

Trabajo Práctico

## TPL 3 - Domain Name System

Fecha de entrega: 29/04/2021

Franco, Juan Martín 149.615

[juanmartin\\_franco@hotmail.com](mailto:juanmartin_franco@hotmail.com)

### 1. Describa el objetivo del sistema DNS, su estructura, elementos que intervienen y tipos de datos (Resource Records) típicos que se pueden consultar.

DNS (Domain Name System): Es un componente fundamental de Internet: asocia nombres de equipo con direcciones IP (y viceversa), lo que permite utilizar [www.google.com.ar](http://www.google.com.ar) en lugar de 216.58.202.35 (por ejemplo).

Objetivo de DNS: El objetivo principal es un espacio de nombres consistente que sea utilizado para referirse a los recursos. Para evitar problemas causados por extensiones especiales, los nombres no necesitarán contener identificadores de red, direcciones, rutas o información similar como parte del nombre.

#### Estructura de DNS:

- Los nombres de la base de datos DNS establecen una estructura lógica de árbol, conocida como espacio de nombres del dominio.
- Cada nodo o dominio del espacio de nombres del dominio tiene un nombre y puede contener subdominios.
- Los dominios y subdominios se agrupan en zonas para permitir la administración distribuida del espacio de nombres (las zonas se describen más adelante).
- El nombre del dominio identifica la posición del mismo en la jerarquía lógica del DNS respecto de su dominio principal, al separar cada rama del árbol con un punto.

#### Elementos que intervienen en DNS:

Un DNS tiene tres componentes principales:

- **EL ESPACIO DE NOMBRES DE DOMINIO y REGISTROS DE RECURSOS**, que son especificaciones para un árbol estructurado de espacio de nombres y datos asociados con los nombres. Conceptualmente, cada nodo e hijo del árbol del espacio de nombres del dominio nombra a un conjunto de información, y solicita operaciones para extraer tipos de información específicos de un conjunto en particular.
- **SERVIDORES DE NOMBRES**, que son programas de servidor donde se aloja la información de la estructura de un árbol de dominio y la establece.
- **RESOLUTORES**, que son programas que extraen información de servidores de nombres en respuesta a consultas de los clientes. Los resolutores deben tener acceso al menos a un servidor de nombres que pueda responder directamente la consulta, o proseguir con la consulta utilizando referencias a otros servidores de nombres.

## Tipos de datos (Resource Records) típicos que se pueden consultar en DNS:

Los tipos de registros DNS más importantes son los siguientes:

**A:** La mayor parte de resoluciones de nombres de dominio en Internet se producen mediante registros tipo A, que contienen una dirección IPv4 en su campo de datos.

**AAAA:** Los registros AAAA, también llamados quad-A, funcionan igual que los registros A salvo que, en lugar de usar una dirección IPv4, usan una dirección Ipv6.

**SOA:** SOA son las iniciales de Start of Authority. Los registros de este tipo contienen información sobre la zona que se organiza a través del archivo de zona (del servidor DNS) y, por ello, son especialmente importantes para la transferencia de zonas.

**CNAME:** Un registro CNAME (canonical name record) contiene un alias, es decir, un nombre alternativo para un dominio, y remite a otro registro A o AAAA ya existente.

**MX:** El nombre del registro MX es una abreviación de mail Exchange. Aquí se definen uno o varios servidores de correo electrónico que pertenezcan al dominio en cuestión.

**NS:** Un registro NS hace referencia al servidor de nombres de un archivo de zona y determina dónde recae la responsabilidad de una zona concreta. Es, por ello, un registro obligatorio en todo archivo de zona.

**2. Utilizando la herramienta dig (o nslookup ) realice consultas al servidor DNS indicado por el docente, (o desde su hogar al provisto por su ISP, o bien alguno de acceso público tal como 8.8.8.8 o 1.1.1.1) para obtener la siguiente información:**

**a. ¿Cuál es la dirección IP del host archivos.unlu.edu.ar ?**

Para realizar este ejercicio, simplemente me dirijo a la terminal y ejecuto el comando:

```
dig archivos.unlu.edu.ar
```

Obteniendo el siguiente resultado:

```
juanma@ubuntu:~$ dig archivos.unlu.edu.ar

; <<>> DiG 9.16.1-Ubuntu <<>> archivos.unlu.edu.ar
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 25018
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL:
1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;archivos.unlu.edu.ar.      IN      A

;; ANSWER SECTION:
archivos.unlu.edu.ar.      5       IN      A      170.210.96.201

;; Query time: 24 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: Sat Apr 24 12:00:34 PDT 2021
;; MSG SIZE rcvd: 65

juanma@ubuntu:~$
```

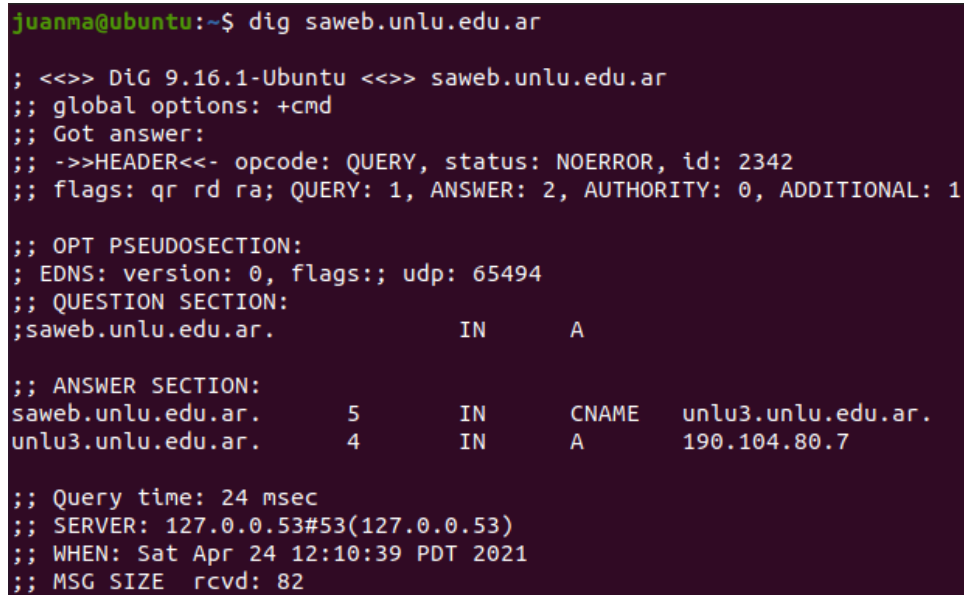
Por lo que la ip del host archivos.unlu.edu.ar es 170.210.96.201

**b. ¿Cuál es la dirección IP del host saweb.unlu.edu.ar? ¿Qué diferencia nota en la respuesta respecto al punto anterior?**

Para realizar este ejercicio repito el procedimiento del ejercicio anterior, cambiando el nombre del host por “saweb.unlu.edu.ar”

Ejecutando el comando:

```
dig saweb.unlu.edu.ar
```



```
juanma@ubuntu:~$ dig saweb.unlu.edu.ar

; <<>> DiG 9.16.1-Ubuntu <<>> saweb.unlu.edu.ar
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 2342
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;saweb.unlu.edu.ar.      IN      A

;; ANSWER SECTION:
saweb.unlu.edu.ar.      5       IN      CNAME   unlu3.unlu.edu.ar.
unlu3.unlu.edu.ar.      4       IN      A       190.104.80.7

;; Query time: 24 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: Sat Apr 24 12:10:39 PDT 2021
;; MSG SIZE rcvd: 82
```

Por lo que la dirección del host saweb.unlu.edu.ar es 190.104.80.7.

La diferencia con la respuesta anterior consta de que este último host fue accedido mediante su alias, el cual se especifica en un registro de tipo CNAME (el alias del host es saweb.unlu.edu.ar) y, el nombre original del host es:

unlu3.unlu.edu.ar

**c. ¿Cuáles son los intercambiadores de mail (mnemónico y dirección IP) del dominio unsa.edu.ar ?**

Los intercambiadores de mail son los registros MX.

Para ver esto utilizo el comando:

```
dig unsa.edu.ar -t MX
```

Obteniendo el siguiente resultado:

```

juanma@ubuntu:~$ dig unsa.edu.ar -t MX

; <<>> DiG 9.16.1-Ubuntu <<>> unsa.edu.ar -t MX
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 56889
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;unsa.edu.ar.                IN      MX

;; ANSWER SECTION:
unsa.edu.ar.                5       IN      MX      10 mx1.unsa.edu.ar.
unsa.edu.ar.                5       IN      MX      20 mx2.unsa.edu.ar.

;; Query time: 52 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: Sun Apr 25 00:40:14 PDT 2021
;; MSG SIZE  rcvd: 80

```

Los intercambiadores de mail son los siguientes:

**mx1.unsa.edu.ar**, cuya dirección ip es 170.210.206.18

Para obtener su dirección IP utilizo:

dig mx1.unsa.edu.ar -t A

```

juanma@ubuntu:~$ dig mx1.unsa.edu.ar -t A

; <<>> DiG 9.16.1-Ubuntu <<>> mx1.unsa.edu.ar -t A
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 44605
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;mx1.unsa.edu.ar.           IN      A

;; ANSWER SECTION:
mx1.unsa.edu.ar.           5       IN      A      170.210.206.18

;; Query time: 48 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: Sun Apr 25 00:43:03 PDT 2021
;; MSG SIZE  rcvd: 60

```

**mx2.unsa.edu.ar**, cuya dirección ip es 190.221.183.218

Para obtenerla utilizo:

dig mx2.unsa.edu.ar -t A

```

juanma@ubuntu:~$ dig mx2.unsa.edu.ar -t A

; <<>> DiG 9.16.1-Ubuntu <<>> mx2.unsa.edu.ar -t A
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 11234
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;mx2.unsa.edu.ar.                IN      A

;; ANSWER SECTION:
mx2.unsa.edu.ar.                5       IN      A      190.221.183.218

;; Query time: 48 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: Sun Apr 25 00:44:47 PDT 2021
;; MSG SIZE  rcvd: 60

```

**d. ¿Cuál es el nombre del host cuya dirección IP es 190.104.80.99 ?**

Para realizar esto, utilizo el comando

```
dig -x 190.104.80.99
```

El cual me permite realizar una búsqueda inversa, lo que quiere decir que, en lugar de buscar el nombre del host y obtener una ip, busco por ip y obtengo el nombre del host.

```

juanma@ubuntu:~$ dig -x 190.104.80.99

; <<>> DiG 9.16.1-Ubuntu <<>> -x 190.104.80.99
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 10801
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;99.80.104.190.in-addr.arpa.     IN      PTR

;; ANSWER SECTION:
99.80.104.190.in-addr.arpa. 5      IN      PTR      router4.unlu.edu.ar.

;; Query time: 56 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: Sat Apr 24 12:05:21 PDT 2021
;; MSG SIZE  rcvd: 88

```

El nombre del host asociado a dicha dirección ip (190.104.80.99) es router4.unlu.edu.ar

**e. ¿Cuáles son los servidores de nombres (mnemónicos y dirección IP) para el dominio icann.org ?**

Para buscar un servidor de nombre de un dominio específico, es necesario especificar esto en la consulta con un parámetro. En el caso de dig, debo utilizar el parámetro -t [TIPO].

Para este caso, necesito ejecutar el siguiente comando:

```
dig icann.org -t NS
```

Ya que el registro NS contiene los servidores de nombre.

El resultado obtenido es el siguiente:

```
juanma@ubuntu:~$ dig icann.org -t NS

; <<>> DiG 9.16.1-Ubuntu <<>> icann.org -t NS
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 25406
;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 1
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;icann.org.                IN      NS

;; ANSWER SECTION:
icann.org.      5      IN      NS      c.icann-servers.net.
icann.org.      5      IN      NS      ns.icann.org.
icann.org.      5      IN      NS      a.icann-servers.net.
icann.org.      5      IN      NS      b.icann-servers.net.

;; Query time: 168 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: Sat Apr 24 12:07:24 PDT 2021
;; MSG SIZE  rcvd: 120
```

Los servidores de nombre son los siguientes:

**c.icann-servers.net**, cuya ip es : 199.43.134.53

```
;; ANSWER SECTION:
c.icann-servers.net.      5      IN      A      199.43.134.53
```

Para obtenerla utilicé el comando:

```
dig c.icann-servers.net -t A
```

**ns.icann.org**, cuya ip es : 199.4.138.53

```
;; ANSWER SECTION:
ns.icann.org.      5      IN      A      199.4.138.53
```

Para obtenerla utilicé el comando:

```
dig ns.icann.org -t A
```

**a.icann-servers.net**, cuya ip es : 199.43.135.53

```
;; ANSWER SECTION:
a.icann-servers.net.      5      IN      A      199.43.135.53
```

Para obtenerla utilicé el comando:

```
dig a.icann-servers.net -t A
```

**b.icann-servers.net**, cuya ip es : 199.43.133.53

```
;; ANSWER SECTION:
b.icann-servers.net.      5      IN      A      199.43.133.53
```

Para obtenerla utilicé el comando:

```
dig b.icann-servers.net -t A
```

**f. ¿Cuál es la dirección IPv6 del host www.nic.ar?**

Para realizar esto utilizo el comando:

```
dig www.nic.ar -t AAAA
```

Ya que, como bien se detalla en el primer punto, los registros de AAAA asocian un host a una dirección IPV6.

```
juanma@ubuntu:~$ dig www.nic.ar -t AAAA

; <<>> DiG 9.16.1-Ubuntu <<>> www.nic.ar -t AAAA
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 31031
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;www.nic.ar.                IN      AAAA

;; ANSWER SECTION:
www.nic.ar.                  5      IN      AAAA    2801:140:5::10
```

Obteniendo como resultado que la dirección ipv6 del host www.nic.ar es: 2801:140:5::10

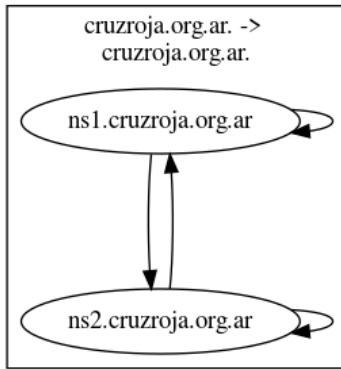
**3. Utilice la herramienta DNS BAJAJ disponible en <http://www.zonecut.net/dns/> para obtener información en forma de grafo acerca del dominio cruzroja.org.ar . ¿Cuáles son los servidores (nombre y dirección IP) para dicho dominio?**

Los servidores para dicho dominio son los siguientes:

ns1.cruzroja.org.ar

ns2.cruzroja.org.ar

El resultado fue provisto por la herramienta DNS BAJAJ.



Ahora, para averiguar la dirección IP de ambos hosts utilizo los siguientes comandos:

```
dig ns1.cruzroja.org.ar -t A
```

Y obtengo la dirección ip: 66.113.160.206

```
juanma@ubuntu:~$ dig ns1.cruzroja.org.ar -t A

; <<>> DiG 9.16.1-Ubuntu <<>> ns1.cruzroja.org.ar -t A
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 39873
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;ns1.cruzroja.org.ar.      IN      A

;; ANSWER SECTION:
ns1.cruzroja.org.ar.      5       IN      A      66.113.160.206
```

Por último, utilizo el comando:

```
dig ns2.cruzroja.org.ar -t A
```

Y obtengo la dirección ip: 66.113.160.206

```
juanma@ubuntu:~$ dig ns2.cruzroja.org.ar -t A

; <<>> DiG 9.16.1-Ubuntu <<>> ns2.cruzroja.org.ar -t A
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 25358
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;ns2.cruzroja.org.ar.      IN      A

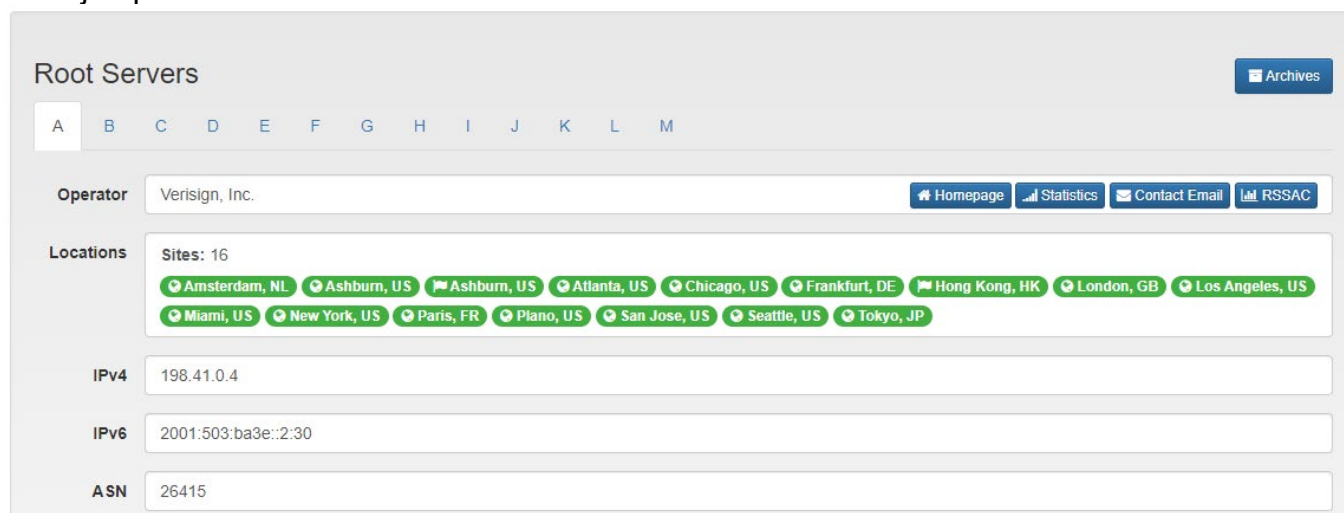
;; ANSWER SECTION:
ns2.cruzroja.org.ar.      5       IN      A      66.113.160.206
```



#### 4. ¿En dónde se encuentra la copia más cercana de un servidor dns raíz? ¿Cuál es el nombre del servidor replicado (o servidores)?

Los servidores raíz originalmente eran 13 (había 13 copias en total), y se identificaban con letras. Obviamente esto no era suficiente con 13 copias en el mundo para abarcar todas las consultas, por lo que cada root server tiene sus copias.

Por ejemplo:



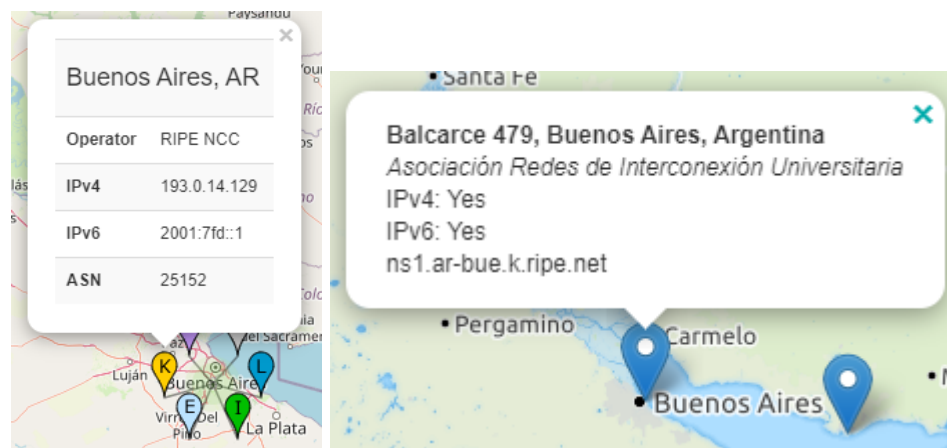
The screenshot shows the 'Root Servers' website. At the top, there's a navigation bar with letters A through M. Below it, the 'Operator' is listed as 'Verisign, Inc.' with links to 'Homepage', 'Statistics', 'Contact Email', and 'RSSAC'. The 'Locations' section shows 'Sites: 16' with a list of locations: Amsterdam, NL; Ashburn, US; Ashburn, US; Atlanta, US; Chicago, US; Frankfurt, DE; Hong Kong, HK; London, GB; Los Angeles, US; Miami, US; New York, US; Paris, FR; Plano, US; San Jose, US; Seattle, US; and Tokyo, JP. Below the locations, the IPv4 address is 198.41.0.4, the IPv6 address is 2001:503:ba3e::2:30, and the ASN is 26415.

En esta imagen se pueden apreciar todas las copias del root server identificado con la letra A distribuido por el mundo.

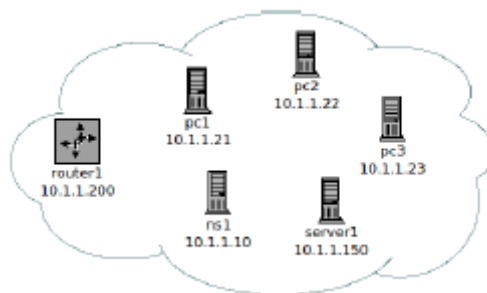
A diferencia de un servidor primario y secundario, que son servidores con IPs diferentes que responden sobre un dominio, en este caso, por ejemplo en el root server A, son 16 servidores con la misma ip, es decir, están clonados.

La copia más cercana de un root server se encuentra en Buenos Aires, en Balcarce 479, San Telmo, Caba.

La letra que identifica a este root server es la letra K, y su nombre de servidor replicado es RIPE NCC.



#### 5. Defina cómo estará compuesta la base de datos de un servidor DNS administrado por Ud., de manera tal que sea el servidor primario del dominio SU-NRO-LEGAJO.tyr.example (.example es un TLD reservado para uso en documentación y ejemplos). De acuerdo al diagrama de la Figura 1, defina:



**Figura 1: Host en la red a definir en dns**

- a. El nombre de todos los hosts en el nuevo dominio, y su respectivo puntero reverso.
- b. El host pc1 y ns1 como name servers del dominio.
- c. **www.SU-NRO-LEGAJO.tyr.example** y **ftp.SU-NRO-LEGAJO.tyr.example** como alias de server1.

**Complete la planilla adjunta a partir de las definiciones previas.**

#### **Documentación de la zona 149615.tyr.example**

Nombre de la zona DNS: 149615.tyr.example

Nombre del servidor DNS: dnstyr.149615.tyr.example

Dirección de correo del contacto: juanmartin\_franco@hotmail.com

Número de serie de la zona: 2021042501

Tiempo de vida en caché: 300 segundos

#### **Resource records para 149615.tyr.example**


HOST	CLASE	TIPO RR	DATOS RR	COMENTARIO
router1	IN	A	10.1.1.200	(ejemplo)
pc1	IN	A	10.1.1.21	
ns1	IN	A	10.1.1.10	
server1	IN	A	10.1.1.150	
pc2	IN	A	10.1.1.22	
pc3	IN	A	10.1.1.23	
www	IN	CNAME	server1	
ftp	IN	CNAME	server1	
	IN	NS	pc1	
	IN	NS	ns1	

#### **Zona de punteros reversos**

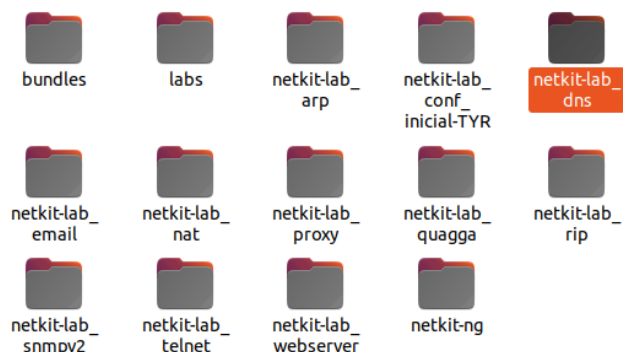
HOST	CLASE	TIPO RR	DATOS DEL RR
200.1.1.10.in-addr.arpa	IN	PTR	router1.149615.tyr.example
21.1.1.10.in-addr.arpa	IN	PTR	pc1.149615.tyr.example
10.1.1.10.in-addr.arpa	IN	PTR	ns1.149615.tyr.example
150.1.1.10.in-addr.arpa	IN	PTR	server1.149615.tyr.example
22.1.1.10.in-addr.arpa	IN	PTR	pc2.149615.tyr.example
23.1.1.10.in-addr.arpa	IN	PTR	pc3.149615.tyr.example

**6. Instale e inicie en el entorno netkit el laboratorio de dns provisto por los docentes disponible en [https://github.com/redesunlu/netkit-labs/raw/master/tarballs/netkit-lab\\_dns-TYR.tar.gz](https://github.com/redesunlu/netkit-labs/raw/master/tarballs/netkit-lab_dns-TYR.tar.gz) y realice las siguientes actividades:**

Para realizar la siguiente actividad, primero ingreso al link y descargo el laboratorio de DNS.

Name	Size	Type	Modified
 netkit-lab_dns	41.9 kB	Folder	22 August 2016, 04...

Una vez descargada, procedo a extraerla en donde están los demás laboratorios de netkit.



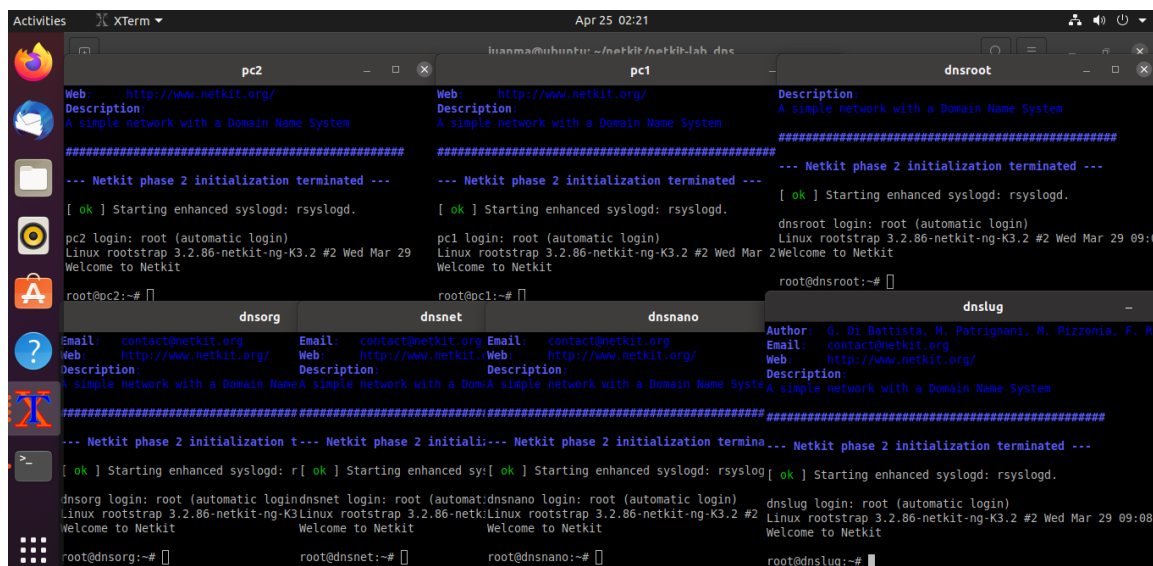
Ahora, voy a iniciar el laboratorio con el comando:

```
lstart
```

El cual cargará 7 máquinas virtuales:

1. pc1
2. pc2
3. dnsroot
4. dnsorg
5. dnsnet
6. dnsnano
7. dnslug

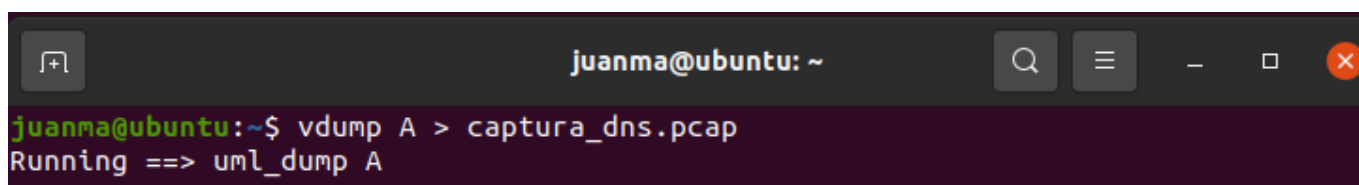
Obteniendo lo siguiente:



### a. Inicie una captura desde el host.

Para realizar una captura, debo utilizar el comando:

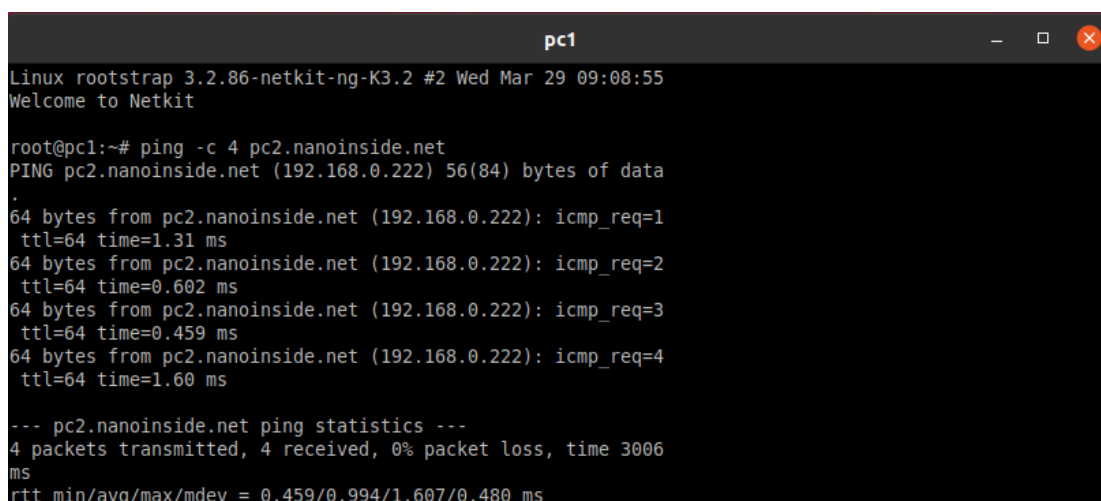
```
vdump A > captura_dns.pcap
```



### b. Desde pc1.lugroma3.org, ejecute el comando ping -c 4 pc2.nanoinside.net

Ahora, ejecuto el comando:

```
ping -c 4 pc2.nanoinside.net
```



### c. Una vez recibidas las 4 respuestas ICMP, detenga la captura.

Ahora, procedo a detener la captura con CTRL + C sobre la terminal que la está realizando.

```
juanma@ubuntu: ~  
juanma@ubuntu:~$ vdump A > captura_dns.pcap  
Running ==> uml_dump A  
^C caught signal 2, cleaning up and exiting  
juanma@ubuntu:~$
```

d. Analice la captura y describa cómo es el proceso de resolución de nombres para determinar la dirección ip de pc2.nanoinside.net, representando gráficamente el intercambio de mensajes dns, e indicando el propósito de cada uno.

Para analizar la captura realizada, utilizo el comando:

```
tshark -r captura_dns.pcap
```

Obteniendo el siguiente resultado:

```
juanma@ubuntu:~/Desktop$ tshark -r captura_dns.pcap  
1 0.000000 96:e9:fc:27:78:fd → Broadcast ARP 42 Who has 192.168.0.11? Tell 192.168.0.111  
2 0.000286 36:06:c5:7e:d5:8e → 96:e9:fc:27:78:fd ARP 42 192.168.0.11 is at 36:06:c5:7e:d5:8e  
3 0.000554 192.168.0.111 → 192.168.0.11 DNS 78 Standard query 0x84c0 A pc2.nanoinside.net  
4 0.016122 36:06:c5:7e:d5:8e → Broadcast ARP 42 Who has 192.168.0.5? Tell 192.168.0.11  
5 0.016328 e6:79:67:cb:b0:ae → 36:06:c5:7e:d5:8e ARP 42 192.168.0.5 is at e6:79:67:cb:b0:ae  
6 0.016470 192.168.0.11 → 192.168.0.5 DNS 89 Standard query 0x04ed A pc2.nanoinside.net OPT  
7 0.017123 192.168.0.11 → 192.168.0.5 DNS 70 Standard query 0x193a NS <Root> OPT  
8 0.043557 192.168.0.5 → 192.168.0.11 DNS 126 Standard query response 0x04ed A pc2.nanoinside.net NS dnsnet.net A 192.168.0.2 OPT  
9 0.044561 36:06:c5:7e:d5:8e → Broadcast ARP 42 Who has 192.168.0.2? Tell 192.168.0.11  
10 0.044766 42:37:ff:57:e0:24 → 36:06:c5:7e:d5:8e ARP 42 192.168.0.2 is at 42:37:ff:57:e0:24  
11 0.044900 192.168.0.11 → 192.168.0.2 DNS 89 Standard query 0x6149 A pc2.nanoinside.net OPT  
12 0.048876 192.168.0.2 → 192.168.0.11 DNS 127 Standard query response 0x6149 A pc2.nanoinside.net NS dnsnano.nanoinside.net A 192.168.0.22 OPT  
13 0.050631 192.168.0.5 → 192.168.0.11 DNS 110 Standard query response 0x193a NS <Root> NS ROOT-SERVER A 192.168.0.5 OPT  
14 0.050939 36:06:c5:7e:d5:8e → Broadcast ARP 42 Who has 192.168.0.22? Tell 192.168.0.11  
15 0.051067 7a:4a:1a:19:b1:0a → 36:06:c5:7e:d5:8e ARP 42 192.168.0.22 is at 7a:4a:1a:19:b1:0a  
16 0.051209 192.168.0.11 → 192.168.0.22 DNS 89 Standard query 0x9d69 A pc2.nanoinside.net OPT  
17 0.070034 192.168.0.22 → 192.168.0.11 DNS 143 Standard query response 0x9d69 A pc2.nanoinside.net A 192.168.0.222 NS dnsnano.nanoinside.net A 192.168.0.22 OPT  
18 0.071048 192.168.0.11 → 192.168.0.111 DNS 132 Standard query response 0x84c0 A pc2.nanoinside.net A 192.168.0.222 NS dnsnano.nanoinside.net A 192.168.0.22  
19 0.072233 96:e9:fc:27:78:fd → Broadcast ARP 42 Who has 192.168.0.222? Tell 192.168.0.111  
20 0.072328 de:fc:2e:30:0d:d5 → 96:e9:fc:27:78:fd ARP 42 192.168.0.222 is at de:fc:2e:30:0d:d5  
21 0.073140 192.168.0.111 → 192.168.0.222 ICMP 98 Echo (ping) request id=0x05ed, seq=1/256, ttl=64  
22 0.073290 192.168.0.222 → 192.168.0.111 ICMP 98 Echo (ping) reply id=0x05ed, seq=1/256, ttl=64 (request in 21)  
23 0.073873 192.168.0.111 → 192.168.0.11 DNS 86 Standard query 0x8d98 PTR 222.0.168.192.in-addr.arpa  
24 0.077136 192.168.0.11 → 192.168.0.5 DNS 97 Standard query 0xd1a2 PTR 222.0.168.192.in-addr.arpa OPT  
25 0.081787 192.168.0.5 → 192.168.0.11 DNS 161 Standard query response 0xd1a2 PTR 222.0.168.192.in-addr.arpa PTR pc2.nanoinside.net NS 5.0.168.192.in-addr.arpa A 192.168.0.5 OPT  
26 0.083379 192.168.0.11 → 192.168.0.111 DNS 150 Standard query response 0x8d98 PTR 222.0.168.192.in-addr.arpa PTR pc2.nanoinside.net NS 5.0.168.192.in-addr.arpa A 192.168.0.5  
27 1.073586 192.168.0.111 → 192.168.0.222 ICMP 98 Echo (ping) request id=0x05ed, seq=2/512, ttl=64
```

```

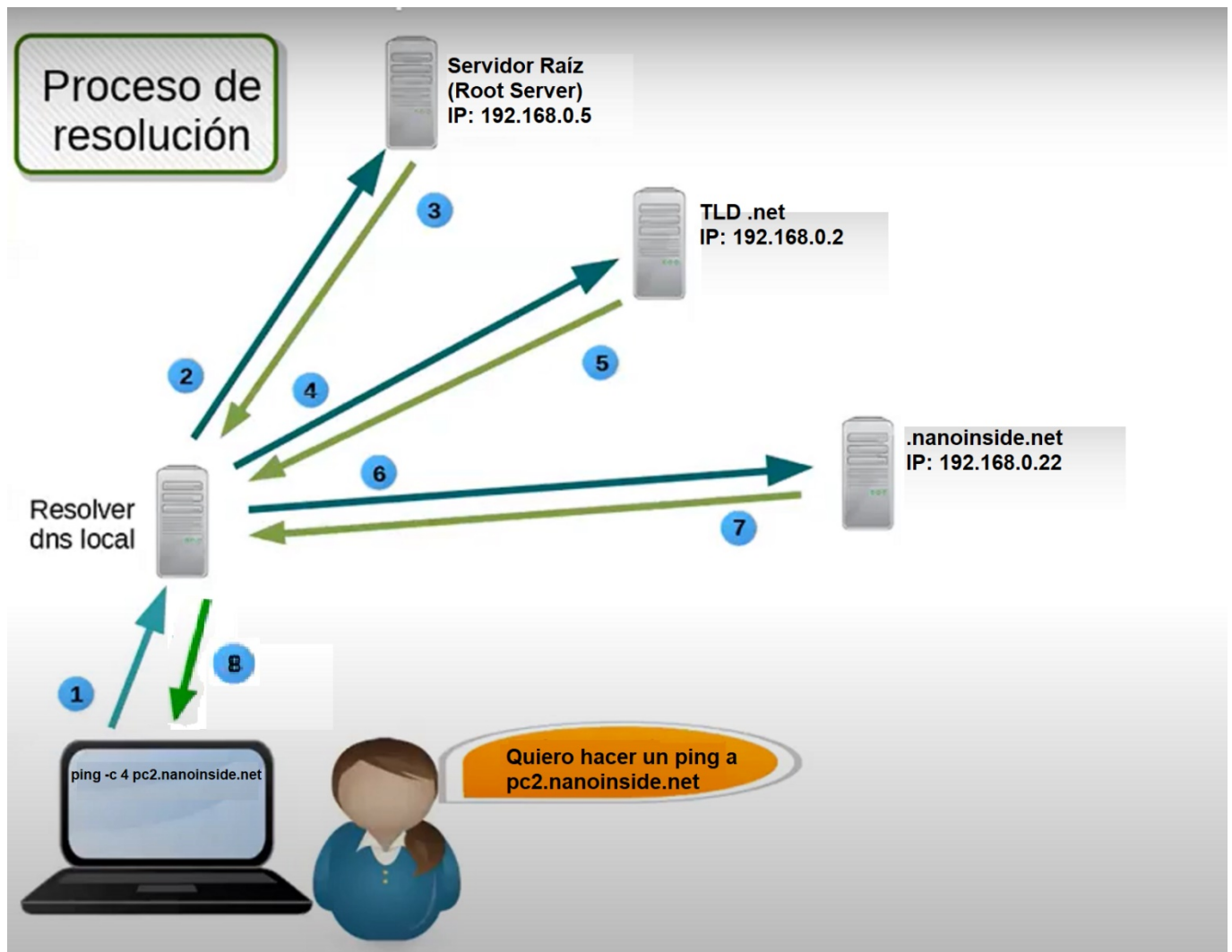
27 1.073586 192.168.0.111 → 192.168.0.222 ICMP 98 Echo (ping) request id=0x05ed, seq=2/512, ttl=64
28 1.073841 192.168.0.222 → 192.168.0.111 ICMP 98 Echo (ping) reply id=0x05ed, seq=2/512, ttl=64 (request in 27)
29 1.079162 192.168.0.111 → 192.168.0.11 DNS 86 Standard query 0xbb7f PTR 222.0.168.192.in-addr.arpa
30 1.083774 192.168.0.11 → 192.168.0.5 DNS 97 Standard query 0x6a67 PTR 222.0.168.192.in-addr.arpa OPT
31 1.084116 192.168.0.5 → 192.168.0.11 DNS 161 Standard query response 0x6a67 PTR 222.0.168.192.in-addr.arpa PTR pc2.nanoinside.net NS 5
.0.168.192.in-addr.arpa A 192.168.0.5 OPT
32 1.089325 192.168.0.11 → 192.168.0.5 DNS 70 Standard query 0xcc7 NS <Root> OPT
33 1.089882 192.168.0.5 → 192.168.0.11 DNS 110 Standard query response 0xcc7 NS <Root> NS ROOT-SERVER A 192.168.0.5 OPT
34 1.094724 192.168.0.11 → 192.168.0.111 DNS 150 Standard query response 0xbb7f PTR 222.0.168.192.in-addr.arpa PTR pc2.nanoinside.net NS
5.0.168.192.in-addr.arpa A 192.168.0.5
35 2.074247 192.168.0.111 → 192.168.0.222 ICMP 98 Echo (ping) request id=0x05ed, seq=3/768, ttl=64
36 2.074503 192.168.0.222 → 192.168.0.111 ICMP 98 Echo (ping) reply id=0x05ed, seq=3/768, ttl=64 (request in 35)
37 2.074887 192.168.0.111 → 192.168.0.11 DNS 86 Standard query 0xe669 PTR 222.0.168.192.in-addr.arpa
38 2.083445 192.168.0.11 → 192.168.0.5 DNS 97 Standard query 0x9ad8 PTR 222.0.168.192.in-addr.arpa OPT
39 2.083765 192.168.0.5 → 192.168.0.11 DNS 161 Standard query response 0x9ad8 PTR 222.0.168.192.in-addr.arpa PTR pc2.nanoinside.net NS 5
.0.168.192.in-addr.arpa A 192.168.0.5 OPT
40 2.088282 192.168.0.11 → 192.168.0.5 DNS 70 Standard query 0x6de6 NS <Root> OPT
41 2.089570 192.168.0.5 → 192.168.0.11 DNS 110 Standard query response 0x6de6 NS <Root> NS ROOT-SERVER A 192.168.0.5 OPT
42 2.094808 192.168.0.11 → 192.168.0.111 DNS 150 Standard query response 0xe669 PTR 222.0.168.192.in-addr.arpa PTR pc2.nanoinside.net NS
5.0.168.192.in-addr.arpa A 192.168.0.5
43 3.079206 192.168.0.111 → 192.168.0.222 ICMP 98 Echo (ping) request id=0x05ed, seq=4/1024, ttl=64
44 3.079658 192.168.0.222 → 192.168.0.111 ICMP 98 Echo (ping) reply id=0x05ed, seq=4/1024, ttl=64 (request in 43)
45 3.083575 192.168.0.111 → 192.168.0.11 DNS 86 Standard query 0x6272 PTR 222.0.168.192.in-addr.arpa
46 3.084376 192.168.0.11 → 192.168.0.5 DNS 97 Standard query 0x10b2 PTR 222.0.168.192.in-addr.arpa OPT
47 3.088432 192.168.0.11 → 192.168.0.5 DNS 70 Standard query 0xf2a NS <Root> OPT
48 3.089682 192.168.0.5 → 192.168.0.11 DNS 161 Standard query response 0x10b2 PTR 222.0.168.192.in-addr.arpa PTR pc2.nanoinside.net NS 5
.0.168.192.in-addr.arpa A 192.168.0.5 OPT
49 3.090294 192.168.0.5 → 192.168.0.11 DNS 110 Standard query response 0xf2a NS <Root> NS ROOT-SERVER A 192.168.0.5 OPT
50 3.091517 192.168.0.11 → 192.168.0.111 DNS 150 Standard query response 0x6272 PTR 222.0.168.192.in-addr.arpa PTR pc2.nanoinside.net NS
5.0.168.192.in-addr.arpa A 192.168.0.5
51 5.064121 7a:4a:1a:19:b1:0a → 36:06:c5:7e:d5:8e ARP 42 Who has 192.168.0.11? Tell 192.168.0.22
52 5.064610 36:06:c5:7e:d5:8e → 7a:4a:1a:19:b1:0a ARP 42 192.168.0.11 is at 36:06:c5:7e:d5:8e
53 5.078810 36:06:c5:7e:d5:8e → 96:e9:fc:27:78:fd ARP 42 Who has 192.168.0.111? Tell 192.168.0.11
54 5.078969 96:e9:fc:27:78:fd → 36:06:c5:7e:d5:8e ARP 42 192.168.0.111 is at 96:e9:fc:27:78:fd
55 5.090035 de:fc:2e:30:0d:d5 → 96:e9:fc:27:78:fd ARP 42 Who has 192.168.0.111? Tell 192.168.0.222
56 5.090134 96:e9:fc:27:78:fd → de:fc:2e:30:0d:d5 ARP 42 192.168.0.111 is at 96:e9:fc:27:78:fd
57 5.090140 e6:79:67:cb:b0:ae → 36:06:c5:7e:d5:8e ARP 42 Who has 192.168.0.11? Tell 192.168.0.5
58 5.090267 36:06:c5:7e:d5:8e → e6:79:67:cb:b0:ae ARP 42 192.168.0.11 is at 36:06:c5:7e:d5:8e

```

Para explicar esto, decidí utilizar una imagen (editada en Paint, sobre la plantilla que utilizó Fernando para explicar el funcionamiento de DNS) ya que me parece una forma muy sencilla de entender que fue lo que comprendí y en caso de haber un error de comprensión que sea más fácil de visualizar.

A continuación adjunto la imagen junto con la descripción paso por paso del proceso de resolución hasta llegar a los 4 pings realizados.





### Paso a paso del proceso de resolución:

1- Se hace la consulta para averiguar la dirección IP de `pc2.nanoinside.net`, la cual va a ser resuelta por mi resolver dns local (en mi caso, el provisto por mi ISP). Como el resolver va a ser el encargado de encontrar de alguna manera esa dirección IP, se dice que es una query recursiva, por lo que RD = SI.

Q = A `pc2.nanoinside.net` RD = SI

2- Ahora, mi resolver va a intentar encontrar esa dirección IP consultando por el registro A de dicho host (`pc2.nanoinside.net`) de manera iterativa. Para hacer esto, pregunta a algún root server si conoce dicho host.

Q = A `pc2.nanoinside.net` RD = NO

Como el root server no es quien me va a entregar la respuesta definitiva (ya que sería algo poco lógico, porque se provocaría una sobrecarga si eso fuese así), RD = NO.

3- El root server, no conoce a `pc2.nanoinside.net`, pero si conoce al host que tiene autoridad sobre el TLD `.net`, por lo que me pasa la lista de NS de `.net` y sus direcciones IP en la Additional Section.

R = A pc2.nanoinside.net  
Authority Section  
NS .net → dnsnet.net  
Additional Section  
A dnsnet.net → 192.168.0.2

4- Una vez obtenida la dirección IP del TLD .net, mi resolver procede a consultarle a dicho TLD (ya conoce su IP) si conoce la dirección IP del host pc2.nanoinside.net

Q = A pc2.nanoinside.net RD = NO

5- El TLD .net no conoce a pc2.nanoinside.net, pero si conoce al host (o los hosts) que tiene(n) autoridad sobre el dominio .nanoinside.net, por lo que me pasa su información.

R = A pc2.nanoinside.net  
Authority Section  
NS .nanoinside → dnsnano.nanoinside.net  
Additional Section  
A dnsnano.nanoinside.net → 192.168.0.22

6- Ahora, mi resolver va a consultarle a la última dirección IP recibida, quien es autoridad de .nanoinside.net, si conoce al host pc2.nanoinside.net.

Q = A pc2.nanoinside.net RD = NO

7- Efectivamente, .nanoinside.net conoce al host pc2.nanoinside.net, por lo que me envía su dirección IP.

R = A pc2.nanoinside.net → 192.168.0.222

8- Por último, mi resolver logró ubicar la dirección IP de pc2.nanoinside.net, por lo que me la envía.

R = A pc2.nanoinside.net → 192.168.0.222 (del resolver a pc1)

Una vez finalizados todos estos pasos, se procede a realizar los pings.

```
21  0.073140 192.168.0.111 → 192.168.0.222 ICMP 98 Echo (ping) request  id=0x05ed, seq=1/256, ttl=64
22  0.073290 192.168.0.222 → 192.168.0.111 ICMP 98 Echo (ping) reply   id=0x05ed, seq=1/256, ttl=64 (request in 21)
```

Luego de encontrada la dirección IP y realizado 1 ping, se accede al host utilizando los punteros reversos (se puede visualizar en la siguiente imagen la presencia de 222.0.168.192.in-addr.arpa).

```
38  2.083445 192.168.0.11 → 192.168.0.5  DNS 97 Standard query 0x9ad8 PTR 222.0.168.192.in-addr.arpa OPT
39  2.083765 192.168.0.5 → 192.168.0.11  DNS 161 Standard query response 0x9ad8 PTR 222.0.168.192.in-addr.arpa PTR pc2.nanoinside.net NS 5
.0.168.192.in-addr.arpa A 192.168.0.5 OPT
40  2.088282 192.168.0.11 → 192.168.0.5  DNS 70 Standard query 0x6de6 NS <Root> OPT
41  2.089570 192.168.0.5 → 192.168.0.11  DNS 110 Standard query response 0x6de6 NS <Root> NS ROOT-SERVER A 192.168.0.5 OPT
42  2.094808 192.168.0.11 → 192.168.0.111 DNS 150 Standard query response 0xe669 PTR 222.0.168.192.in-addr.arpa PTR pc2.nanoinside.net NS
5.0.168.192.in-addr.arpa A 192.168.0.5
```



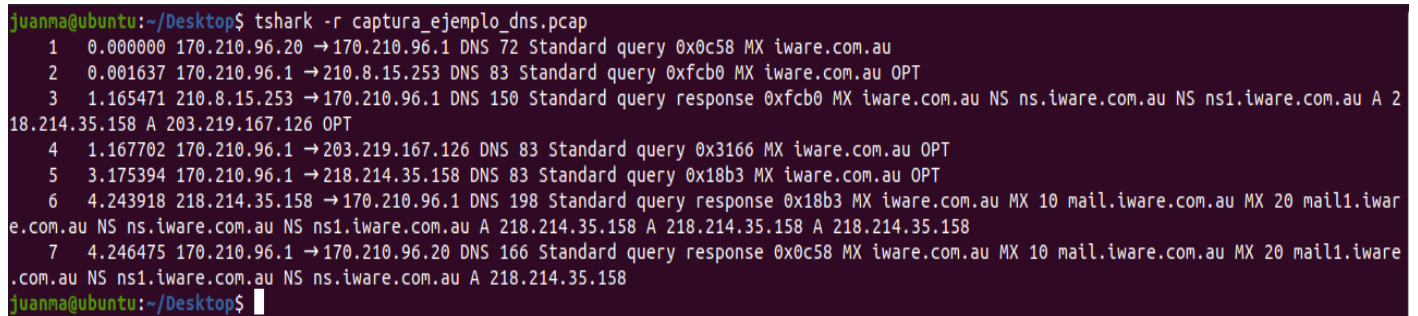
## 7. Analice la captura `captura_ejemplo_dns.pcap` y represente el intercambio de mensajes. ¿Puede indicar alguna particular que observe en la misma?

Una vez descargada la captura desde [http://www.labredes.unlu.edu.ar/sites/www.labredes.unlu.edu.ar/files/site/data/tyr/captura\\_ejemplo\\_dns.pcap](http://www.labredes.unlu.edu.ar/sites/www.labredes.unlu.edu.ar/files/site/data/tyr/captura_ejemplo_dns.pcap), procedo a guardarla en el escritorio para analizarla con tshark.

Para realizar dicho análisis, ejecuto el comando:

```
tshark -r captura_ejemplo_dns.pcap
```

Obteniendo el siguiente resultado:



```
juanna@ubuntu:~/Desktop$ tshark -r captura_ejemplo_dns.pcap
 1  0.000000 170.210.96.20 → 170.210.96.1 DNS 72 Standard query 0xc58 MX iware.com.au
 2  0.001637 170.210.96.1 → 210.8.15.253 DNS 83 Standard query 0xfcb0 MX iware.com.au OPT
 3  1.165471 210.8.15.253 → 170.210.96.1 DNS 150 Standard query response 0xfcb0 MX iware.com.au NS ns.iware.com.au NS ns1.iware.com.au A 218.214.35.158 A 203.219.167.126 OPT
 4  1.167702 170.210.96.1 → 203.219.167.126 DNS 83 Standard query 0x3166 MX iware.com.au OPT
 5  3.175394 170.210.96.1 → 218.214.35.158 DNS 83 Standard query 0x18b3 MX iware.com.au OPT
 6  4.243918 218.214.35.158 → 170.210.96.1 DNS 198 Standard query response 0x18b3 MX iware.com.au MX 10 mail.iware.com.au MX 20 mail1.iware.com.au NS ns.iware.com.au NS ns1.iware.com.au A 218.214.35.158 A 218.214.35.158 A 218.214.35.158
 7  4.246475 170.210.96.1 → 170.210.96.20 DNS 166 Standard query response 0xc58 MX iware.com.au MX 10 mail.iware.com.au MX 20 mail1.iware.com.au NS ns1.iware.com.au NS ns.iware.com.au A 218.214.35.158
juanna@ubuntu:~/Desktop$
```

### Análisis de la captura:

Primero, en el frame 1, logro interpretar que desde el host cuya ip es 170.210.96.20 se pide el registro MX del host iware.com.au, y esto se le envía al resolver cuya ip es 170.210.96.1.

Luego, en el frame 2, el resolver le pregunta a algún root server (supongo) si conoce el registro MX de iware.com.au.

En el frame 3, el root server le indica que no conoce al host iware.com.au pero si conoce a los NS que son autoridad de dicho dominio (y, en la Additional Section, le envía la dirección IP de ambos NS).

En el 4to y 5to frame, el resolver consulta por el registro MX hacia ambos NS (203.219.167.126 y 218.214.35.158).

En el 6to frame, uno de los NS le responde al resolver 3 registros MXs, 3NSs y 3IPs idénticas.

En el último frame, el resolver le envía al host principal (quien le solicitó el registro MX) los 3 registros MX obtenidos, 2 NSs y una dirección IP.

### Las particularidades que noto a simple vista son:

1- Como se ve en el frame 4 y 5, primero se envía una query (frame 4) a 203.219.167.126, pero no se obtiene respuesta alguna dentro del tiempo esperado (expiró el timer), por lo que se envía la misma query (frame 5) a 218.214.35.158, sobre la cual SI se obtiene respuesta en el siguiente frame (frame 6).

2- Mirando con más atención la captura, las 3 IPs devueltas en el frame 6 son idénticas, lo que me hace pensar que los 3 registros MX tienen asociada la misma IP.

## 8. ¿Cómo un desarrollador de aplicaciones puede acceder al servicio DNS? (Por ej. si es necesario resolver, en una aplicación de software, mnemónicos a direcciones IP o viceversa)

Según mi forma de ver las cosas, suponiendo que estoy desarrollando una aplicación en un lenguaje X, por ejemplo Java (Mi lenguaje favorito) existen librerías asociadas a la resolución de dichos “problemas”.

Por ejemplo, googleando un poco encontré que java cuenta con dnsjava (<https://github.com/dnsjava/dnsjava>).

Según su documentación, puede ser utilizada para hacer consultas, transferencias de zonas y DDNS.

En su respectivo Javadoc, dnsjava cuenta con 5 packages.

org.xbill.DNS:

### Interfaces:

1. PacketLogger: Cuenta con un método log para registrar los paquetes enviados o recibidos.
2. Resolver: Permite modificar el timer para esperar una respuesta antes de “renunciar” a la query, enviar un mensaje y esperar una respuesta, enviar un mensaje de manera asíncrona, settear puertos, etc.
3. ZoneTransferHandler: Como lo indica su nombre, esta clase maneja las transferencias de zonas.

Además de las interfaces, cuenta con clases para manejar la caché, headers, TTL, etc.

El resto de los paquetes (org.xbill.DNS.config, org.xbill.DNS.spi, org.xbill.DNS.tools, org.xbill.DNS.utils) contienen clases con métodos para configurar, referentes a NSs, etc.)

También se pueden manejar mediante los métodos de la clase InetAddress, la cual representa una dirección IP.

Dicha clase cuenta con métodos como `getByName(String host)`, el cual recibe el nombre de host y devuelve la dirección IP de dicho host, en caso de no encontrar un host con ese nombre pasado como argumento, arroja una excepción del tipo `UnknownHostException`.

Como método inverso, existe la posibilidad de obtener el nombre de host de una dirección específica utilizando el método `getHostName()`.

## 9. ¿Quién tiene a su cargo la administración de los nombres de dominio bajo el dominio .ar ? ¿Qué y cuáles son las zonas especiales? ¿Qué requisito especial se requiere para solicitar un dominio .org.ar?

La raíz es manejada por ICANN, y es la que define cuales son los posibles dominios de nivel superior (TLD).

Ahora bien, icann delega en algún organismo la autoridad sobre el dominio “.ar” para que esa autoridad lo administre, lo que significa que ya no es responsabilidad de icann el manejo de los subdominios de .ar.

Quien define cuales son los dominios válidos de .ar es el organismo al cual icann le delegó la autoridad, en este caso particular es **nic.ar**.

Existen zonas llamadas zonas especiales, las cuales permiten ordenar el espacio de Internet y brindar especificidad a los sitios '.ar'.

Las zonas especiales (según nic.ar) son las siguientes: '.coop.ar', '.gob.ar', '.int.ar', '.mil.ar', '.musica.ar', '.mutual.ar', '.org.ar', '.senasa.ar' o '.tur.ar'

Según la página de nic.ar, los dominios ".org.ar" están reservados únicamente para organizaciones sin fines de lucro.

Sólo pueden registrar este tipo de dominios Personas Jurídicas, argentinas o extranjeras.

Al momento de solicitar la Habilitación de Zonas Especiales desde Trámites a Distancia deberán presentar una copia del Estatuto / Contrato Social.