

## Sumario

UT 03: Servicio DNS - Prácticas.....	2
1 Actividades de introducción.....	2
2 Configuración de servicio DNS.....	8
2.1 Configuración de cliente DNS.....	9
2.2 Configuración de servidor DNS.....	12
2.2.1 Instalación del paquete.....	12
2.2.2 Servicios y puertos.....	12
2.2.3 Ficheros de configuración de zona.....	12
2.2.4 Ficheros de opciones globales.....	14
2.2.5 Arrancar el servicio.....	15
2.2.6 Confirmar que el servidor también es cliente.....	15
2.3 Configuración de resolución inversa.....	16

# UT 03: Servicio DNS - Prácticas

## 1 Actividades de introducción

Realizar estas actividades básicas:

1. Ejecutar el comando "nslookup <url>", donde <url> es alguna dirección web conocida. Observar la salida del comando. ¿Qué servidor DNS está respondiendo? ¿Qué direcciones IP se asocian con la URL?

```
$ nslookup www.amazon.com
Server:          127.0.0.53
Address:        127.0.0.53#53
```

```
Non-authoritative answer:
www.amazon.com canonical name = www.cdn.amazon.com.
www.cdn.amazon.com canonical name = d3ag4hukkh62yn.cloudfront.net.
Name: d3ag4hukkh62yn.cloudfront.net
Address: 13.224.110.250
```

```
$ nslookup www.iesclaradelrey.es
Server:          127.0.0.53
Address:        127.0.0.53#53
```

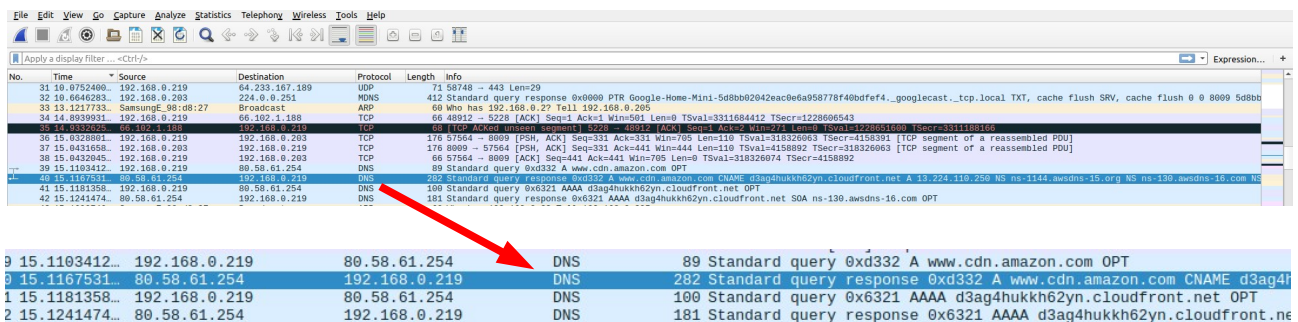
```
Non-authoritative answer:
Name: www.iesclaradelrey.es
Address: 217.160.0.86
Name: www.iesclaradelrey.es
Address: 2001:8d8:100f:f000::2b5
```

NOTA: Sobre la dirección stub-resolver 127.0.0.53:

(<https://unix.stackexchange.com/questions/500536/what-are-dns-server-resolver-and-stub-resolver>)

```
$ dig @8.8.8.8 www.madrid.org +trace
```

2. Realizar el ejercicio anterior capturando con wireshark los paquetes transmitidos por la red. ¿Qué protocolos y puertos se están usando?



The screenshot shows a Wireshark capture of network traffic. A red arrow points to a DNS query packet (No. 89) from source 192.168.0.219 to destination 80.58.61.254. The packet details show it is a Standard query for 'www.cdn.amazon.com' with type OPT. Below it, the response (No. 181) is shown, returning the IP address 13.224.110.250 for the CNAME 'd3ag4hukkh62yn.cloudfront.net'.

No.	Time	Source	Destination	Protocol	Length	Info
31	15.0752490	192.168.0.219	94.233.197.189	UDP	74	58748 → 443 Len=29
32	15.0646283	192.168.0.203	224.0.0.251	MDNS	412	Standard query response 0x0000 PTR Google-Home-Mini-5d8bb02842eac0e6a958778f40bdfef4...googlecast_tcp.local TXT, cache flush SRV, cache flush 0 8009 5d8bb
33	15.1217733	SamsungE_98:08:27	Broadcast	ARP	60	Who has 192.168.0.27 Tell 192.168.0.203
34	15.0839931	192.168.0.219	66.102.1.188	TCP	60	48912 → 5228 [ACK] Seq=1 Ack=1 Win=0 Len=0 TSval=3311684412 TSecr=1228606543
35	15.0532425	66.102.1.188	192.168.0.219	TCP	60	TCP Acked (reset segment) 5228 → 48912 [ACK] Seq=1 Ack=2 Win=271 Len=0 TSval=1228606543 TSecr=3311684412
36	15.0228031	192.168.0.219	192.168.0.203	TCP	176	57564 → 8009 [PSH, ACK] Seq=331 Ack=332 Win=105 Len=110 TSval=318326063 TSecr=1153921 [TCP segment of a reassembled PDU]
37	15.0431658	192.168.0.203	192.168.0.219	TCP	176	8009 → 57564 [PSH, ACK] Seq=331 Ack=441 Min=444 Len=110 TSval=4158892 TSecr=318326063 [TCP segment of a reassembled PDU]
38	15.0432045	192.168.0.219	192.168.0.203	TCP	66	57564 → 8009 [ACK] Seq=441 Ack=705 Len=0 TSval=318326074 TSecr=4158892
39	15.1193412	192.168.0.219	80.58.61.254	DNS	89	Standard query 0xd332 A www.cdn.amazon.com OPT
40	15.1167531	80.58.61.254	192.168.0.219	DNS	282	Standard query response 0xd332 A www.cdn.amazon.com CNAME d3ag4hukkh62yn.cloudfront.net A 13.224.110.250 NS ns-1144.awsdns-16.org NS ns-130.awsdns-16.com NS
41	15.1161258	192.168.0.219	80.58.61.254	DNS	100	Standard query 0x6321 AAAA d3ag4hukkh62yn.cloudfront.net OPT
42	15.1241474	80.58.61.254	192.168.0.219	DNS	181	Standard query response 0x6321 AAAA d3ag4hukkh62yn.cloudfront.net SOA ns-130.awsdns-16.com OPT

9	15.1103412...	192.168.0.219	80.58.61.254	DNS	89	Standard query 0xd332 A www.cdn.amazon.com OPT
9	15.1167531...	80.58.61.254	192.168.0.219	DNS	282	Standard query response 0xd332 A www.cdn.amazon.com CNAME d3ag4h
1	15.1181358...	192.168.0.219	80.58.61.254	DNS	100	Standard query 0x6321 AAAA d3ag4hukkh62yn.cloudfront.net OPT
2	15.1241474...	80.58.61.254	192.168.0.219	DNS	181	Standard query response 0x6321 AAAA d3ag4hukkh62yn.cloudfront.net

Para ver los puertos utilizados y más información de cada trama, miramos la parte inferior de la pantalla de wireshark:

```

▶ Frame 40: 282 bytes on wire (2256 bits), 282 bytes captured (2256 bits) on interface 0
▶ Ethernet II, Src: AskeyCom_91:81:f7 (d4:7b:b0:91:81:f7), Dst: IntelCor_f5:13:ce (d4:25:8b:f5:13:ce)
▶ Internet Protocol Version 4, Src: 80.58.61.254, Dst: 192.168.0.219
▶ User Datagram Protocol, Src Port: 53, Dst Port: 42833
▼ Domain Name System (response)
  Transaction ID: 0xd332
  ▶ Flags: 0x8180 Standard query response, No error
  Questions: 1
  Answer RRs: 2
  Authority RRs: 4
  Additional RRs: 1
  ▶ Queries
  ▶ Answers
  ▶ Authoritative nameservers
  ▶ Additional records
  [Request In: 39]
  [Time: 0.0006411894 seconds]

```

```

0000  d4 25 8b f5 13 ce d4 7b b0 91 81 f7 08 00 45 00  %.....{ .....E.
0010  01 0c 11 49 00 00 f8 11 60 dc 50 3a 3d fe c0 a8  ...I.....`P:=...
0020  00 db 00 35 a7 51 00 f8 d7 cf d3 32 81 80 00 01  ...5.Q.....2...
0030  00 02 00 04 00 01 03 77 77 77 03 63 64 6e 06 61  .....w ww-cdn-a
0040  6d 61 7a 6f 6e 03 63 6f 6d 00 00 01 00 01 c0 0c  mazon.co m.....
0050  00 05 00 01 00 00 00 29 00 1f 0e 64 33 61 67 34  .....) ...d3ag4
0060  68 75 6b 6b 68 36 32 79 6e 0a 63 6c 6f 75 64 66  hukkh62y n-cloudf
0070  72 6f 6e 74 03 6e 65 74 00 c0 30 00 01 00 01 00  ront.net ..0.....
0080  00 00 35 00 04 0d e0 6e fa c0 30 00 02 00 01 00  ..5.....0.....
0090  02 a2 f9 00 17 07 6e 73 2d 31 31 34 34 09 61 77  .....ns -1144 aw
00a0  73 64 6e 73 2d 31 35 03 6f 72 67 00 c0 30 00 02  sdns-15- org..0..
00b0  00 01 00 02 a2 f9 00 13 06 6e 73 2d 31 33 30 09  ..... ns-130
00c0  61 77 73 64 6e 73 2d 31 36 c0 1b c0 30 00 02 00  awsdns-1 6...0...
00d0  01 00 02 a2 f9 00 19 07 6e 73 2d 32 30 32 31 09  ..... ns-2021
00e0  61 77 73 64 6e 73 2d 36 30 02 63 6f 02 75 6b 00  awsdns-6 0.co.uk
00f0  c0 30 00 02 00 01 00 02 a2 f9 00 13 06 6e 73 2d  -0.....-ns-

```

The screenshot shows the Wireshark interface with the following details:

- Packet List:** Shows a list of packets. Packet 78 is highlighted, corresponding to the details pane below.
- Packet Details:**
  - Frame 78: 104 bytes on wire (832 bits), 104 bytes captured (832 bits) on interface 0
  - Ethernet II, Src: AskeyCom\_91:81:f7 (d4:7b:b0:91:81:f7), Dst: IntelCor\_f5:13:ce (d4:25:8b:f5:13:ce)
  - Internet Protocol Version 4, Src: 80.58.61.254, Dst: 192.168.0.219
  - User Datagram Protocol, Src Port: 53, Dst Port: 58055
  - Domain Name System (response)
    - Transaction ID: 0x37a4
    - Flags: 0x8180 Standard query response, No error
    - Questions: 1
    - Answer RRs: 1
    - Authority RRs: 0
    - Additional RRs: 1
    - Queries
      - iesclaradelrey.es: type A, class IN
    - Answers
      - iesclaradelrey.es: type A, class IN, addr 217.160.0.86
    - Additional records
      - <Root>: type OPT

Red arrows in the image point to the 'Answers' section, specifically to the record for 'iesclaradelrey.es: type A, class IN, addr 217.160.0.86', and to the 'Additional records' section.

- Investigar las características y propietarios de los dominios oficialmente registrados, que pueden encontrarse en la web <http://www.iana.org/domains/root/db/> ¿Qué empresa o persona es responsable de administrar el dominio ".es"?

<https://www.iana.org/domains/root/db/es.html>

- Elegir un posible nombre de dominio para una nueva empresa. Acceder a la web de algún registrador de nombres de dominio y hacer la simulación de la contratación del dominio. ¿Qué precios y características tienen los servicios investigados?

*Mi clase.com ... Buscar en Hostalia.es...*

- Ejecutar las siguientes consultas DNS y explicar los resultados:

```
$ nslookup www.google.es 8.8.8.8  
$ nslookup www.google.es ns1.google.com
```

*En la primera búsqueda obtenemos la dirección del servidor de dominio [www.google.es](http://www.google.es) a través del servidor DNS 8.8.8.8.*

*En la segunda buscamos la misma dirección a través del servidor DNS 216.239.32.10. Ambos responden con la misma IP en versión 6, diferente en v4.*

*(Ver <https://tecadmin.net/authoritative-non-authoritative-dns-server/>)*

```
jose@Audax:~$ nslookup google.com 8.8.8.8  
Server:      8.8.8.8  
Address:    8.8.8.8#53
```

```
Non-authoritative answer:  
Name: google.com  
Address: 172.217.16.238  
Name: google.com  
Address: 2a00:1450:4003:80b::200e
```

```
jose@Audax:~$ nslookup google.com ns1.google.com  
Server:      ns1.google.com  
Address:    216.239.32.10#53
```

```
Name: google.com  
Address: 172.217.17.14  
Name: google.com  
Address: 2a00:1450:4003:802::200e
```

6. Ejecutar el siguiente comando y observar la salida:

```
$ dig @8.8.8.8 www.madrid.org +trace
```

*dig es un comando que interroga a los servidores DNS a través de internet, mostrando la información que se les requiere. En este caso, sobre el dominio [www.madrid.org](http://www.madrid.org)*

*El comando investigará la información disponible sobre [www.madrid.org](http://www.madrid.org) en todos los servidores de la jerarquía DNS a partir del servidor DNS del dominio desde el que estamos buscando la información.*

```
; <<>> DiG 9.11.5-P4-5.lubuntu2-Ubuntu <<>> @8.8.8.8 www.madrid.org +trace
; (1 server found)
;; global options: +cmd
.          15137  IN      NS      g.root-servers.net.
.          15137  IN      NS      d.root-servers.net.
.          15137  IN      NS      f.root-servers.net.
.          15137  IN      NS      c.root-servers.net.
.          15137  IN      NS      a.root-servers.net.
.          15137  IN      NS      l.root-servers.net.
.          15137  IN      NS      i.root-servers.net.
.          15137  IN      NS      j.root-servers.net.
.          15137  IN      NS      k.root-servers.net.
.          15137  IN      NS      b.root-servers.net.
.          15137  IN      NS      m.root-servers.net.
.          15137  IN      NS      h.root-servers.net.
.          15137  IN      NS      e.root-servers.net.
.          15137  IN      RRSIG   NS 8 0 518400 20191110170000 20191028160000 22545 .
M1huSfVid4zXELkGjH34yibkWi1vX0z2uGActG+4JuJg1aAuFlvS/7mc
6+3JdMTUaHocL3sy092+QoIluw0mn0PPiPdE0YiT8Svk0l0m6FjDPLdg
WboquL0bbw/TXAdhP43XFyVew4/hIy2RQL5G3JLk0gsbM08PfQXZovbJ
eL8pdNyGr2/6WrmXeWsuAs9MjKNSMLgLyDIE965pow997Paurzlb0r0H
LqcSWTfovJxXhmhtGqDH/j/spDF7cr7tVvHYc9hyLho0qzIgHN0qcMfH
+GHdLPdJyoRyjAwyxkdt0X+VU5v0NnDBxu4X5YRrInwscwp/FnPf EkwNFg==
;; Received 525 bytes from 8.8.8.8#53(8.8.8.8) in 7 ms

org.       172800  IN      NS      d0.org.afilias-nst.org.
org.       172800  IN      NS      a2.org.afilias-nst.info.
org.       172800  IN      NS      b0.org.afilias-nst.org.
org.       172800  IN      NS      c0.org.afilias-nst.info.
org.       172800  IN      NS      a0.org.afilias-nst.info.
org.       172800  IN      NS      b2.org.afilias-nst.org.
org.       86400   IN      DS      9795 7 1 364DFAB3DAF254CAB477B5675B10766DDAA24982
org.       86400   IN      DS      9795
3922B31B6F3A4EA92B19EB7B52120F031FD8E05FF0B03BAFCF9F891B FE7FF8E5
org.       86400   IN      RRSIG   DS 8 1 86400 20191111170000 20191029160000 22545 .
YmUjJ0D2W3N107Pd5rYKK6i0Y+gbw7kDKhzkLSKRmk6CSeJGwU+NrmL
eVmYXHJd3V3x4bmt9VrSsrpHpZgaJY7EZdmJCIuCGaBbKcPH2mPstKAX
U/uFfza2YAqpHhyF8rvk4xPLJK6PMse/9M7ivPI4h3dy5xyXoqNUmo2
E8bxVe1L0oyzjQp4U5Jn/LH1JeKrEpzWkaSnkXDenjaIcVhED2cgFnQ/
cgk519hokYS0FCr1LZ4ETZucZ+Vkl7SMWYoX6VnlpNLIuEU6ZaGh800n
EWyZbRmwZZ26Tkfa+AeaMxEzZFo6t7SABkGPW/uTLMKhNoLh6xuccEYQ MTZKxg==
;; Received 816 bytes from 202.12.27.33#53(m.root-servers.net) in 27 ms

madrid.org. 86400   IN      NS      icmcom.madrid.org.
madrid.org. 86400   IN      NS      olimpia.madrid.org.
h9p7u7tr2u91d0v0ljs9llgidnp90u3h.org. 86400 IN NSEC3 1 1 1 D399EAAB H9PAES2EQ3K44BAR1F3TIU00J45719RJ
NS SOA RRSIG DNSKEY NSEC3PARAM
h9p7u7tr2u91d0v0ljs9llgidnp90u3h.org. 86400 IN RRSIG NSEC3 7 2 86400 20191119184616 20191029174616
36752 org. nrG0w7rJJacivxd0fKH3opb5NWe7PDhAjb0XTVZUfbod1oh0psuA8qsv
```

```
ci22T9M+2bhUIdjGFLCB3WHv8rEtc77xhI2vtUSGZFdmUMDu6aYu0C2k
FL4LNaBWqZ2IYnnHU9w/5/dX1bhGeEAGFCHXDb/XUZB0g0QyNySo1cP w68=
49mid4kgm2ag3ed7rk4dt2kigumr5at3.org. 86400 IN NSEC3 1 1 1 D399EAAB 49N00PMB9FOGFDm63SJ5QSC5EE9MRP54
A RRSIG
49mid4kgm2ag3ed7rk4dt2kigumr5at3.org. 86400 IN RRSIG NSEC3 7 2 86400 20191115152753 20191025142753
36752 org. ErKCe0I8cSHBWNVA5i/1f/C6ySn6V81c2jNBGjEhR2pfGsoqsJUaXVcq
w+jVldsAFZXCAGb7oVeYmtU7c0wY3EQUMrWhk0yNh0vjoogJq06/nQZ/
QCA3p72PPwh8pR5zyXMYcgpdzBduUj1JTkFMYcEH+Kf6bhWYR71t3RsJ WxI=
;; Received 611 bytes from 199.249.120.1#53(b2.org.afilias-nst.org) in 36 ms

www.madrid.org. 1800 IN CNAME www.madrid.org.c.footprint.net.
;; Received 87 bytes from 213.0.53.140#53(olimpia.madrid.org) in 8 ms
```

7. Ejecutar los siguientes comandos para obtener los registros DNS de una web conocida:

```
$ dig @8.8.8.8 madrid.org SOA
$ dig @8.8.8.8 madrid.org NS
$ dig @8.8.8.8 madrid.org A
```

```
jose@Audax:~$ dig @8.8.8.8 madrid.org SOA
```

```
<<>> DiG 9.11.5-P4-5.lubuntu2-Ubuntu <<>> @8.8.8.8 madrid.org SOA
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->HEADER<<- opcode: QUERY, status: NOERROR, id: 44594
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;madrid.org. IN SOA

;; ANSWER SECTION:
madrid.org. 3599 IN SOA ddi-gm.madrid.org. mail.icmcom.madrid.org.
2013031078 43200 7200 2592000 172800

;; Query time: 50 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: mar oct 29 19:58:31 CET 2019
;; MSG SIZE rcvd: 94
```

```
jose@Audax:~$ dig @8.8.8.8 madrid.org NS
```

```
<<>> DiG 9.11.5-P4-5.lubuntu2-Ubuntu <<>> @8.8.8.8 madrid.org NS
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->HEADER<<- opcode: QUERY, status: NOERROR, id: 38103
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;madrid.org. IN NS

;; ANSWER SECTION:
madrid.org. 3398 IN NS icmcom.madrid.org.
madrid.org. 3398 IN NS olimpia.madrid.org.
```

```
;; Query time: 41 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: mar oct 29 19:58:40 CET 2019
;; MSG SIZE rcvd: 82
```

```
jose@Audax:~$ dig @8.8.8.8 madrid.org A
```

```
; <<> DiG 9.11.5-P4-5.1ubuntu2-Ubuntu <<> @8.8.8.8 madrid.org A
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->HEADER<<- opcode: QUERY, status: NOERROR, id: 26106
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1
```

```
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;madrid.org. IN A
```

```
;; AUTHORITY SECTION:
madrid.org. 1608 IN SOA ddi-gm.madrid.org. mail.icmcom.madrid.org.
2013031078 43200 7200 2592000 172800
```

```
;; Query time: 7 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: mar oct 29 19:58:42 CET 2019
;; MSG SIZE rcvd: 94
```

8. En un sistema Windows, consultar el contenido de la cache DNS mediante el comando:

```
> ipconfig /displaydns
```

## 2 Configuración de servicio DNS

Para la realización de actividades relacionadas con el servicio DNS, utilizaremos Virtualbox, configurando una máquina virtual como servidor y dos máquinas virtuales como clientes, de acuerdo con las siguientes características:

- Para el grupo de trabajo XX
  - Servidor Ubuntu 18.04 IP 192.168.1XX.254 / 24 (hostname servidorXX)
  - Cliente Ubuntu 18.04: IP 192.168.1XX.1 / 24 (hostname ubuntuXX)
  - (opcional) Cliente Windows: IP 192.168.1XX.2 / 24 (hostname winXX)
  - Todas las máquinas virtuales tendrán además una interfaz de red en modo NAT.
- Usaremos el dominio ficticio miclase.local
- Para las capturas realizadas en este documento usaremos:
  - Servidor Ubuntu 18.04 IP 192.168.100.254 / 24 (hostname servidordns)
  - Cliente Ubuntu 18.04 IP 192.168.100.1 / 24 (hostname clientedns)

Configuraciones de red iniciales: (solo con IP fijas y sin puerta de enlace)

```
jose@servidordns:~$ ip a
(...)
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
   link/ether 08:00:27:b6:ae:24 brd ff:ff:ff:ff:ff:ff
   inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
       valid_lft 84407sec preferred_lft 84407sec
   inet6 fe80::ca58:e691:1ea6:5d95/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
   link/ether 08:00:27:9a:bb:aa brd ff:ff:ff:ff:ff:ff
   inet 192.164.100.254/24 brd 192.164.100.255 scope global noprefixroute enp0s8
       valid_lft forever preferred_lft forever
   inet6 fe80::414d:9956:48a0:dda/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

```
jose@clientedns:~$ ip a
(...)
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
   link/ether 08:00:27:69:33:74 brd ff:ff:ff:ff:ff:ff
   inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
       valid_lft 86264sec preferred_lft 86264sec
   inet6 fe80::2af2:1812:908d:5c32/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
   link/ether 08:00:27:d5:4e:0f brd ff:ff:ff:ff:ff:ff
   inet 192.168.100.1/24 brd 192.168.100.255 scope global noprefixroute enp0s8
       valid_lft forever preferred_lft forever
   inet6 fe80::395a:2664:621b:e42a/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

```
jose@clientedns:~$ ping 192.168.100.254
PING 192.168.100.254 (192.168.100.254) 56(84) bytes of data:
64 bytes from 192.168.100.254: icmp_seq=1 ttl=64 time=0.925 ms
^C
--- 192.168.100.254 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1000ms
rtt min/avg/max/mdev = 0.849/0.887/0.925/0.038 ms
```



## 2.1 Configuración de cliente DNS

Antes de configurar el servicio DNS por la red interna, verificamos que nuestra máquina está conectada a la red y a los servicios DNS de internet:

```
jose@clientedns:~$ ping www.google.es
PING www.google.es (172.217.16.227) 56(84) bytes of data.
64 bytes from mad08s04-in-f3.1e100.net (172.217.16.227): icmp_seq=1 ttl=63
time=4.78 ms
64 bytes from mad08s04-in-f3.1e100.net (172.217.16.227): icmp_seq=2 ttl=63
time=4.80 ms
^C
--- www.google.es ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 4.785/4.794/4.804/0.069 ms
jose@clientedns:~$ nslookup www.google.es
Server:      127.0.0.53
Address:    127.0.0.53#53
```

```
Non-authoritative answer:
Name: www.google.es
Address: 172.217.16.227
Name: www.google.es
Address: 2a00:1450:4003:804::2003
```

Sin embargo, todavía no hemos configurado nuestro dominio **miclase.local**

```
jose@clientedns:~$ nslookup miclase.local
Server:      127.0.0.53
Address:    127.0.0.53#53
** server can't find miclase.local: NXDOMAIN
```

Nuestro cliente es capaz de conectar consigo mismo a través del nombre, pero no reconoce el nombre de otros equipos que están en su red, ni el del dominio:

```
jose@clientedns:~$ ping clientedns
PING clientedns (127.0.1.1) 56(84) bytes of data.
64 bytes from clientedns (127.0.1.1): icmp_seq=1 ttl=64 time=0.050 ms
^C
--- clientedns ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1011ms
rtt min/avg/max/mdev = 0.050/0.060/0.070/0.010 ms
jose@clientedns:~$ ping servidorDNS
ping: servidorDNS: Name or service not known
jose@clientedns:~$ ping clientedns.miclase.local
ping: clientedns.miclase.local: Name or service not known
```

Vamos a realizar varias aproximaciones a la solución:

### 1. Resolución mediante fichero de hosts

La primera forma de trabajar con nombres de equipos es la utilización de un archivo de

hosts.

En **Windows** tendríamos que editar el fichero `%SYSTEMROOT%\system32\drivers\etc\hosts`, añadiendo una entrada para `servidordns`, `servidordns.miclase.local` y `clientesdns.miclase.local`, indicando las IPs correspondientes.

En **Linux** tendríamos que editar el fichero `/etc/hosts`. Quedaría así:

```
jose@clientesdns:/etc$ cat hosts
127.0.0.1 localhost
127.0.1.1 clientesdns
# Equipos de la red local
192.168.100.1 clientesdns
192.168.100.1 clientesdns.miclase.local
192.168.100.254 servidordns
192.168.100.254 servidordns.miclase.local
```

Y ahora sí tendríamos conectividad usando los nombres de los equipos:

```
jose@clientesdns:/etc$ ping servidordns
PING servidordns (192.168.100.254) 56(84) bytes of data.
64 bytes from servidordns (192.168.100.254): icmp_seq=1 ttl=64 time=0.679 ms
^C
--- servidordns ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 0.679/0.784/0.889/0.105 ms
jose@clientesdns:/etc$ ping servidordns.miclase.local
PING servidordns.miclase.local (192.168.100.254) 56(84) bytes of data.
64 bytes from servidordns (192.168.100.254): icmp_seq=2 ttl=64 time=0.823 ms
^C
```

(Pero NO hemos utilizado DNS)

## 2. Configuración de cliente DNS

Revertimos el fichero `/etc/host` para cambiar la configuración a DNS. En estos momentos, el cliente no es capaz de encontrar al servidor mediante nombre (aunque sí mediante IP):

```
jose@clientesdns:/etc$ ping servidordns
ping: servidordns: Name or service not known
jose@clientesdns:/etc$ ping 192.168.100.254
PING 192.168.100.254 (192.168.100.254) 56(84) bytes of data.
64 bytes from 192.168.100.254: icmp_seq=1 ttl=64 time=0.795 ms
```

Para configurar el cliente DNS y tener capacidad de resolución por nombres de máquinas y de dominios, tenemos que modificar/revisar dos ficheros de configuración:

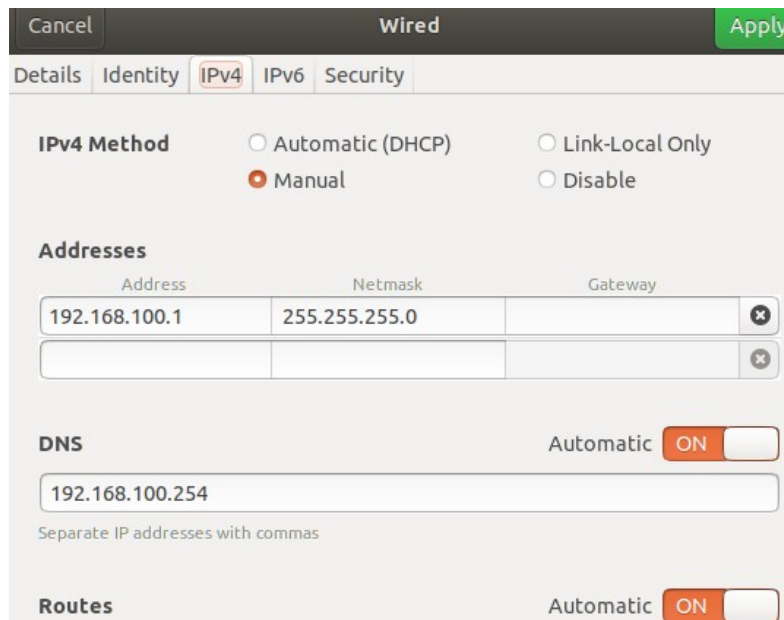
`/etc/nsswitch.conf` - Debemos asegurarnos de que está activa la opción `dns`:

```
hosts: files mdns4_minimal [NOTFOUND=return] dns myhostname
```

`/etc/resolv.conf` - Tradicionalmente, se añade al servidor en una línea "nameserver"

```
jose@clientedns:/etc$ cat resolv.conf  
(...)  
nameserver 192.168.100.254
```

Sin embargo, esta forma de configurar DNS ya no es válida en Ubuntu 18.04. Para hacerlo más fácil podemos recurrir a la interfaz gráfica:



O, mejor aún, en caso de que nuestra red esté configurada a través de netplan, podemos modificar el fichero correspondiente para que quede así:

```
/etc/netplan$ cat 01-network-manager-all.yaml  
# Let NetworkManager manage all devices on this system  
network:  
  version: 2  
  renderer: networkd  
  ethernets:  
    enp0s3:  
      dhcp4: yes  
    enp0s8:  
      dhcp4: no  
      dhcp6: no  
      addresses: [192.168.100.254/24, ]  
      gateway4: 192.168.100.100  
      nameservers:  
        search: [miclase.local]  
        addresses: [192.168.100.100]
```

Y aplicar la configuración con:

```
/etc/netplan$ sudo netplan apply
```

Pero para que funcione, el servidor debe tener configurado DNS (y aún no lo tenemos).

## 2.2 Configuración de servidor DNS

### 2.2.1 Instalación del paquete

Usaremos el servidor DNS Bind, que debemos instalar en la máquina `servidordns`:

```
jose@servidordns:/etc$ sudo apt-get install bind9
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  bind9utils net-tools python3-ply
Suggested packages:
  bind9-doc resolvconf python-ply-doc
The following NEW packages will be installed:
  bind9 bind9utils net-tools python3-ply
0 upgraded, 4 newly installed, 0 to remove and 3 not upgraded.
Need to get 854 kB of archives.
After this operation, 4.353 kB of additional disk space will be used.
Do you want to continue? [Y/n]
(...)
```

### 2.2.2 Servicios y puertos

Aparece un nuevo demonio en ejecución, denominado "named":

```
jose@servidordns:/etc$ ps -ef | grep named
bind      4727      1  0 00:30 ?        00:00:00 /usr/sbin/named -f -u bind
jose      4799    1763  0 00:30 pts/0    00:00:00 grep --color=auto named
```

Vemos que ha abierto los puertos 53 TCP y UDP:

```
jose@servidordns:/etc$ netstat -ltun
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp    0      0 192.168.100.254:53     0.0.0.0:*                LISTEN
tcp    0      0 10.0.2.15:53          0.0.0.0:*                LISTEN
tcp    0      0 127.0.0.1:53          0.0.0.0:*                LISTEN
tcp    0      0 127.0.0.53:53         0.0.0.0:*                LISTEN
(...)
udp    0      0 192.168.100.254:53     0.0.0.0:*                *
udp    0      0 10.0.2.15:53          0.0.0.0:*                *
udp    0      0 127.0.0.1:53          0.0.0.0:*                *
udp    19200  0 127.0.0.53:53         0.0.0.0:*                *
```

### 2.2.3 Ficheros de configuración de zona

Los ficheros de configuración del servicio están en `/etc/bind`

```
jose@servidordns:/etc/bind$ ls -al
total 68
drwxr-sr-x  2 root bind  4096 oct 20 00:29 .
drwxr-xr-x 122 root root 12288 oct 20 00:29 ..
-rw-r--r--  1 root root  2761 ago 10 08:26 bind.keys
```

```
-rw-r--r-- 1 root root 237 mar 23 2018 db.0
-rw-r--r-- 1 root root 271 mar 23 2018 db.127
-rw-r--r-- 1 root root 237 mar 23 2018 db.255
-rw-r--r-- 1 root root 353 mar 23 2018 db.empty
-rw-r--r-- 1 root root 270 mar 23 2018 db.local
-rw-r--r-- 1 root root 3171 mar 23 2018 db.root
-rw-r--r-- 1 root bind 463 mar 23 2018 named.conf
-rw-r--r-- 1 root bind 490 mar 23 2018 named.conf.default-zones
-rw-r--r-- 1 root bind 165 mar 23 2018 named.conf.local
-rw-r--r-- 1 root bind 890 mar 23 2018 named.conf.options
-rw-r----- 1 bind bind 77 oct 20 00:29 rndc.key
-rw-r--r-- 1 root root 1317 mar 23 2018 zones.rfc1918
```

Los ficheros de configuración que debemos modificar son `named.conf.local` (registros de recursos de zonas) y `named.conf.options` (opciones locales)

Guardaremos una copia de estos dos ficheros de configuración y a continuación realizaremos los cambios oportunos:

1. Para que el servidor actúe como maestro y tenga autoridad sobre el dominio `miclase.local`, añadiremos un registro NS para `servidordns.miclase.local`
2. Daremos de alta el nombre de `clientedns.miclase.local` (registro A)
3. Configuraremos además varios alias (registros CNAME)
  - `ns1.miclase.local` – alias de `servidordns.miclase.local`
  - `www.miclase.local` – alias de `servidordns.miclase.local`
  - `ftp.miclase.local` – alias de `servidordns.miclase.local`
  - `mail.miclase.local` – alias de `clientedns.miclase.local`
4. El equipo `clientedns.miclase.local` actuará como servidor de correo del dominio (registro MX)
5. El tiempo en cache de las respuestas negativas será de 3 horas.

Con todas estas premisas, realizamos las operaciones indicadas editando los ficheros que se encuentran en `/etc/bind`:

```
jose@servidordns:/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";
//

//Zona de búsqueda directa
zone "miclase.local"{
    type master;
    file "/etc/bind/db.miclase.local";
};
```

```
jose@servidordns:/etc/bind$ cat db.miclase.local
; Fichero db.miclase.local
$TTL 1D
@      IN      SOA      miclase.local.  administrador.miclase.local. (
                                1          ; Serial
                                604800     ; Refresh
                                86400      ; Retry
                                2419200    ; Expire
                                10800 )    ; Negative Cache TTL (3 horas)
; Servidores DNS del dominio
                IN      NS      servidordns.miclase.local.
; Hosts
clientesdns IN      A        192.168.100.1
servidordns IN     A        192.168.100.254

; Alias
ns1      IN      CNAME     servidordns
www      IN      CNAME     servidordns
ftp      IN      CNAME     servidordns
mail     IN      CNAME     clientesdns

; Servidores de correo (MTA)
@      IN      MX      10      clientesdns
```

Tras editar los ficheros, comprobamos que no hay error de configuración:

```
jose@servidordns:/etc/bind$ sudo named-checkconf /etc/bind/named.conf
jose@servidordns:/etc/bind$ sudo named-checkzone miclase.local
/etc/bind/db.miclase.local
```

```
zone miclase.local/IN: loaded serial 1
OK
```

### 2.2.4 Ficheros de opciones globales

```
jose@servidordns:/etc/bind$ cat named.conf.options
options {
(...)
    listen-on {
        192.168.100.0/24;
    };

    listen-on-v6 { none; };
};
```

Antes de probar el servicio vamos a asegurarnos de que funcione con Ipv4 modificando este último fichero:

```
/etc/default$ cat named
#
```

```
# run resolvconf?  
RESOLVCONF=no
```

```
# startup options for the server  
OPTIONS="-u bind -4"
```

### 2.2.5 Confirmar que el servidor también es cliente

En este punto es importante confirmar que nuestro servidor, aparte de ser "servidor DNS", tiene que ser cliente DNS de sí mismo, para que pueda acceder al servicio de resolución de nombres (tal como se explicaba en el apartado 2.1 de este manual). Para ello nos aseguraremos de que tiene la configuración adecuada como cliente DNS, bien como parámetro en la configuración de red del sistema a través de menús, o bien a través de Netplan. En el caso de Netplan, nuestro archivo de configuración de red debe ser algo parecido a esto:

```
/etc/netplan$ cat 01-network-manager-all.yaml  
# Let NetworkManager manage all devices on this system  
network:  
  version: 2  
  renderer: networkd  
  ethernets:  
    enp0s3:  
      dhcp4: yes  
    enp0s8:  
      dhcp4: no  
      dhcp6: no  
      addresses: [192.168.100.254/24, ]  
      gateway4: 192.168.100.100  
      nameservers:  
        search: [miclase.local]  
        addresses: [192.168.100.254]
```

### 2.2.6 Arrancar el servicio

A continuación reiniciamos el servicio Bind9:

```
jose@servidordns:/etc/bind$ sudo systemctl restart bind9  
jose@servidordns:/etc/bind$ sudo systemctl status bind9  
● bind9.service - BIND Domain Name Server  
   Loaded: loaded (/lib/systemd/system/bind9.service; enabled; vendor preset: en  
   Active: active (running) since Sat 2018-10-20 01:29:03 CEST; 4s ago  
     Docs: man:named(8)  
  Process: 2662 ExecStop=/usr/sbin/rndc stop (code=exited, status=0/SUCCESS)  
 Main PID: 2665 (named)  
    Tasks: 4 (limit: 2321)  
   CGroup: /system.slice/bind9.service  
           └─2665 /usr/sbin/named -f -u bind
```

Y por último chequeamos que **el servidor está resolviendo nombres** para el dominio que hemos creado, `miclase.local`. A continuación vemos cómo funciona correctamente para el comando `nslookup` y también conecta con la máquina cliente a través de un ping al nombre que hemos configurado en DNS:

```
jose@servidordns:/etc$ nslookup servidordns
Non-authoritative answer:
Name: servidordns
Address: 127.0.1.1
```

```
jose@servidordns:/etc$ nslookup servidordns.miclase.local
Non-authoritative answer:
Name: servidordns.miclase.local
Address: 192.168.100.254
```

```
jose@servidordns:/etc$ nslookup mail.miclase.local
Non-authoritative answer:
mail.miclase.local canonical name = clientedns.miclase.local.
Name: clientedns.miclase.local
Address: 192.168.100.1
```

```
jose@servidordns:/etc$ ping clientedns.miclase.local
PING clientedns.miclase.local (192.168.100.1) 56(84) bytes of data:
64 bytes from 192.168.100.1 (192.168.100.1): icmp_seq=1 ttl=64 time=0.741 ms
```

Para verificar que hemos configurado bien el servidor DNS, podemos dar de alta un nuevo cliente. Por ejemplo, usamos una máquina Linux, que conectamos a nuestra red con la IP 192.168.100.5, la configuramos para que resuelva por DNS y que se conecte al Servidor DNS que hemos creado (IP 192.168.100.254). Este es el resultado:

```
$ ip a
(...)inet 192.168.100.5/24 brd 192.168.100.255
```

```
$ cat /etc/resolv.conf
# Generated by NetworkManager
nameserver 192.168.100.254
```

```
$ ping servidordns.miclase.local
PING servidordns.miclase.local (192.168.100.254) 56(84) bytes of data:
64 bytes from 192.168.100.254 (192.168.100.254): icmp_seq=1 ttl=64 time=0.799 ms
```

## 2.3 Configuración de resolución inversa

Para configurar correctamente la resolución inversa en nuestro dominio `miclase.local` debemos añadir los registros apropiados en los ficheros de configuración.

En `/etc/bind/named.conf.local` añadimos la configuración para búsqueda inversa:

```
(...)  
//Zona de búsqueda inversa  
zone "0.100.168.192.in-addr.arpa"{  
    type master;  
    file "/etc/bind/db.100.168.192";  
};
```

Y a continuación añadimos el fichero `/etc/bind/db.100.168.192`



```
;  
; BIND reverse data file for 192.168.100  
;  
$TTL 1D  
$ORIGIN 100.168.192.in-addr.arpa.  
@ IN SOA servidorDNS.miclase.local. root.miclase.local. (  
    1242760444  
    14400  
    3600  
    60480  
    604800  
)  
 IN NS servidorDNS.miclase.local.  
254.100.168.192.in-addr.arpa. IN PTR servidorDNS.miclase.local.  
1.100.168.192.in-addr.arpa. IN PTR clienteDNS.miclase.local.
```

Comprobamos el funcionamiento de la resolución inversa:

```
jose@servidorDNS:/etc/bind$ host 192.168.100.254  
254.100.168.192.in-addr.arpa domain name pointer servidorDNS.  
254.100.168.192.in-addr.arpa domain name pointer servidorDNS.local.
```