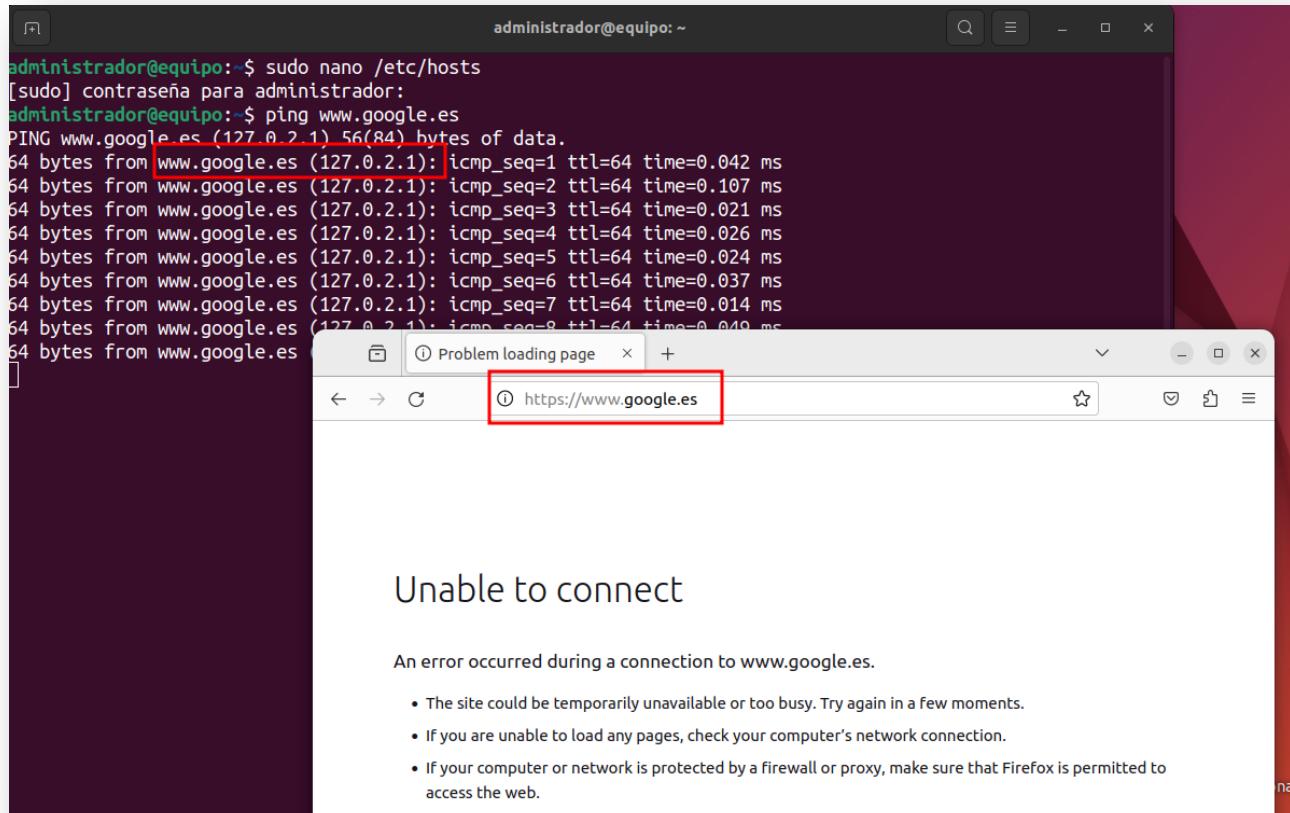
```

En la captura anterior edito el archivo "/etc/hosts" para añadir una nueva línea en la que indico que la IP 127.0.2.1 (una IP de loopback) está asociada al nombre www.google.es.

En cuanto se guarde el archivo, inmediatamente sucederá que no podré acceder a la web www.google.es porque estoy apuntándome a mí mismo y no tengo ningún servidor web en este nodo. Además sucederá



que podré hacer ping a esa dirección y recibiré respuesta pero es porque me estoy haciendo ping a mí mismo y no al www.google.es que todos pensamos.



```
administrador@equipo:~$ sudo nano /etc/hosts
[sudo] contraseña para administrador:
administrador@equipo:~$ ping www.google.es
PING www.google.es (127.0.2.1) 56(84) bytes of data.
64 bytes from www.google.es (127.0.2.1): icmp_seq=1 ttl=64 time=0.042 ms
64 bytes from www.google.es (127.0.2.1): icmp_seq=2 ttl=64 time=0.107 ms
64 bytes from www.google.es (127.0.2.1): icmp_seq=3 ttl=64 time=0.021 ms
64 bytes from www.google.es (127.0.2.1): icmp_seq=4 ttl=64 time=0.026 ms
64 bytes from www.google.es (127.0.2.1): icmp_seq=5 ttl=64 time=0.024 ms
64 bytes from www.google.es (127.0.2.1): icmp_seq=6 ttl=64 time=0.037 ms
64 bytes from www.google.es (127.0.2.1): icmp_seq=7 ttl=64 time=0.014 ms
64 bytes from www.google.es (127.0.2.1): icmp_seq=8 ttl=64 time=0.049 ms
64 bytes from www.google.es
[...]
```

The terminal window title is "administrador@equipo: ~". The browser window title is "Problem loading page" and the address bar shows "https://www.google.es". The browser content says "Unable to connect" and "An error occurred during a connection to www.google.es." with a list of troubleshooting steps.

En caso de que la IP o el nombre no se encuentren en este archivo "hosts" el sistema operativo comprobará si tiene alguna dirección IP de servidores DNS para poder preguntarles a ellos y poder resolver.

Como ya hemos visto anteriormente en el curso, la información de cuales son los servidores DNS de consulta es algo opcional. En la configuración de IP manual que se hace tanto en Windows como en Linux, pasar esta información requiere de indicarla explícitamente.

Tipos de servidores DNS

Mantener la relación entre todas las IPs y nombres no es sencillo. No solo por la cantidad de registros sino también por la cantidad inmensa de consultas que reciben estos servidores y otros factores.

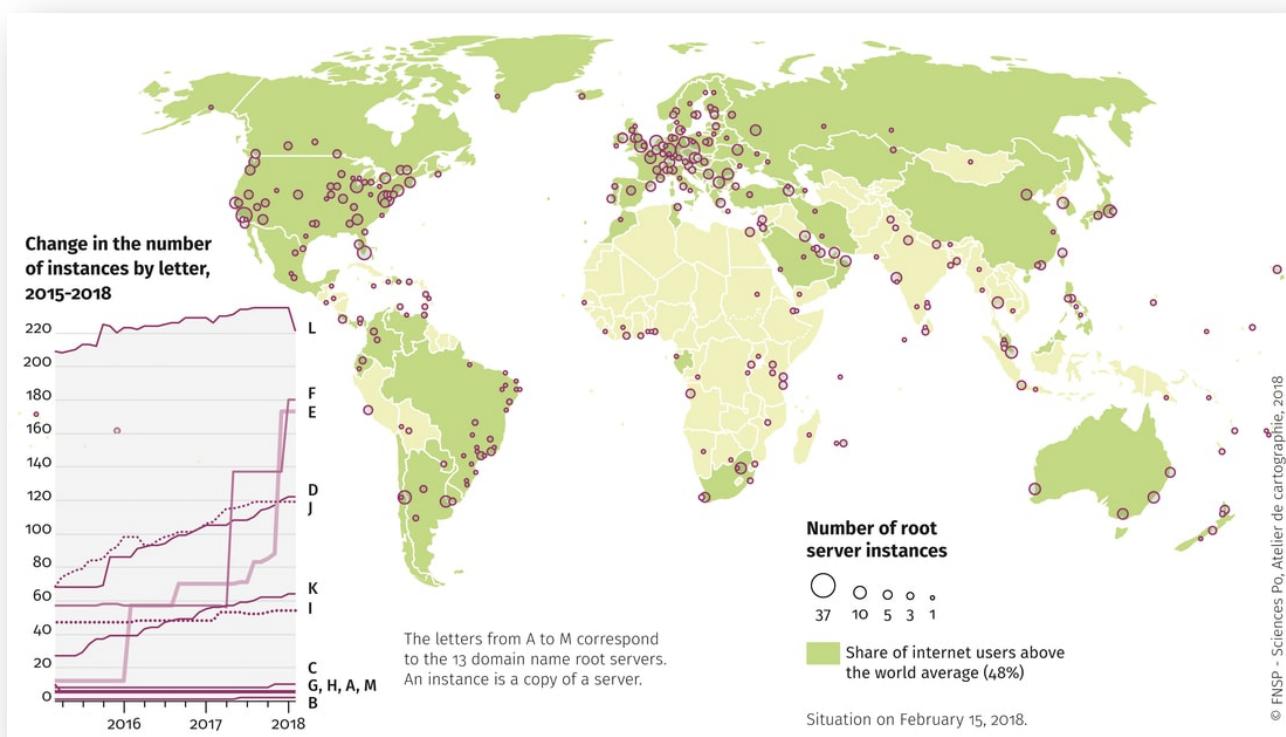
La solución elegida ha sido usar una estructura de árbol donde los servidores DNS solo dominan un subconjunto de relaciones.



Los **servidores recursivos**, también llamados “resolvers” son los de más bajo nivel, los más cercanos a las hojas del árbol) y atienden las peticiones directas de los clientes (representados por las hojas del árbol). Es probable que en su agenda solo tengan cacheadas las entradas más habituales.

Un servidor recursivo de DNS puede ser un nodo de tu red, puede que tu propio router incluya un servidor DNS, es probable que uses los DNS de su proveedor de servicio o tal vez te sepas alguno externo como los que ofrece Google (8.8.8.8 y 8.8.4.4) o CloudFlare (1.1.1.1).

El siguiente nivel lo constituyen los **servidores raíz**, también llamados “root servers”. Estos servidores se encargan de redirigir la pregunta al servidor que tenga información más específica. Tradicionalmente se usa la parte derecha del nombre para apuntar al servidor DNS adecuado. Puedes obtener más información de estos servidores raíz de DNS en <https://root-servers.org>.



En un nombre o una URL como “edu.xunta.gal” debemos distinguir tres partes separadas por puntos (.) Empezando por la derecha tenemos el TLD o Top-Level Domain. Hasta hace no mucho tiempo esta parte era una referencia geográfica. En la actualidad podemos encontrar aquí todo tipo de palabras.

A continuación, en segundo lugar desde la derecha tenemos lo que se conoce como el **“dominio”**. Digamos que es la puerta de entrada para la parte privada de cada empresa u organización.

Estos dos elementos son obligatorios pero, opcionalmente podemos añadir tantos como queramos a la izquierda de ellos, a estos se los conoce como **“subdominios”**.



En nuestro ejemplo, el nombre "edu.xunta.gal" está compuesto por un TLD que es ".gal", por un dominio ".xunta" y por un subdominio "edu". Cada uno de estos niveles es responsabilidad de un servidor DNS

Los servidores raíz tiene guardadas la IPs de los servidores DNS que resuelven IPs para los TLD como los acabados en ".es", ".com", etc.

Esos servidores que almacenan la información de alguno de los subdominios se llaman **servidores de dominio de nivel superior** o también "TLD". De esta manera, uno o varios servidores TLD conocen todos los subdominios de, por ejemplo, los dominios que acaban en ".gal".

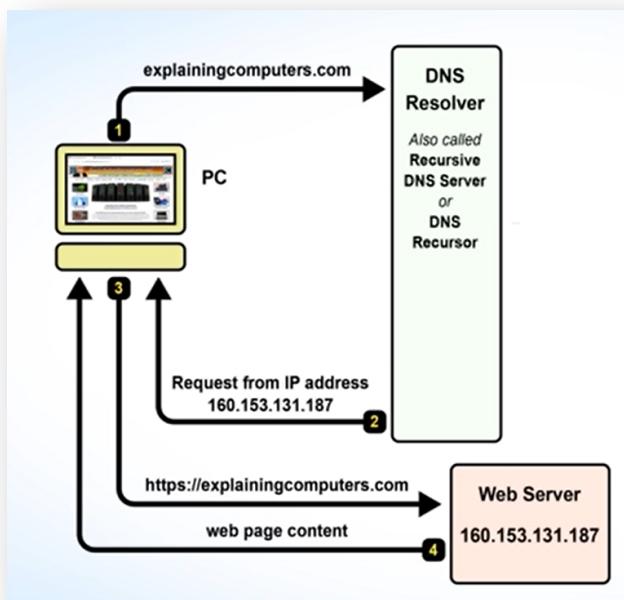
Por último tenemos a los **servidores autoritativos** que son aquellos que tienen la información final y más actual de la relación entre una IP y su nombre. Normalmente estos servidores pertenecen a la propia organización o bien a ISPs que ofrecen el servicio de compra de dominios.

Funcionamiento

Cuando un equipo tiene que resolver un nombre de dominio en una IP, por ejemplo en una petición web de un navegador, realizará las siguientes acciones:

En primer lugar consultará su archivo "hosts" local para ver si tiene alguna entrada que coincida y pueda resolver.

En caso de no poder resolver en local hará una consulta al servidor DNS que tiene configurado. Estos servidores llamados recursivos o "resolvers" mantienen cacheadas las últimas consultas mientras dure su TTL. Así, si un cliente hace una consulta y el servidor DNS tiene una respuesta cacheada aún activa, le contestará con la resolución. Posteriormente el cliente usará la IP para, en este caso, dirigir su solicitud web a un servidor por su IP.



Antes o después la consulta en el “resolver” caducará y desaparecerá o tal vez nunca haya existido porque nunca se ha recibido esa solicitud en concreto. En cualquier caso, cuando un resolver no conoce la respuesta debe encargarse de conseguir la respuesta.

Una posibilidad habitual es consultar a otro resolver de nivel superior como por ejemplo el 8.8.8.8 o los DNS ofrecidos por los ISPs. En este caso se dice que el DNS está configurado con “**forwarders**”. Esta configuración mejora el rendimiento, permite filtrar contenidos y reduce el tráfico externo.

La otra posibilidad es preguntar directamente a un servidor raíz. Estos 13 servidores repartidos por el mundo no resuelven directamente pero sí informan de quien es el responsable de mantener ese dominio de nivel superior (TLD). De esta manera, si nuestra petición es sobre un dominio acabado en “.es”, el servidor raíz nos devolverá los servidores DNS (nameservers) que gestionan todos los subdominios de “.es”.

La información que manejan los servidores raíz se puede consultar en
<https://www.iana.org/domains/root/db>



iana
Internet Assigned Numbers Authority

[Domains](#) [Protocols](#) [Numbers](#) [About](#)

Root Zone Database

The Root Zone Database represents the delegation details of top-level domains, including gTLDs such as .com, and country-code TLDs such as .uk. As the manager of the DNS root zone, we are responsible for coordinating these delegations in accordance with our [policies](#) and [procedures](#).

Much of this data is also available via the WHOIS protocol at whois.iana.org.

DOMAIN	TYPE	TLD MANAGER
.aaa	generic	American Automobile Association, Inc.
.aarp	generic	AARP
.abarth	generic	Not assigned
.abb	generic	ABB Ltd
.abbott	generic	Abbott Laboratories, Inc.
.abbie	generic	Abbie Inc.
.abc	generic	Disney Enterprises, Inc.
.able	generic	Able Inc.
.abogado	generic	Registry Services, LLC
.abuhhabi	generic	Abu Dhabi Systems and Information Centre
.ac	country-code	Internet Computer Bureau Limited
.academy	generic	Binky Moon, LLC
.accenture	generic	Accenture plc

Para cada uno de los dominios de alto nivel se indica qué organización es la encargada de administrar ese dominio. A estas organizaciones se las denomina **NIC (Network Information Center)** y son las responsables de mantener la base de datos con los nombres de dominio registrados bajo ese dominio de alto nivel que administran. Por lo tanto debe mantener los DNS autoritativos de su TLD. Además, y por delegación de la IANA pueden definir políticas de registro y delegar la gestión de dominios a registradores acreditados.

En el caso del dominio ".es", el NIC es RedIRIS y los DNS que aporta son los que aparecen en la siguiente captura.

iana
Internet Assigned Numbers Authority

[Domains](#) [Protocols](#) [Numbers](#) [About](#)

Delegation Record for .ES

(Country code top-level domain)

ccTLD Manager

Red.es
 Edificio Bronze
 Plaza Manuel Gómez Moreno
 Madrid 28020
 Spain

Administrative Contact

Alberto Martínez Lacambra
 Red.es
 Edificio Bronze
 Plaza Manuel Gómez Moreno
 Madrid 28020
 Spain
 Email: esnic-admin@red.es
 Voice: +34 91 212 76 24
 Fax: +34 91 555 78 64

Technical Contact

JUAN VICENTE RODRIGUEZ PÉREZ
 Red.es
 Edificio Bronze
 Plaza Manuel Gómez Moreno s/n
 Madrid 28020
 Spain
 Email: esnic-technic@red.es
 Voice: +34 91 212 76 20
 Fax: +34 91 556 88 64

Name Servers

HOST NAME	IP ADDRESSES
a.nic.es	194.69.254.1 2001:678:40::0:0:64:41
c.nic.es	194.3.4.51 2001:678:44::0:0:0:53
g.nic.es	204.61.237.1 2001:500:14:7001:ad:0:0:1
h.nic.es	194.33.53 2001:678:40::0:0:0:53

Registry Information

URL for registration services: <http://www.nic.es/>
 WHOIS Server: whois.nic.es

IANA Reports

- IANA Report on the Redelegation of the .ES Top-Level Domain (2005-08-05)

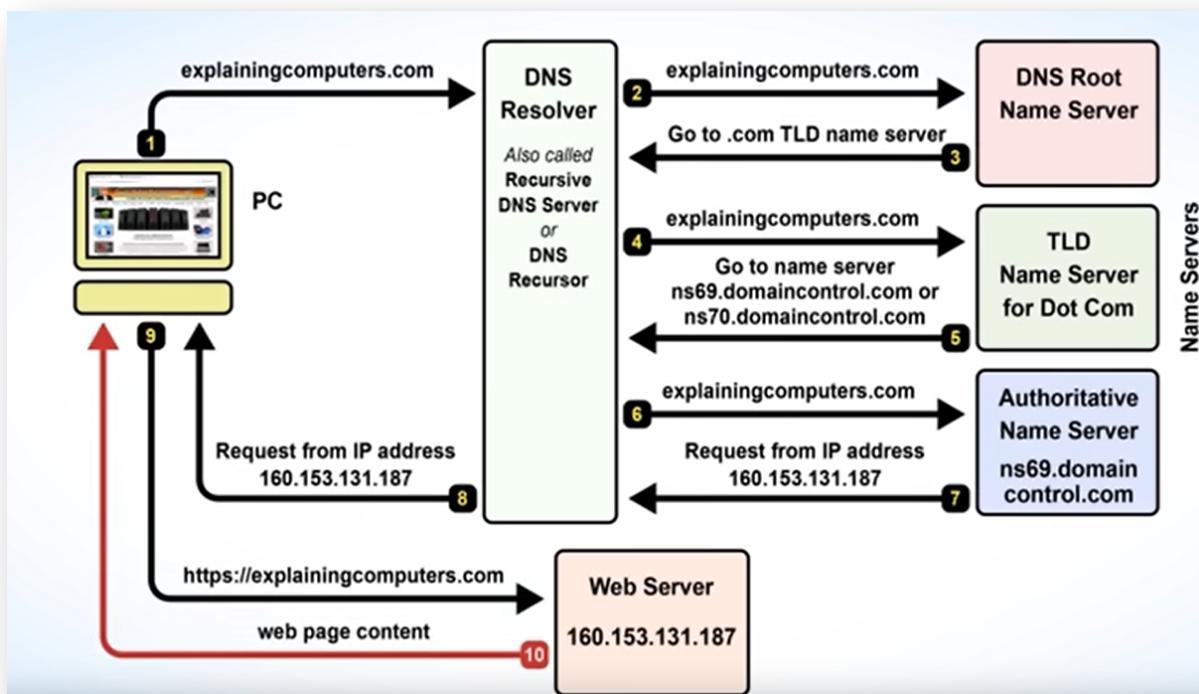
Record last updated 2024-12-03. Registration date 1988-04-14.

Los nameservers de los NIC de cada TLD deben responder a todas las peticiones de los dominios registrados bajo su TLD contestando con la dirección del nameserver autoritativo que sí tiene la respuesta.

Por ejemplo, en el TLD ".es", los servidores DNS de nic.es (los nameservers de red.es), deben contestar en qué DNS está delegada la gestión de un determinado dominio. Esta delegación se puede hacer:

- En servidores propios del proveedor de hosting.
- En servidores propios de la empresa
- En servidores "públicos" como Cloudflare, Google o AWS

A estos servidores delegados se les llama autoritativos porque tienen la gestión directa de la resolución de nombres e IPs y, por tanto devolverán una respuesta autoritativa al DNS resolver que lo solicitó.



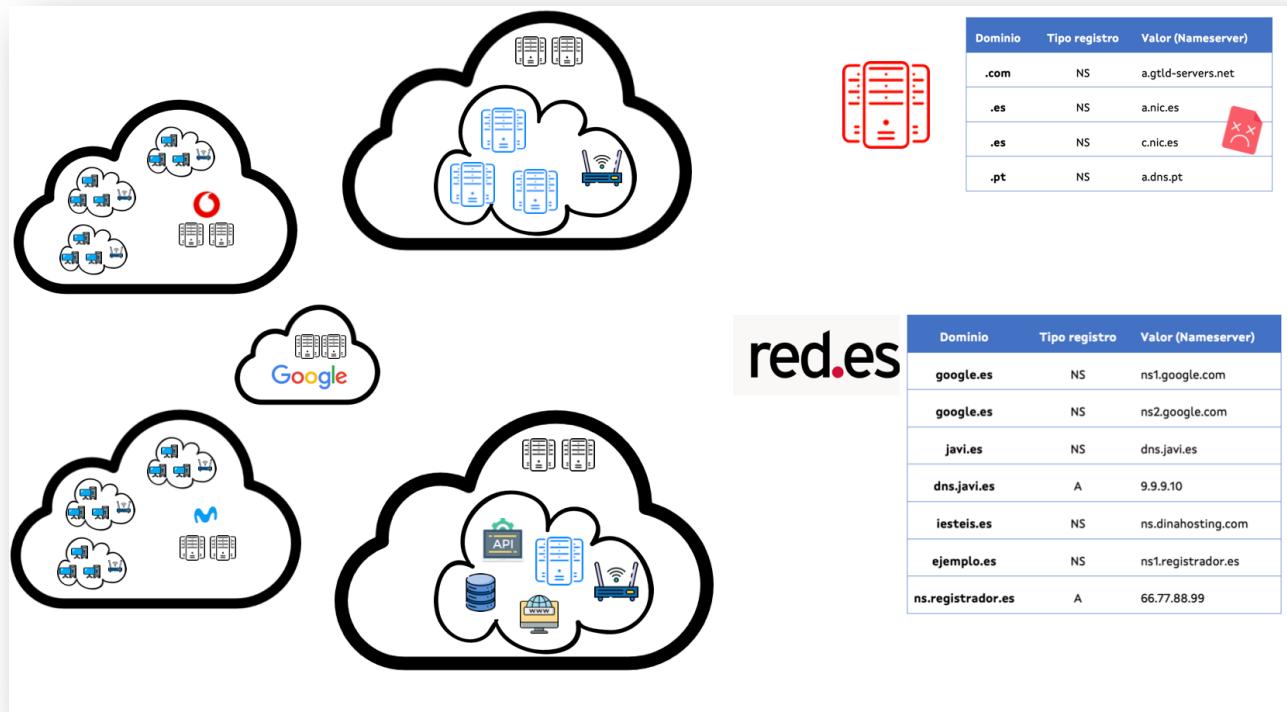
Finalmente, con la IP ya resuelta se devuelve a los DNS resolvers que la cachearán durante TTL segundos por si otro cliente hace la misma consulta.

Una relación nombre-IP que permanezca fija durante mucho tiempo como en una web personal o un servidor de correo pueden tener TTLs altos como por ejemplo 86400 segundo (1 día). Reduciremos el tráfico de red y el tiempo de respuesta porque todo quedará solucionado en el resolver.

En cambio, en un escenario de migración o con un balanceador de carga, interesa que el TTL sea bajo como por ejemplo de 300 segundo (5 minutos). Es cierto que habrá muchos fallos de caché que



requerirán de una consulta a un servidor raíz y consecuentemente se aumentará el tiempo de respuesta pero conseguiremos tener las rutas a las nuevas IPs mucho antes.



Registros DNS

A: es el tipo de registro más común. Sirve para resolver un nombre en una dirección IPv4 y se parecen a algo como midominio.es apunta a 11.22.33.44. Normalmente se acompaña de un campo TTL que indica en tiempo de caducidad hasta la siguiente actualización.

AAAA: funciona exactamente igual que el tipo A pero para direcciones IPv6.

CNAME: este tipo de registro sirve para crear un alias de manera que se puedan tener subdominios apuntando a la misma IP que el dominio comprado. Puede que tengamos en un mismo equipo, en una misma IP, varios servicios como un servidor Apache, un servidor de correo o un ftp. En ese caso nos gustaría (por estética y por ampliaciones futuras) tener varios CNAMEs que hagan apuntar www.midominio.es a dominio.es, imap.midominio.es a dominio.es y ftp.dominio.es a dominio.es

ANAME o ALIAS: este tipo de registro solo es permitido por algunos proveedores de DNS y funciona igual de CNAME pero apuntando a otro dominio. Podría ser algo parecido a otrodominio.es apunta a midominio.es.



MX: en este tipo de registro se apunta al servidor donde se deben entregar los correos electrónicos.

Tendrá un aspecto como "con una prioridad 10 usa el servidor mail1.midominio.com para los correos que tengan dominio @midominio.com". Es habitual encontrar listados con varios servidores de correo. En ese caso de deben identificar con distintos niveles de prioridad.

SOA: este tipo de registro indica el inicio de la autoridad y almacena información administrativa de la zona DNS. Una zona DNS es una sección del espacio de nombres del dominio que un administrador ha definido y delegado. Imaginemos que midominio.es tiene varios subdominios como tienda.midominio.es, blog.midominio.es y soporte.midominio.es. Si los dos primeros subdominios tienen poca carga y el tercero muchísima puede que el administrador decida crear dos zonas, una para tienda y blog, otra para soporte. Un registro de este tipo tendrá el aspecto de "el servidor primario (o master) de la zona es ns1.midominio.es tiene como contacto administrador (RNAME) este email (dos puntos son arroba) y la versión es la X".

NS: este registro representa el servidor autoritativo de la zona DNS. Sería algo similar a "el servidor ns1.midominio.es es el servidor autoritativo para el dominio midominio.es". Normalmente se incluyen al menos dos registros NS.

SRV: este registro se parece al CNAME con la diferencia que aquí se añade número de puerto. Una línea de este tipo de registro podría ser "con prioridad 10, un servicio en miservicio.midominio.es escucha en el puerto 1234 las peticiones que le lleguen al dominio midominio.es". De estos registros se aprovechan las aplicaciones que están programadas para conectarse así y normalmente descubrir servicios. Por ejemplo, los gestores de correo pueden detectar SRV de tipo SMTP o IMAP para enviar o recibir correo. Con solo escribir el dominio de tu buzón podrían detectar la IP/nombre y el puerto de los servidores de correo.

PTR: este registro es el complementario al registro A y resuelven las direcciones IP en nombres. Este tipo de registro lo usan los detectores de SPAM para comprobar si un correo que llega de un dominio es auténtico o no. Al servidor de correo le llega un correo donde se ve el dominio y la IP origen. Si al hacer una consulta PTR de esa IP sale el mismo dominio, el correo es auténtico.

TXT: en este registro se suele almacenar información administrativa en forma de texto como información general o de contacto. Puede tener una forma de "Para el dominio mi dominio.es la persona de contacto es Javi"

Hay más tipos de registros que se pueden consultar en <https://www.iana.org/assignments/dns-parameters/dns-parameters.xhtml#dns-parameters-4>



Registrar un dominio

Para registrar un nuevo dominio es necesario acudir a un registrador autorizado que pueda asignar un dominio en un TLD. Por ejemplo godaddy.com o AWS Route53.

Después del registro dos o más nameservers se asignarán al nuevo dominio. Probablemente el registrador dispone de sus propios servidores nameservers y nos asignará algunos. También está la posibilidad de seleccionar otros nameservers si se desea, de esta manera podríamos gestionar nosotros mismos nuestra propia zona.

A continuación, en los nameservers asignados hay que completar los registros DNS. Habitualmente tendremos que añadir un registro A apuntando a la IP donde está el equipo que queremos conectar a internet, un MX para configurar el correo electrónico, un CNAME para crear alias en subdominios y puede que un TXT para guardar información administrativa.

Después solo queda esperar a que los cambios se propaguen por internet.

Zonas DNS

Una zona es un área de control administrativo de una porción del espacio de nombres de un dominio. Es decir, cuando compramos un dominio podemos subdividirlo a nuestro gusto en subdominios especializados. Algo parecido a las carpetas y subcarpetas. Puede que tengamos una única carpeta "fotos" o puede que queramos dividir en subcarpetas.

Para cada zona DNS tendremos varios registros. En muchas ocasiones los registros pueden formar conjuntos. Por ejemplo, las líneas

www IN A 9.9.9.1 y www IN A 9.9.9.2

son dos líneas que están indicando registros de tipo A para el mismo subdominio aunque con diferente IP.

A un nivel más técnico, los registros siguen la siguiente estructura:

- Owner: donde indicamos el subdominio
- Class: donde pondremos IN de Internet
- Type: donde especificamos el tipo de registro, por ejemplo, A



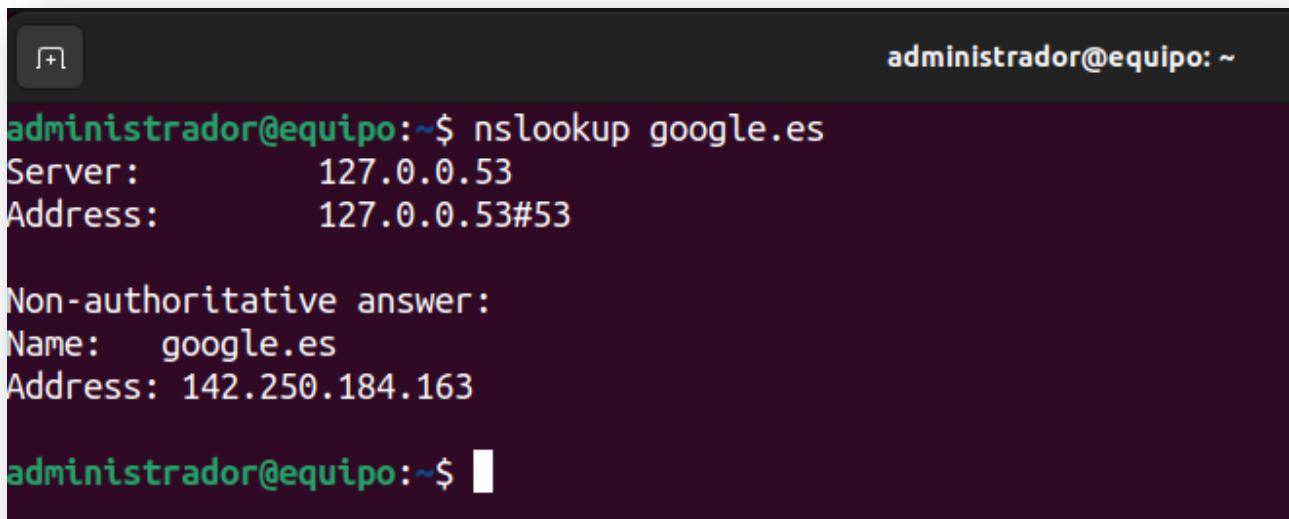
- Value: en este ejemplo pondríamos la IP con la que se resolvería ese subdominio. En algunos casos el valor puede tener más información, por ejemplo en los registros MX se puede indicar una prioridad y el nombre del servidor.

NSLOOKUP

El comando nslookup está disponible en todos los sistemas operativos habituales y, al igual que "ping" es una herramienta muy sencilla y, a la vez, tremadamente poderosa.

La herramienta nslookup se puede ejecutar por comando simplemente con el nombre y con todas las opciones que queramos pasadas por parámetro.

```
nslookup google.es
```



```
administrador@equipo:~$ nslookup google.es
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
Name:   google.es
Address: 142.250.184.163

administrador@equipo:~$
```

Obtendremos como resultado la IP de "google.es". Esta manera de ejecutar el comando nslookup está bien para pequeñas consultas o grandes automatizaciones. Durante el resto del documento ejecutaremos nslookup de modo interactivo simplemente escribiendo "nslookup" y entrando en su propio shell que identificaremos por el prompt de signo mayor (>).

En el funcionamiento por defecto (set type=A) simplemente tenemos que escribir el nombre del dominio y nslookup preguntará a su DNS configurado para resolver la IP.

En la siguiente captura se ve como la primera consulta la hace sobre sí mismo (127.0.0.53 al puerto 53) y recibe una respuesta no autoritativa indicando que la IP es la 142.250.200.99



```
administrador@equipo:~$ nslookup 1
> google.es 2
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
Name:   google.es 3
Address: 142.250.200.99
>
```

Podemos forzar el uso de un servidor DNS en concreto para hacer partir la consulta desde una rama distinta. En la siguiente captura podemos ver como mi equipo resuelve la IP de "Instagram.es" en la misma IP tanto si uso mi DNS como si uso el 8.8.8.8 de Google pero me devuelve otra dirección si uso el DNS 1.1.1.1 de CloudFlare.

```
administrador@equipo:~$ nslookup
administrador@equipo:~$ nslookup
> instagram.es
Server:      127.0.0.53 1
Address:     127.0.0.53#53

Non-authoritative answer:
Name:   instagram.es 2
Address: 157.240.5.12
> server 8.8.8.8
Default server: 8.8.8.8
Address: 8.8.8.8#53
> instagram.es
Server:      8.8.8.8
Address:     8.8.8.8#53 3

Non-authoritative answer:
Name:   instagram.es 4
Address: 157.240.5.12
> server 1.1.1.1
Default server: 1.1.1.1
Address: 1.1.1.1#53
> instagram.es
Server:      1.1.1.1
Address:     1.1.1.1#53 5

Non-authoritative answer:
Name:   instagram.es 6
Address: 31.13.83.8
>
```

Para cambiar de servidor solamente hay que indicar la palabra server con la IP del servidor DNS.



Hasta ahora todas las respuestas que hemos recibido son "no autoritativas" lo que significa que la información original no está en este servidor por lo que es posible que esté cacheada y, por tanto, desactualizada.

Si queremos averiguar cual o cuales son los servidores DNS autoritativos para un determinado nombre debemos cambiar el tipo de búsqueda con:

```
set type=NS
```

```
> wikipedia.es
Server:      192.168.0.1
Address:     192.168.0.1#53

Non-authoritative answer:
Name: wikipedia.es
Address: 185.15.58.226
>
> set type=NS
>
> wikipedia.es
Server:      192.168.0.1
Address:     192.168.0.1#53

Non-authoritative answer:
wikipedia.es    nameserver = ns0.wikimedia.org.
wikipedia.es    nameserver = ns1.wikimedia.org.
wikipedia.es    nameserver = ns2.wikimedia.org.

Authoritative answers can be found from:
>
> █
```

En esta captura podemos ver como cambiamos el tipo de consulta para ver los nameservers autoritativos que tiene "Wikipedia.es". Alguno de estos DNS es el principal, para consultarlos debemos cambiar el modo de la consulta a registro SOA (Start of authority) y volver a consultar el dominio.

```
> set type=SOA
>
> wikipedia.es
Server:      192.168.0.1
Address:     192.168.0.1#53

Non-authoritative answer:
wikipedia.es
    origin = ns0.wikimedia.org
    mail addr = hostmaster.wikimedia.org
    serial = 2024021317
    refresh = 43200
    retry = 7200
    expire = 1209600
    minimum = 3600

Authoritative answers can be found from:
> █
```



Instalación BIND9

Siempre, antes de hacer una instalación desde los repositorios de Linux es buena idea hacer un "sudo apt update" para actualizar el listado de aplicaciones y así poder elegir de entre las últimas versiones.

Para descargar e instalar "BIND9" simplemente hay que escribir "sudo apt install bind9" y aceptar.

```
administrador@equipo:~$ sudo apt install bind9
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  bind9-dnsutils bind9-host bind9-libs bind9-utils
Paquetes sugeridos:
  bind-doc resolvconf
Se instalarán los siguientes paquetes NUEVOS:
  bind9 bind9-utils
Se actualizarán los siguientes paquetes:
  bind9-dnsutils bind9-host bind9-libs
3 actualizados, 2 nuevos se instalarán, 0 para eliminar y 335 no actualizados.
Se necesita descargar 1.842 kB de archivos.
Se utilizarán 1.525 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n]
```

Como es habitual, en el proceso de instalación se crean y configuran algunos aspectos a nivel de sistema operativo que conviene conocer:

- Se ha creado un usuario del sistema (no tiene home ni shell) llamado "bind"
- Se ha creado un grupo llamado "bind"
- Se ha creado el servicio "bind9"
- Se ha creado la carpeta /var/cache/bind que será usada para cachear las consultas durante su TTL
- Se ha creado una carpeta en /etc/bind con los archivos de configuración de BIND9

En una instalación sobre Ubuntu 22.04 el servicio "bind9" se encuentra en ejecución y configurado en "enable" lo que hará que se ejecute en cada arranque. Para ver el estado del servicio ejecutaremos

```
sudo systemctl status bind9
```

Y con cada cambio que hagamos en la configuración debemos reiniciarlo con un "restart" y volver a comprobar su estado con un "status".



El servidor DNS usa el puerto 53 tanto TCP como UDP así que otra buena forma de comprobar que todo está funcionando OK es hacer un

```
ss -ltun
```

Con este comando listamos las conexiones de escucha tanto tcp como udp y mostramos el número de puerto usado. Si encontramos que nuestro equipo está escuchando en la 53 es que todo va ok.

```
administrador@equipo:~$ ss -ltnw
Netid State Recv-Q Send-Q Local Address:Port          Peer Address:Port Process
icmp6 UNCONN 0      0           *:58                  *:*
udp  UNCONN 0      0           192.168.0.35:53       0.0.0.0:*
udp  UNCONN 0      0           192.168.0.35:53       0.0.0.0:*
udp  UNCONN 0      0           127.0.0.1:53        0.0.0.0:*
udp  UNCONN 0      0           127.0.0.1:53        0.0.0.0:*
udp  UNCONN 0      0           127.0.0.53:lo:53     0.0.0.0:*
udp  UNCONN 0      0           0.0.0.0:5353       0.0.0.0:*
udp  UNCONN 0      0           0.0.0.0:631        0.0.0.0:*
udp  UNCONN 0      0           0.0.0.0:55189      0.0.0.0:*
udp  UNCONN 0      0           [:1]:53            [:]:*
udp  UNCONN 0      0           [:1]:53            [:]:*
udp  UNCONN 0      0           [fe80::4cfe:20a0:85c:ff6a]@enp0s1:53  [:]:*
udp  UNCONN 0      0           [fe80::4cfe:20a0:85c:ff6a]@enp0s1:53  [:]:*
udp  UNCONN 0      0           [:]:5353          [:]:*
udp  UNCONN 0      0           [:]:54071         [:]:*
tcp  LISTEN 0      10          127.0.0.1:53        0.0.0.0:*
tcp  LISTEN 0      10          127.0.0.1:53        0.0.0.0:*
tcp  LISTEN 0      128         127.0.0.1:631       0.0.0.0:*
tcp  LISTEN 0      10          192.168.0.35:53       0.0.0.0:*
tcp  LISTEN 0      10          192.168.0.35:53       0.0.0.0:*
tcp  LISTEN 0      5           127.0.0.1:953       0.0.0.0:*
tcp  LISTEN 0      5           127.0.0.1:953       0.0.0.0:*
tcp  LISTEN 0      4096        127.0.0.53:lo:53     0.0.0.0:*
tcp  LISTEN 0      128         [:1]:631          [:]:*
tcp  LISTEN 0      10          [fe80::4cfe:20a0:85c:ff6a]@enp0s1:53  [:]:*
tcp  LISTEN 0      10          [fe80::4cfe:20a0:85c:ff6a]@enp0s1:53  [:]:*
tcp  LISTEN 0      5           [:1]:953          [:]:*
tcp  LISTEN 0      5           [:1]:953          [:]:*
tcp  LISTEN 0      10          [:1]:53           [:]:*
tcp  LISTEN 0      10          [:1]:53           [:]:*
administrador@equipo:~$
```

Configuración de BIND9

Lo primero que hay que hacer es configurar una zona DNS. Un único servidor DNS con BIND puede mantener una o varias zonas simultáneamente. En los servidores proveedores de hosting, BIND maneja sin problemas millones de zonas

Para cada zona que queramos manejar con BIND9 tenemos que configurar una **zona directa** que hará la conversión del dominio a una IP. También debemos configurar una **zona inversa** encargada de hacer la conversión de IP a nombre. Posteriormente debemos crear los archivos de zona directa e inversa.

Por último tendremos que reiniciar el servidor DNS y comprobar su funcionamiento con herramientas como ping y nslookup.

En una instalación limpia tendremos los siguientes archivos de configuración:



```
administrador@equipo:/etc/bind$ ls -la
total 64
drwxr-sr-x  2 root bind  4096 feb  5 13:07 .
drwxr-xr-x 127 root root 12288 feb  5 13:07 ..
-rw-r--r--  1 root root  2403 ene 28 15:30 bind.keys
-rw-r--r--  1 root root   237 sep 23 23:16 db.0
-rw-r--r--  1 root root   271 sep 23 15:35 db.127
-rw-r--r--  1 root root   237 sep 23 15:35 db.255
-rw-r--r--  1 root root   353 sep 23 15:35 db.empty
-rw-r--r--  1 root root   270 sep 23 15:35 db.local
-rw-r--r--  1 root bind  463 sep 23 23:16 named.conf
-rw-r--r--  1 root bind  498 sep 23 15:35 named.conf.default-zones
-rw-r--r--  1 root bind  165 sep 23 15:35 named.conf.local
-rw-r--r--  1 root bind  846 sep 23 15:35 named.conf.options
-rw-r----- 1 bind bind  100 feb  5 13:07 rndc.key
-rw-r--r--  1 root root 1317 sep 23 15:35 zones.rfc1918
administrador@equipo:/etc/bind$
```

El archivo "named.conf" es el archivo principal. Dentro de este archivo solo se mencionan los archivos "named.conf.default-zones", "named.conf.local" y "named.conf.options".

En el archivo "named.conf.default-zones" tenemos la definición de algunas zonas comunes. En este archivo se definen sus zonas y se hace referencia a su archivo.

```
administrador@equipo:/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 localhost forward and reverse zones, and for
// broadcast zones as per RFC 1912

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

zone "127.in-addr.arpa" {
    type master;
    file "/etc/bind/db.127";
};

zone "0.in-addr.arpa" {
    type master;
    file "/etc/bind/db.0";
};

zone "255.in-addr.arpa" {
```

La primera zona se denomina punto ". ". Y hace referencia a los servidores raíz. El archivo al que hace referencia contiene los registros DNS de los 13 servidores raíz.



La interpretación del archivo "root.hints" se interpreta en forma de 4 columnas. La primera es el nombre completo del dominio. Si nos fijamos veremos que todos los dominios acaban en punto (.) por lo que solamente un punto significa la raíz. La segunda columna es el TTL. La tercera columna es el tipo de registro y la cuarta columna es el valor. Por ejemplo, la primera línea empieza por un punto indicando que se refiere a la raíz, su TTL , un registro de tipo NS (indica el servidor DNS) y el nombre del servidor DNS.

La siguiente línea empieza por un nombre de dominio (el que definimos en la línea anterior), su TTL, un tipo de registro A (para relacionarlo con una IP) y como valor la IP de la relación.

Con solo esas líneas, nuestro servidor DNS sabría que en caso de necesitar acudir a un servidor raíz (.) acudiría a a.root-servers.net. Como tampoco conoce la IP de esa dirección volvería a hacer una consulta pero en este caso no con el dominio raíz (como antes) sino que ahora consultando la IP de a.root-servers.net. Se le devolvería la IP 198.41.0.4 y allí seguiría la consulta.

```
;
;                               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
;
```

El resto de las zonas son la de localhost que resuelve las consultas locales, las zonas de resolución inversa para las IPs de loopback (127), la de subred (0) y la de broadcast(255).

El archivo "named.conf.options" contiene información general sobre el servidor de BIND9. Modificaremos este archivo si queremos añadir "forwarders". Esto se utiliza cuando tenemos un servidor DNS gestionado por nosotros y queremos que antes de hacer una consulta a los DNS raíz se la haga antes a otro servidor DNS o a alguno de los DNS ofrecidos por nuestro ISP.



```

administrador@equipo:/etc/bind$ cat 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 {
    //     0.0.0.0;
    // };

    //=====
    // 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
    //=====
    dnssec-validation auto;

    listen-on-v6 { any; };
};

administrador@equipo:/etc/bind$ █

```

Otro archivo de configuración es el "named.conf.local". Este archivo está vacío, solo tiene unas líneas comentadas y es el que usaremos para configurar nuestras zonas directa e inversa.

En nuestro escenario solo tendremos una única zona pero es de buenas prácticas crear un directorio para guardar las tablas de registro de cada zona y además hacerlo en un archivo independiente para cada zona. Crearemos la carpeta /etc/bind/zonas con este objetivo.

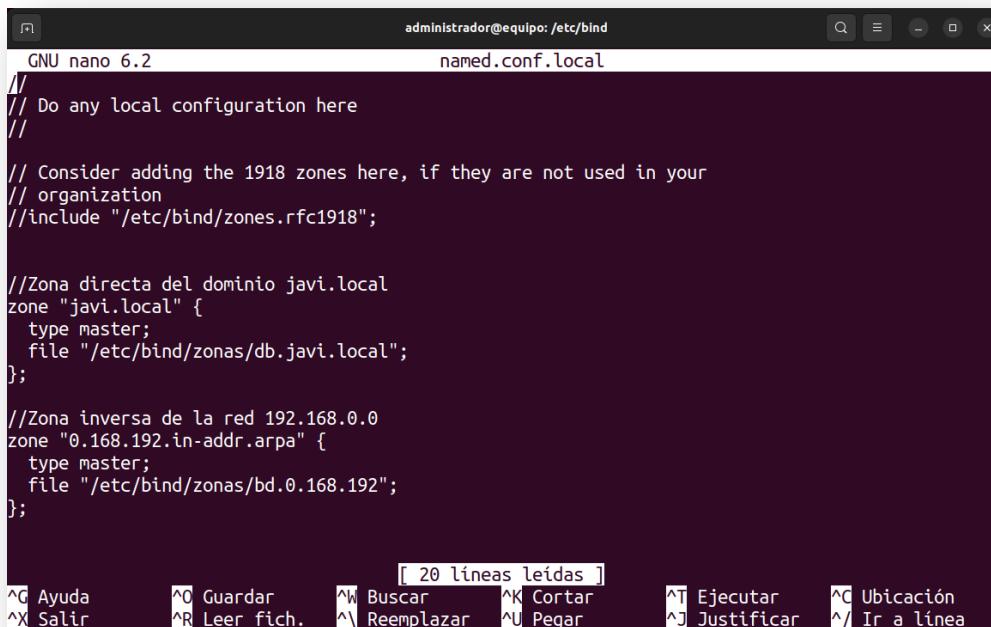
Configuramos nuestra zona directa para el dominio "javi.local" indicando que el archivo con los registros DNS estará en la ruta "/etc/bind/zonas/db.javi.local". No es obligatorio pero por convenio y buenas prácticas nombramos a estos archivos de registros DNS como "db" y en nombre completo del dominio administrado.

También configuramos la zona inversa de una manera similar. El cambio más evidente es en el nombre de la zona en el que usamos una especie de dominio que tiene en su parte derecha "in-addr-arpa" (internet address address and routing parameter area)

Para configurar la zona "javi.local" en mi red local modificaré el archivo "named.conf.local" dejándolo como en la siguiente captura.



Es habitual tomar el archivo "db.local" como plantilla para guardar los registros DNS de cada zona. Para



```

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 del dominio javi.local
zone "javi.local" {
    type master;
    file "/etc/bind/zonas/db.javi.local";
};

//Zona inversa de la red 192.168.0.0
zone "0.168.192.in-addr.arp" {
    type master;
    file "/etc/bind/zonas/db.0.168.192";
};

```

[20 líneas leídas]

^G Ayuda ^O Guardar ^W Buscar ^K Cortar ^T Ejecutar ^C Ubicación
^X Salir ^R Leer fich. ^M Reemplazar ^U Pegar ^J Justificar ^/ Ir a línea

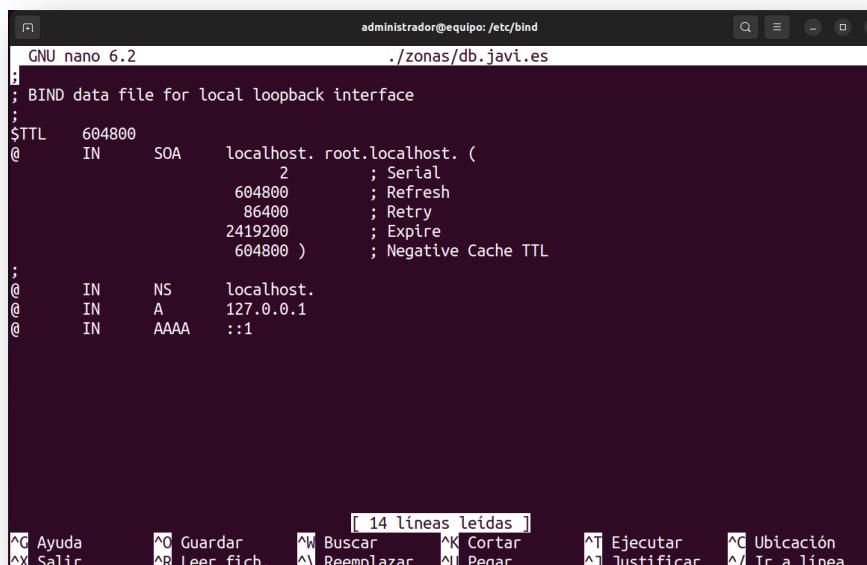
ello copiamos este archivo en la ruta que especificamos anteriormente y lo editamos.

```

sudo cp /etc/bin/db.local /etc/bind/zonas/db.javi.local
sudo nano /etc/bin/zonas/db.javi.local

```

El contenido inicial de este archivo es el siguiente:



```

GNU nano 6.2                               ./zonas/db.javi.local
; BIND data file for local loopback interface
;
$TTL    604800
@      IN      SOA     localhost. root.localhost. (
                        2                   ; Serial
                        604800              ; Refresh
                        86400               ; Retry
                        2419200             ; Expire
                        604800 )            ; Negative Cache TTL
;
@      IN      NS      localhost.
@      IN      A       127.0.0.1
@      IN      AAAA   ::1

```

[14 líneas leídas]

^G Ayuda ^O Guardar ^W Buscar ^K Cortar ^T Ejecutar ^C Ubicación
^X Salir ^R Leer fich. ^M Reemplazar ^U Pegar ^J Justificar ^/ Ir a línea

Este archivo representa la base de datos de registros DNS que aplicaremos a una zona concreta.



La primera línea después de los comentarios es una variable global de TTL en el que indicamos el tiempo en segundos que un registro es válido por defecto (podremos modificar individualmente otros registros).

A partir de aquí ya todo son registros que siguen el siguiente patrón:

- Propietario. Si acaba con punto es un nombre DNS FQDN. Si acaba sin punto es un subdominio (NombreRelativo+NombreZona)
- Clase: indica familia de protocolos. Siempre usaremos "IN"
- Tipo: indica el tipo de registro que definimos.
- Valor: datos de configuración para ese registro.

Los tipos de registro más habituales son:

- SOA: (Start of authority) que indica cual es el DNS primario de la zona. Hay que indicar nombre del DNS primario, correo de contacto (se usa el punto en lugar de la arroba) y otros parámetros.
- NS: nombre de los DNS autoritativos de esta zona. Como mínimo debe estar el indicado en SOA.
- A: Indica que este registro se debe usar para traducir un nombre a una IP (nombre -> IP)
- CNAME: Usado para que varios nombres apunten a una misma IP.

```
GNU nano 6.2                               ./zonas/db.javi.local
;
; BIND data file for local loopback interface
;
$TTL    604800
javi.local.     IN      SOA    dns.javi.local. administrador.javi.local. (
                           2          ; Serial
                           604800    ; Refresh
                           86400     ; Retry
                          2419200   ; Expire
                           604800 )  ; Negative Cache TTL
;
javi.local.     IN      NS     dns.javi.local.
dns            IN      A      192.168.0.35
router         IN      A      192.168.0.1
nodo1.javi.local. IN      A      192.168.0.11
nodo2          IN      A      192.168.0.250
puertaenlace  IN      CNAME  router
[ 19 líneas leídas ]
^G Ayuda      ^O Guardar     ^W Buscar      ^K Cortar      ^T Ejecutar     ^C Ubicación
^X Salir      ^R Leer fich.  ^E Reemplazar  ^U Pegar       ^J Justificar  ^I Ir a línea
```



En el registro SOA estoy indicando que el DNS raíz en mi dominio FQDN es "dns.javi.local." Fíjate que también lo estoy indicando con un nombre FQDN porque acaba en punto. También indico la dirección de correo electrónico de algún responsable o administrador. Escribiremos este correo sin usar la arroba porque en los archivos de configuración de BIND la arroba representa un alias del dominio (el usado en la declaración de "name.conf.local"+punto). En algunos archivos de configuración veréis una "@", esto es un alias de nombre de dominio FQDN. En el caso de este ejemplo "@" representa a "javi.local." En otras ocasiones encontraréis un espacio vacío que representa al propietario del último registro definido.

El siguiente registro es un NS en el que estamos indicando que "dns.javi.local." es un DNS autoritativo de "javi.local."

Los 4 siguientes registros son de tipo A en el que asociamos un nombre de dominio a una IP. Como mínimo deberíamos tener un registro tipo A por cada NS. En estos 4 registros lo más interesante es las distintas maneras que tenemos de definir el propietario de la primera columna. Si acaba en punto es un nombre FQDN completo. Si acaba sin punto entonces solo se está indicando el subdominio y hay que añadirle el nombre de la zona completa (javi.local.)

El último registro es un alias en el que le estamos diciendo que el subdominio "puertaenlace" (puertaenlace.javi.local.) es un alias de "router" (router.javi.local.).

Una vez configurada la zona directa pasamos a configurar la zona inversa. Las buenas prácticas nos dicen que es bueno definirlo en un archivo aparte así que usamos el que definimos anteriormente en "named.conf.local". Podemos copiar la plantilla que usamos anteriormente o reutilizar el archivo de la zona directa que acabamos de crear.

```
sudo cp /etc/bind/zonas/db.javi.local /etc/bind/zonas/db.0.168.192
```

Después lo editamos y lo configuramos de una manera parecida a la zona directa. En este archivo hay dos novedades.

La primera consiste en el uso del registro PTR para asignar a una IP un nombre de dominio FQDN. En estos registros, el propietario debe ser una IP escrita de manera inversa con "in-addr.arpa." al final o bien solo el número del host sin punto para que lo complete solo. El valor de ese registro debe ser un nombre FQDN completo (con punto al final).



```
GNU nano 6.2                                administrador@equipo: /etc/bind
                                              ./zonas/db.0.168.192

;

; BIND data file for local loopback interface
;

$TTL    604800
0.168.192.in-addr.arpa. IN      SOA    dns.javi.local. administrador.javi.local. (
                                  2           ; Serial
                                  604800     ; Refresh
                                  86400      ; Retry
                                 2419200    ; Expire
                                 604800 )   ; Negative Cache TTL
;

0.168.192.in-addr.arpa.      IN      NS      dns.javi.local.
35                           IN      PTR     dns.javi.local.
1                            IN      PTR     router.javi.local.
11                           IN      PTR     nodo1.javi.local.
250.0.168.192.in-addr.arpa. IN      PTR     nodo2.javi.local.

[ 19 líneas leidas ]
^G Ayuda      ^O Guardar      ^W Buscar      ^K Cortar      ^T Ejecutar      ^C Ubicación
^X Salir      ^R Leer fich.  ^V Reemplazar  ^U Pegar       ^J Justificar  ^/ Ir a línea
```

El archivo "/etc/bind/db.0.168.192" quedaría como en la captura anterior. Si nos fijamos que están las IPs de los nameservers y todos aquello registros "A" que teníamos en la zona directa.

Una vez que tenemos editados los archivos de configuración podemos usar una de las herramientas que incluye BIND para la comprobación de la sintaxis de los archivos de configuración ya que es fácil y habitual equivocarse al escribir.

Con "named-checkconf" comprobamos la sintaxis de los archivos en los que definimos las zonas en el archivo "/etc/bin/named.conf.local"

```
named-checkconf /etc/bind/named.conf.local
```

Si no hay ninguna salida del comando anterior significa que el archivo está escrito sin errores de sintaxis. Ojo, es no quiere decir que el archivo sea correcto y haga lo que pretendíamos.

Con "named-checkzone" comprobamos los archivos de zona y para ello tenemos que pasarle la zona (javi.local o 0.168.192.in-addr.arpa) sin punto final y la ruta al archivo de zona donde están definidos los registros que se le aplican.

```
named-checkzone javi.local /etc/bind/zonas/db.javi.local
```



Este comando sí devuelve una salida indicando si el archivo está "OK" en su sintaxis.

Debemos comprobar todas las zonas, tanto la directa como la inversa, por ello también ejecutamos el siguiente comando:

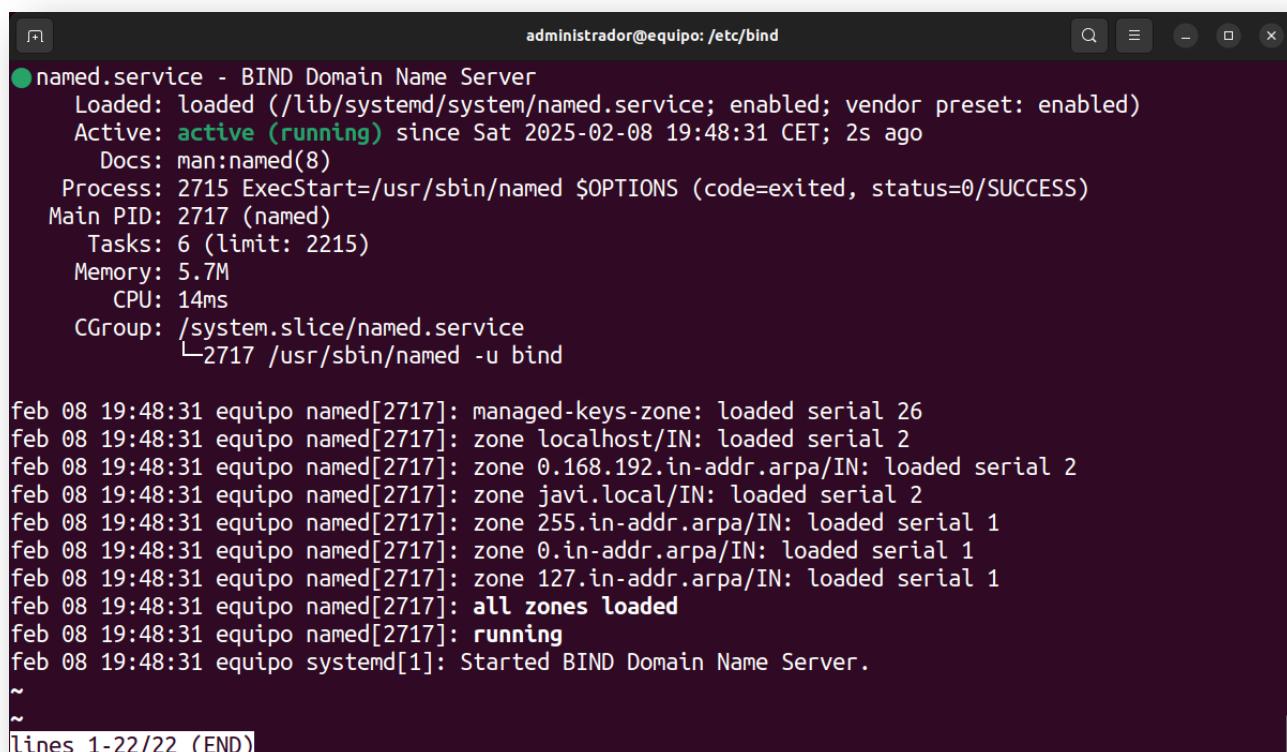
```
named-checkzone 0.168.192.in-addr.arpa /etc/binnd/zonas/db.0.168.192
```

Una vez que hemos comprobado la sintaxis y corregido los errores si los hubiera es momento de reiniciar el servicio de BIND9 para forzar la lectura de los archivos de configuración con los nuevos datos. Lo hacemos con el siguiente comando:

```
sudo systemctl restart bind9
```

Una vez reiniciado hacemos un status del servicio para ver si hubo errores y confirmamos que se está escuchando en el puerto 53 (tcp y udp) de nuestra máquina.

```
sudo systemctl status bind9
ss -lutan
```



```
administrador@equipo: /etc/bind
● named.service - BIND Domain Name Server
   Loaded: loaded (/lib/systemd/system/named.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2025-02-08 19:48:31 CET; 2s ago
     Docs: man:named(8)
     Process: 2715 ExecStart=/usr/sbin/named $OPTIONS (code=exited, status=0/SUCCESS)
    Main PID: 2717 (named)
      Tasks: 6 (limit: 2215)
     Memory: 5.7M
        CPU: 14ms
       CGroup: /system.slice/named.service
               └─2717 /usr/sbin/named -u bind

feb 08 19:48:31 equipo named[2717]: managed-keys-zone: loaded serial 26
feb 08 19:48:31 equipo named[2717]: zone localhost/IN: loaded serial 2
feb 08 19:48:31 equipo named[2717]: zone 0.168.192.in-addr.arpa/IN: loaded serial 2
feb 08 19:48:31 equipo named[2717]: zone javi.local/IN: loaded serial 2
feb 08 19:48:31 equipo named[2717]: zone 255.in-addr.arpa/IN: loaded serial 1
feb 08 19:48:31 equipo named[2717]: zone 0.in-addr.arpa/IN: loaded serial 1
feb 08 19:48:31 equipo named[2717]: zone 127.in-addr.arpa/IN: loaded serial 1
feb 08 19:48:31 equipo named[2717]: all zones loaded
feb 08 19:48:31 equipo named[2717]: running
feb 08 19:48:31 equipo systemd[1]: Started BIND Domain Name Server.
~
~
lines 1-22/22 (END)
```

Llega el momento de las pruebas y para ello usaremos "nslookup", cambiamos se servidor DNS y probamos a hacer peticiones A, PTR, NS, SOA... Por último podemos configurar los datos IP de un nodo para que use este DNS y poder hacer pruebas con el comando "ping"



Archivo de configuración global

En el archivo "named.conf.options" se guardan las opciones globales del servidor DNS. Por defecto solo tenemos explícitamente configurado las opciones:

- Directory, donde se especifica la ruta de la carpeta donde BIND dejará la base de datos en la que guarda configuraciones y cachés.
- Forwarders. En principio está comentada pero aquí es donde podríamos especificar las IPs de otros resolver a los que reenviar las consultas DNS
- Listen-on-v6 y listen-on definen en cual o cuales interfaces escucha el servidor DNS. Por defecto escucha en todas las interfaces pero también podemos restringirlo indicando la IP (v4 o v6) de la interfaz del servidor o bien alguna palabra reservada como "any".
- Allow-query: este parámetro indica la IP, subred (con máscara) en las que están permitidas las peticiones. Si no se especifica están todas las IPs permitidas.
- Allow-recursion: en este parámetro indicamos las IPs o redes que pueden realizar consultas recursivas.
- Recursion: indica si este servidor DNS permite o no consultas recursivas. Si lo activamos, el servidor DNS buscará resolución en otros servidores DNS. Si lo desactivamos, el servidor solo resolverá exclusivamente los nombres de su zona.

La combinación de estos parámetros puede variar dependiendo del uso que queramos dar a nuestro servidor DNS. Pongamos dos ejemplos extremos.

En el primer escenario imaginemos una empresa que quiere usar un DNS en su red interna. Los equipos de su red querrán resolver los nombres de su infraestructura como "impresra" o "dc01" (los autoritativos) y también querrán resolver nombres externos como "Google.es" y, por lo tanto tendrán que preguntar a otros DNS y comportarse como recursivos.

```

1. options {
2.   directory "/var/cache/bind";
3.
4.   // Escuchar en todas las interfaces disponibles
5.   listen-on { any; };
6.   listen-on-v6 { any; };
7.
8.   // Permitir consultas solo desde la red interna
9.   allow-query { 192.168.6.0/24; 127.0.0.1; };
10.
11.  // Habilitar recursión para resolver cualquier dominio
12.  recursion yes;
13.  allow-recursion { 192.168.6.0/24; 127.0.0.1; };
14.
15.  // Configurar servidores externos para reenviar consultas que no pueda resolver localmente
16.  forwarders {
17.    8.8.8.8; // Google DNS
18.    1.1.1.1; // Cloudflare DNS
19.  };
20.
21.  // Habilitar caché para mejorar el rendimiento

```



```

22.     max-cache-size 256M;
23.     max-cache-ttl 86400;
24.
25.     // Registrar estadísticas para analizar tráfico
26.     statistics-file "/var/cache/bind/named.stats";
27.
28.     // Seguridad: evitar ataques de amplificación DNS
29.     rate-limit {
30.         responses-per-second 5;
31.     };
32. };
33.

```

Otro escenario distinto podría ser el de un servidor DNS autoritativo público como los que pueden tener registradores como Dinahosting. Estos servidores solo deberían responder a consultas sobre sus dominios registrados y no deberían perder el tiempo resolviendo consultas generalistas sobre otros dominios que no gestiona.

```

1. options {
2.     directory "/var/cache/bind";
3.
4.     // Escuchar en todas las interfaces disponibles
5.     listen-on { any; };
6.     listen-on-v6 { any; };
7.
8.     // Permitir consultas desde cualquier cliente en Internet
9.     allow-query { any; };
10.
11.    // NO permitir recursión: solo responde sobre sus propias zonas
12.    recursion no;
13.
14.    // Seguridad adicional: evitar ataques de transferencia de zona no autorizados
15.    allow-transfer { none; };
16.
17.    // Deshabilitar respuesta NXDOMAIN para seguridad
18.    auth-nxdomain no;
19. };
20.

```

No es necesario entender al detalle la configuración pero sí tener claro que con ella se puede adaptar un servidor DNS para escenarios muy concretos.

Práctica

En esta práctica montaremos un servidor DNS para identificar con nombres ordenadores de la red interna.

En este escenario tendremos 2 equipos. Uno hará las funciones de cliente y el otro se servidor. El servidor recibirá tantos nombres como servicios tenga, por ahora solamente DNS.

Los únicos equipos con nombre que identificaremos serán "dns" y "router"



Configurar IP fija en el servidor

Los servidores deben tener IP fija para poder ser localizados. Podemos asignar una IP fija a través de un servidor DHCP configurando la reserva de una IP concreta a la MAC de la tarjeta de red adecuada del servidor.

También podemos hacerlo usando “netplan” y sus archivos de configuración o la herramienta gráfica “Network Manager” con la que se configura la tarjeta de red en los Ubuntu de escritorio.

La configuración de “netplan” podría ser:

```
1. network:  
2.   version: 2  
3.   ethernets:  
4.     enp0s1:  
5.       addresses:  
6.         - 192.168.0.80/24  
7.       routes:  
8.         - to: default  
9.           via: 192.168.0.1  
10.      nameservers:  
11.        addresses:  
12.          - 8.8.8.8
```

Instalación BIND9

```
1. sudo apt update  
2. sudo apt install bind9
```

Comprobar firewall en el puerto 53

```
1. sudo ufw status
```

En caso de tener el firewall activo habría que dar permiso a BIND9 con el siguiente comando:

```
1. sudo ufw allow bind9
```

Comprobar estado del servicio BIND9

```
1. sudo systemctl status bind9
```



En esta captura se puede ver que el servicio DNS de BIND9 está arrancado pero con errores, no funcionará bien.

```
● named.service - BIND Domain Name Server
   Loaded: loaded (/lib/systemd/system/named.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2025-02-11 09:15:37 CET; 7min ago
     Docs: man:named(8)
  Process: 4139 ExecStart=/usr/sbin/named $OPTIONS (code=exited, status=0/SUCCESS)
 Main PID: 4140 (named)
    Tasks: 8 (limit: 2226)
   Memory: 5.5M
      CPU: 39ms
     CGroup: /system.slice/named.service
             └─4140 /usr/sbin/named -u bind

feb 11 09:15:37 equipo named[4140]: SERVFAIL unexpected RCODE resolving './NS/IN': 193.0.14.129>
feb 11 09:15:37 equipo named[4140]: SERVFAIL unexpected RCODE resolving './DNSKEY/IN': 192.5.5.>
feb 11 09:15:37 equipo named[4140]: SERVFAIL unexpected RCODE resolving './NS/IN': 192.5.5.241#>
feb 11 09:15:37 equipo named[4140]: SERVFAIL unexpected RCODE resolving './DNSKEY/IN': 198.41.0>
feb 11 09:15:37 equipo named[4140]: SERVFAIL unexpected RCODE resolving './NS/IN': 198.41.0.4#53
feb 11 09:15:37 equipo named[4140]: SERVFAIL unexpected RCODE resolving './DNSKEY/IN': 192.112.>
feb 11 09:15:37 equipo named[4140]: managed-keys-zone: Unable to fetch DNSKEY set '.':: failure
feb 11 09:15:37 equipo named[4140]: SERVFAIL unexpected RCODE resolving './NS/IN': 192.112.36.4#
feb 11 09:15:37 equipo named[4140]: network unreachable resolving './NS/IN': 2001:dc3::35#53
feb 11 09:15:37 equipo named[4140]: resolver priming query complete: failure
~
```

La siguiente captura muestra un servicio DNS ejecutándose correctamente

```
administrador@equipo:/etc/bind$ sudo systemctl status bind9
● named.service - BIND Domain Name Server
   Loaded: loaded (/lib/systemd/system/named.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2025-02-11 11:02:32 CET; 5s ago
     Docs: man:named(8)
  Process: 4806 ExecStart=/usr/sbin/named $OPTIONS (code=exited, status=0/SUCCESS)
 Main PID: 4807 (named)
    Tasks: 6 (limit: 2226)
   Memory: 5.5M
      CPU: 17ms
     CGroup: /system.slice/named.service
             └─4807 /usr/sbin/named -u bind -4

feb 11 11:02:32 equipo named[4807]: configuring command channel from '/etc/bind/rndc.key'
feb 11 11:02:32 equipo named[4807]: command channel listening on 127.0.0.1#953
feb 11 11:02:32 equipo named[4807]: managed-keys-zone: loaded serial 4
feb 11 11:02:32 equipo named[4807]: zone 0.in-addr.arpa/IN: loaded serial 1
feb 11 11:02:32 equipo named[4807]: zone 255.in-addr.arpa/IN: loaded serial 1
feb 11 11:02:32 equipo named[4807]: zone localhost/IN: loaded serial 2
feb 11 11:02:32 equipo named[4807]: zone 127.in-addr.arpa/IN: loaded serial 1
feb 11 11:02:32 equipo named[4807]: all zones loaded
feb 11 11:02:32 equipo systemd[1]: Started BIND Domain Name Server.
feb 11 11:02:32 equipo named[4807]: running
administrador@equipo:/etc/bind$
```



Solo usar IPv4

Sudo nano /etc/default/named

Editamos el archivo para dejarlo de la siguiente manera:

```

1. #
2. # run resolvconf?
3. RESOLVCONF=yes
4.
5. # startup options for the server
6. OPTIONS="-u bind -4"
7.

```

Con ello hacemos que nuestro servidor DNS solo gestione IP versión 4 y desaparezcan todos los mensajes relacionados con IP versión 6.

Configurar los reenviadores (forwarders)

```
1. sudo nano /etc/bind/named.conf.options
```

Modificamos el archivo para dejarlo de la siguiente manera:

```

1. options {
2.
3.     directory "/var/cache/bind";
4.
5.     listen-on { any; };
6.
7.     allow-query { localhost; 192.168.0.0/24; };
8.
9.     forwarders {
10.         8.8.8.8;
11.     };
12.
13.     dnssec-validation no;
14.
15. };

```

Revisar archivos de configuración

```
1. named-checkconf
```



Actualizar IPs de nameservers raíz

Consultar en "named.conf.default-zones" donde se guardan los registros con los servidores raíz y descargarnos el archivo actual de la web oficial.

```
1. sudo wget -O /usr/share/dns/root.hints https://www.internic.net/domain/named.cache
```

Agregar zonas

Debemos modificar el archivo "/etc/bind/named.conf.local" para dejarlo de la siguiente manera:

```
1. zone "javi.local" IN {
2.     type master;
3.     file "/etc/bind/db.javi.local";
4. };
5.
6.
7. zone "1.168.192.in-addr.arpa" {
8.     type master;
9.     file "/etc/bind/db.0.168.192";
10.};
```

Después crearemos los archivos de registros DNS de cada zona en la ubicación y con el nombre que hemos definido. Aprovechamos "db.local" copiándolo y modificando el nombre.

El archivo de la zona directa debe quedar así:

```
1. ;
2. ; BIND data file for javi.local
3. ;
4. $TTL 604800
5. @ IN SOA dns.javi.local. root.javi.local. (
6.           2          ; Serial
7.           604800      ; Refresh
8.           86400       ; Retry
9.           2419200     ; Expire
10.          604800 )   ; Negative Cache TTL
11. ;
12.      IN NS dns.javi.local.
13. dns IN A 192.168.0.80
14. router IN A 192.168.0.1
15. gw IN CNAME router
16.;
```

El archivo de la zona inversa

```
1. ;
2. ; BIND data file for zona inversa
3. ;
4. $TTL 604800
5. @ IN SOA dns.javi.local. root.javi.local. (
6.           2          ; Serial
7.           604800      ; Refresh
8.           86400       ; Retry
9.           2419200     ; Expire
10.          604800 )   ; Negative Cache TTL
11. ;
12.      IN NS dns.javi.local.
13. 0 IN PTR dns.javi.local.
```



14.

Comprobamos de nuevo con los siguientes comandos

1. `named-checkconf`
2. `named-checkzone javi.local. db.javi.local`
3. `named-checkzone 0.168.192.in-addr.arpa db.0.168.192`

Y si está todo ok reiniciamos el servicio con

1. `sudo systemctl restart bind9`
2. `sudo systemctl status bind9`

```
administrador@equipo:/etc/bind$ sudo systemctl status bind9
● named.service - BIND Domain Name Server
   Loaded: loaded (/lib/systemd/system/named.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2025-02-11 11:42:59 CET; 7s ago
     Docs: man:named(8)
 Process: 5090 ExecStart=/usr/sbin/named $OPTIONS (code=exited, status=0/SUCCESS)
 Main PID: 5091 (named)
   Tasks: 6 (limit: 2226)
    Memory: 5.5M
      CPU: 18ms
     CGroup: /system.slice/named.service
             └─5091 /usr/sbin/named -u bind -4

feb 11 11:42:59 equipo named[5091]: managed-keys-zone: loaded serial 4
feb 11 11:42:59 equipo named[5091]: zone 0.in-addr.arpa/IN: loaded serial 1
feb 11 11:42:59 equipo named[5091]: zone 1.168.192.in-addr.arpa/IN: loaded serial 2
feb 11 11:42:59 equipo named[5091]: zone localhost/IN: loaded serial 2
feb 11 11:42:59 equipo named[5091]: zone 127.in-addr.arpa/IN: loaded serial 1
feb 11 11:42:59 equipo named[5091]: zone 255.in-addr.arpa/IN: loaded serial 1
feb 11 11:42:59 equipo named[5091]: zone javi.local/IN: loaded serial 2
feb 11 11:42:59 equipo named[5091]: all zones loaded
feb 11 11:42:59 equipo named[5091]: running
feb 11 11:42:59 equipo systemd[1]: Started BIND Domain Name Server.
administrador@equipo:/etc/bind$
```