

CONFIGURACIÓN Y DESARROLLO DE APLICACIONES EN RED

Trabajo práctico especial



Grupo Numero 25

Profesor Designado : Leonardo Dominguez

Integrantes:

Cuculich, Agustina.

cuculichagustina@gmail.com

Oroquieta, Luciano.

oroquietaluciano@gmail.com

EJERCICIO 1.a)**Biblioteca**

Nombre de red	Descripción IP's necesarias	Tamaño del bloque asignado	Máscara resultante
Secretaría (SW3)	28 host (router + 27 conexiones) + base + broadcast	$\log_2 30 = 4.9 \rightarrow 5$ $2^5 = 32$ direcciones	/27 (32 - 5)
WLAN (p.baja) (R6)	100 host + router + base + broadcast	$\log_2 103 = 6.6 \rightarrow 7$ $2^7 = 128$ direcciones	/25 (32 - 7)
WLAN (p.alta) (R6)	100 host + router + base + broadcast	$\log_2 103 = 6.6 \rightarrow 7$ $2^7 = 128$ direcciones	/25 (32 - 7)
Centro de cómputo (SW4)	16 host + router + base + broadcast	$\log_2 19 = 4.2 \rightarrow 5$ $2^5 = 32$ direcciones	/27 (32 - 5)
R5 → R6	2 host + base + broadcast	$2^2 = 4$ direcciones	/30 (32 - 2)

Aulas Comunes

Nombre de red	Descripción IP's necesarias	Tamaño del bloque asignado	Máscara resultante
Router 10-A1	100 host + router + base + broadcast	128 direcciones	/25
Router 10-A2	100 host + router + base + broadcast	128 direcciones	/25
Router 10-A3	100 host + router + base + broadcast	128 direcciones	/25
Switch 7	32 host + router + base + broadcast	64 direcciones	/26
R9 → R10	2 host + base + broadcast	4 direcciones	/30

Comedor

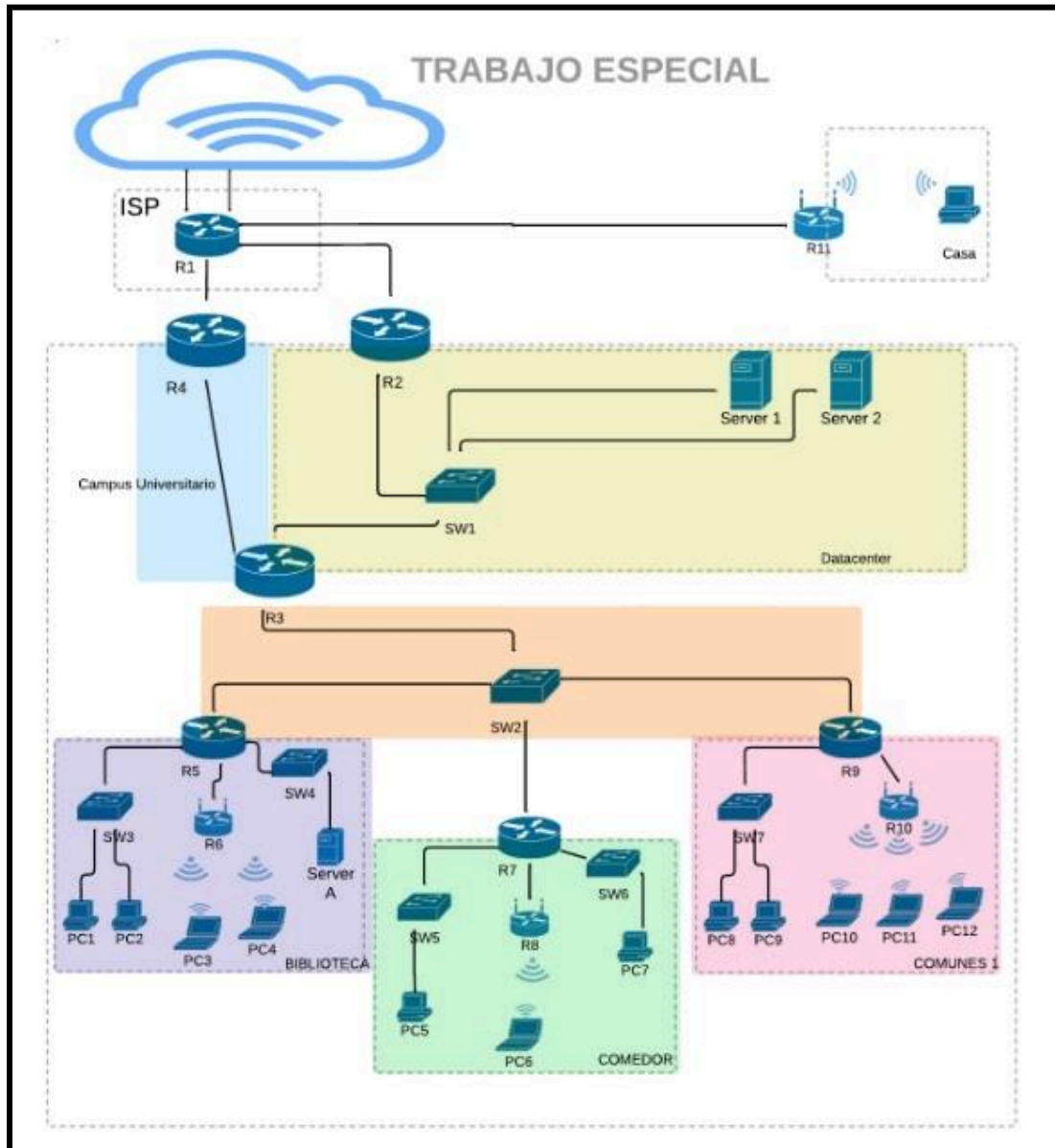
Nombre de red	Descripción IP's necesarias	Tamaño del bloque asignado	Máscara resultante
Router 8	100 host + base + broadcast	128 direcciones	/25
Switch 6	45 host (router + 44 conexiones) + base + broadcast	64 direcciones	/26
Switch 5	8 host (router + 7 conexiones) + base + broadcast	16 direcciones	/28
R7□R8	2 host + base + broadcast	4 direcciones	/30

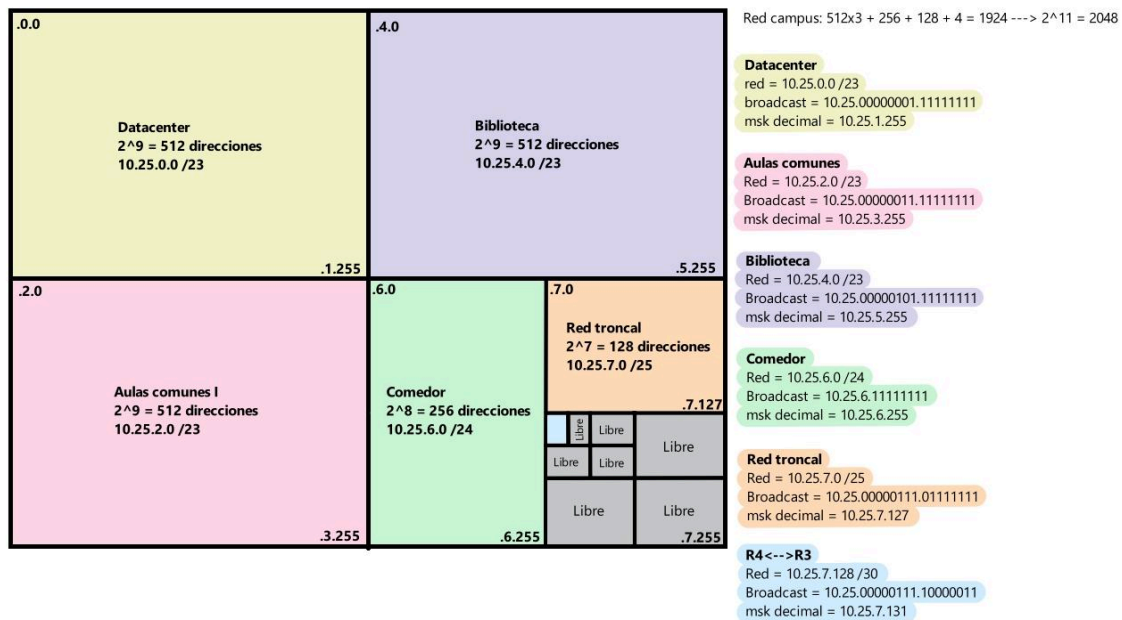
EJERCICIO 1 .b) Campus universitario

Nombre de red	Descripción IP's necesarias	Tamaño del bloque asignado	Máscara resultante
Datacenter	500 host + Base + Broadcast + 2 Routers = 504 direcciones	$2^9 = 512 \text{ direcciones}$	/23 (32 - 9)
Aulas Comunes	R10 (A1): 128 direcciones + R10 (A2): 128 direcciones + R10 (A3): 128 direcciones + Switch 7: 64 direcciones + R9 -> R10: 4 direcciones = 452 direcciones	$2^9 = 512 \text{ direcciones}$	/23 (32 - 9)

Biblioteca	WLAN(p.baja): 128 direcciones + WLAN(p.alta): 128 direcciones + Secretaría: 32 direcciones + C. cómputo: 32 direcciones + R5 → R6: 4 direcciones = 324 direcciones	$\log_2 324 = 8.3 \rightarrow 9$ $2^9 = 512 \text{ direcciones}$	/23 (32 - 9)
Comedor	Switch 5 16 direcciones + Switch 6 (64 direcciones) + Router 8(128 direcciones) + R7↔R8 (4 direcciones) = 212 direcciones	$\log_2 212 = 7.7 \rightarrow 8$ $2^8 = 256 \text{ direcciones}$	/24 (32 - 8)
Red troncal	80 host + Router 3 + Base + Broadcast = 83 direcciones	128 direcciones	/25 (32 - 7)
R4↔R3	2 Direcciones + base + broadcast	4 direcciones	/30 (32-2)

EJERCICIO 1.c)



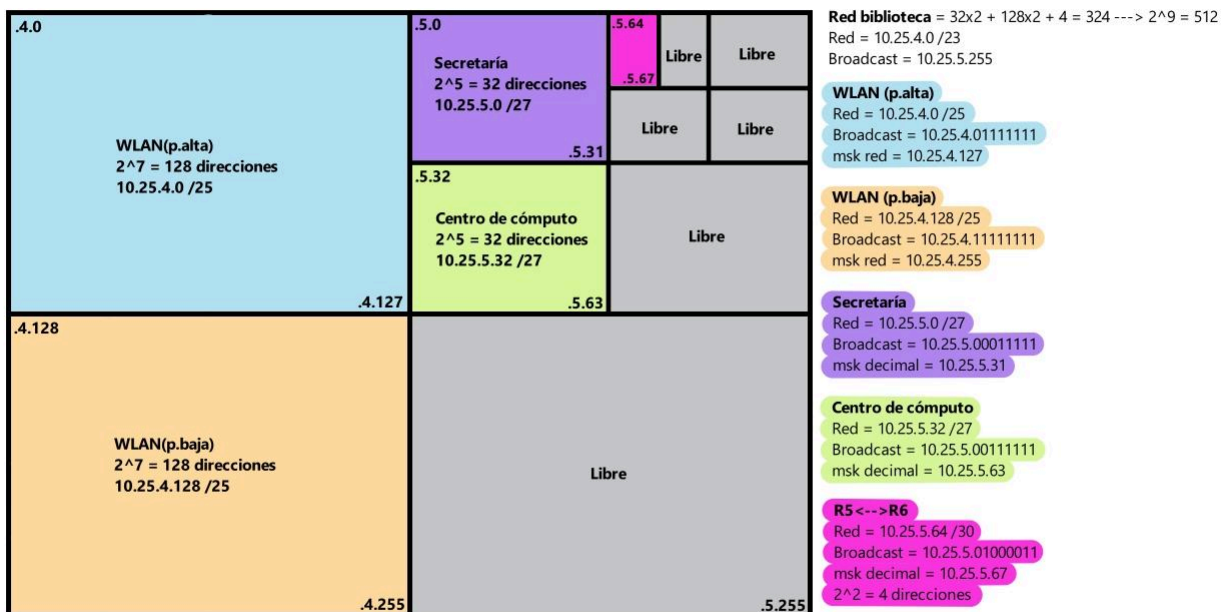


En este VLSM del **CAMPUS UNIVERSITARIO** graficamos solamente el bloque que va de 10.25.0.0 hasta 10.27.7.255 ya que en ese rango vamos a tener nuestras redes del campus. Por otro lado, al requerir un total de 2048 direcciones en total, vamos a trabajar con VLSM /16 (65.536 direcciones disponibles).

Como dato de color nos quedo incomodo poner la direccion base y broadcast cuando subdividimos en 4 direcciones ip y simplemente coloreamos esa parte que iria de (.7.128 hasta .7.131), todo lo demas queda libre o disponible hasta llegar al rango de .7.255 que es lo que tenemos en el diagrama del campus con lo especificado en los incisos.

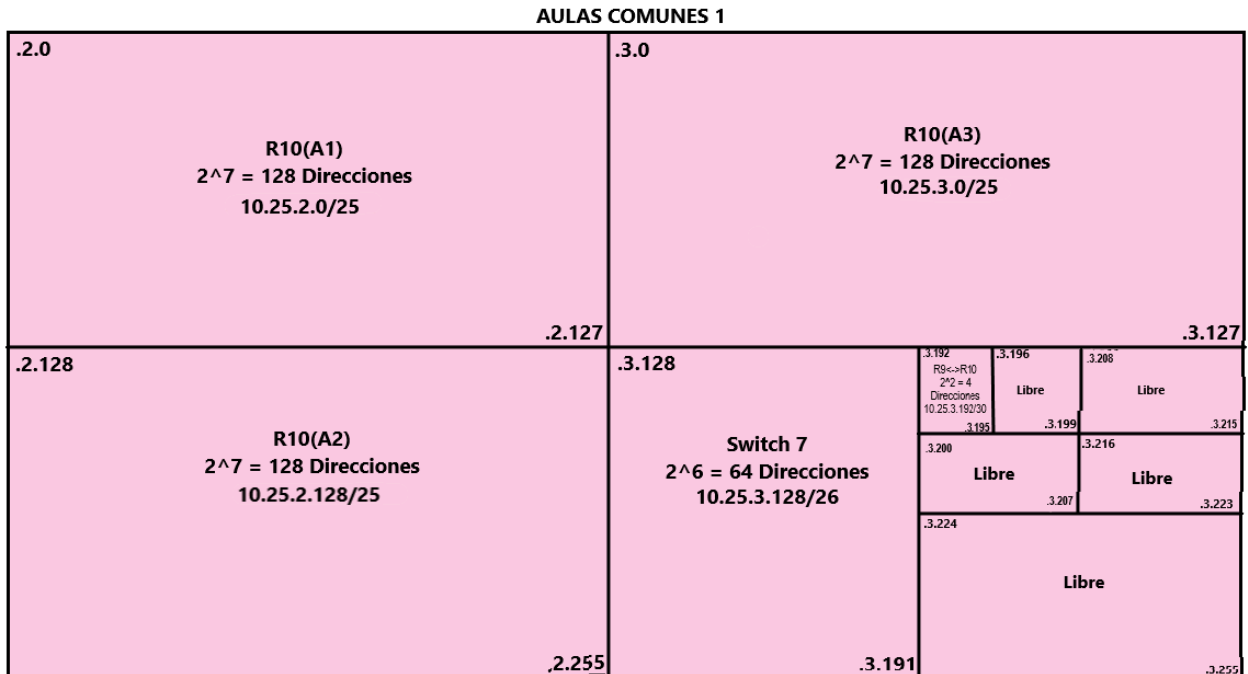
EJERCICIO 1 .d)

Biblioteca



Para cumplir con todos los requerimientos de la red, partimos de un bloque de 512 direcciones el cual dividiremos utilizando el método VLSM (Variable Length Subnet Mask) para un uso óptimo de la cantidad de IP's disponibles. Siguiendo esto, dividimos el bloque en cuatro partes iguales, las cuales tendrán la capacidad de albergar 128 direcciones c/u.

Tomamos el sector de la biblioteca que tenga mayor requerimiento de direcciones IP, en este caso, podría ser la WLAN del piso de arriba o de abajo (ya que soportan la misma cantidad de direcciones). Partiendo de la base 10.25.4.0 y tomando en cuenta la cantidad de IP necesarias (128 direcciones), reservamos aquellas entre 10.25.4.0 (base) hasta la 10.25.5.127 (broadcast). De la misma manera, tomamos el cuadrante de abajo y seguimos el mismo procedimiento (literalmente el mismo ya que tienen misma máscara). El siguiente paso será dividir el tercer cuadrante (aquel que está a la derecha del primer cuadrante que ocupamos), y los dividimos 2 veces, quedando cuatro cuadrantes de 32 direcciones c/u. Utilizaremos 2 de estos cuadrantes más chicos para asignar direcciones a secretaría (32 direcciones) y sala de cómputo (32 direcciones). El paso final será volver a realizar este procedimiento para la última subred (punto a punto) la cual requiere 4 direcciones.



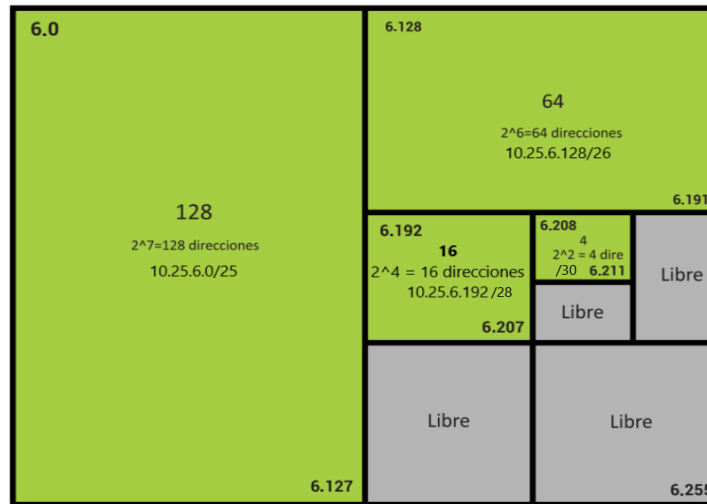
Aulas comunes 1 (Dirección de red = 10.25.2.0) .

En el diseño de mi red utilizando el método de VLSM (Variable Length Subnet Mask), comienzo con un bloque general que tiene 512 direcciones IP disponibles. Para optimizar el uso de direcciones, divido este bloque primero verticalmente y luego horizontalmente, resultando en tres bloques de 128 direcciones cada uno.

Cuando llego a la dirección .255 en una de las subredes, completo el bit y esto provoca un salto a la próxima dirección IP disponible, que sería .3.0 por ejemplo. Continúo este proceso hasta alcanzar el máximo de direcciones disponibles para las aulas comunes 1, que es .3.255.

Esto me permite optimizar el uso de direcciones IP y garantizar que cada dispositivo en mi red tenga una dirección única y adecuada para su función dentro de la red.

COMEDOR



Red comedor (10.25.6.0).

El VLSM de esta red tiene la capacidad de alojar alrededor de 258 direcciones IP. Para que funcione correctamente hemos dividido la red en 4 subredes.

La primera subred que es el router 8 (180 direcciones) la cuales van de las 10.25.6.0 hasta la 10.25.6.127. La siguiente subred es la de switch 6 (64 direcciones) siempre va a ser uno más que el broadcast y esto se repite para el resto de subredes por lo tanto la sub red va a quedar desde la 10.25.6.128 hasta la 10.25.6.191. la tercera subred es del switch 5 (16 direcciones) que va desde la 10.25.6.192 hasta la 10.25.6.207 y la ultima subred es la coneccion de routers de R7->R8 (4 direcciones) la cuales van desde 10.25.6.211 y su maximo de red es hasta la 10.25.6.255 .

EJERCICIO 1 .e)**CAMPUS UNIVERSITARIO**

Nombre de red	Direccion base	Mascara	Rango Asignable	Broadcast
Datacenter	10.25.0.0	/23	10.25.0.1 10.25.1.254	10.25.1.255
Aulas Comunes	10.25.2.0	/25	10.25.2.1 10.25.3.254	10.25.3.255
Biblioteca	10.25.4.0	/23	10.25.4.1 10.25.5.254	10.25.5.255
Comedor	10.25.6.0	/24	10.35.6.1 10.35.6.254	10.35.6.255
Red troncal	10.25.7.0	/25	10.25.7.1 10.25.7.126	10.25.7.127
R4<-->R3	10.25.7.128	/30	10.25.7.129 10.25.7.130	10.25.7.131

AULAS COMUNES 1

Nombre de red	Direccion base	Mascara	Rango Asignable	Broadcast
R10(A1)	10.25.2.0	/25	10.25.2.1- 10.25.2.127	10.25.2.128
R10(A2)	10.25.2.128	/25	10.25.2.129- 10.25.2.254	10.25.2.255
R10(A3)	10.25.3.0	/25	10.25.3.1- 10.25.3.127	10.25.3.128
Switch 7	10.25.3.128	/26	10.25.3.129- 10.25.3.191	10.25.3.192

R9←→R10	10.25.3.192	/30	10.25.3.193- 10.25.3.194	10.25.3.195
---------	-------------	-----	-----------------------------	-------------

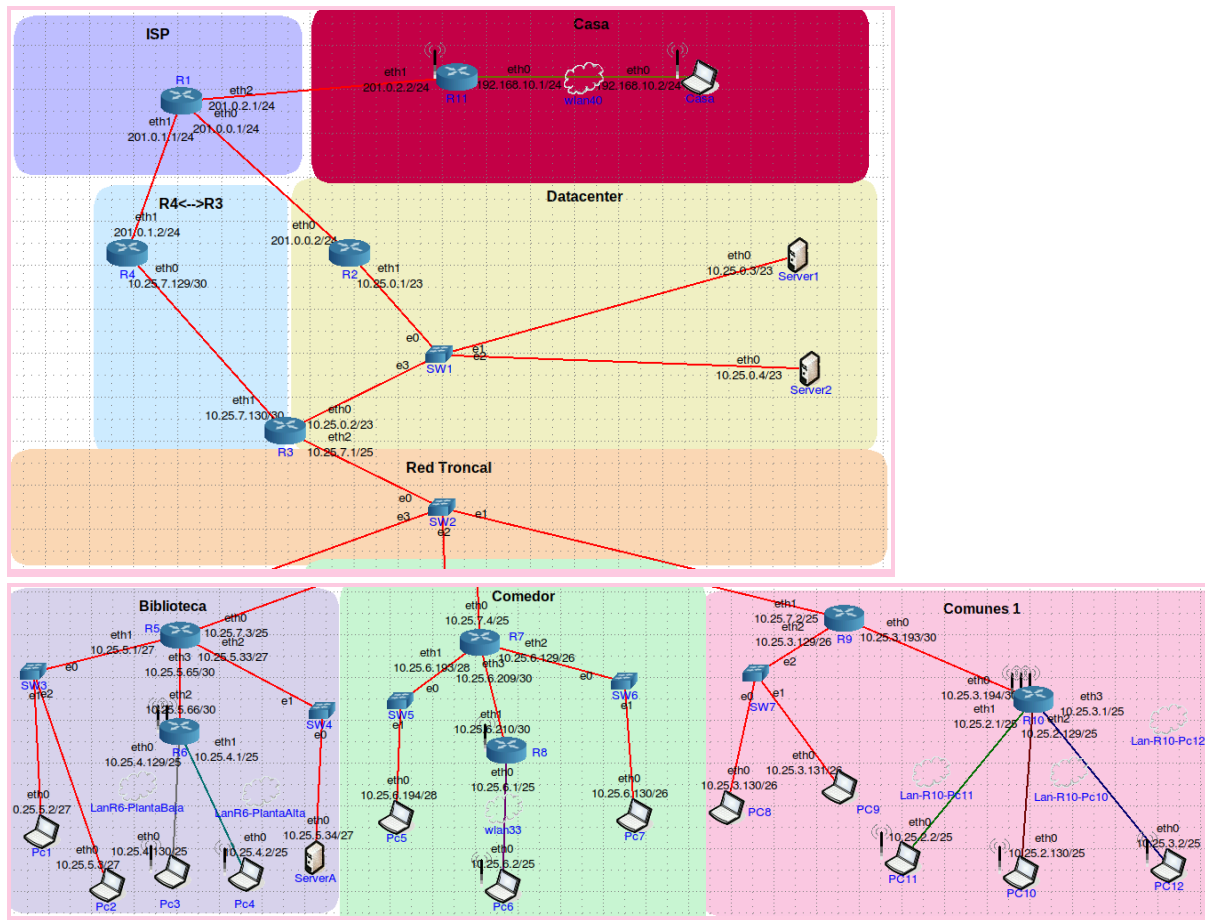
COMEDOR

Nombre de red	Direccion base	Mascara	Rango Asignable	Broadcast
R8	10.25.6.0	/24	10.25.6.1 10.25.6.126	10.25.1.127
Switch 6	10.25.6.128	/26	10.25.6.129 10.25.6.190	10.25.191
Switch 5	10.25.6.192	/28	10.25.6.193 10.25.6.206	10.25.6.207
R8→R7	10.25.6.208	/30	10.25.6.209 10.25.6.210	10.25.6.211

BIBLIOTECA

Nombre de red	Direccion base	Mascara	Rango Asignable	Broadcast
Secretaría (SW3)	10.25.5.0	/27	10.25.5.1 - 10.25.5.30	10.25.5.31
WLAN(p.baja) (R6)	10.25.4.128	/25	10.25.4.129 - 10.25.4.254	10.25.4.255
WLAN(p.alta) (R6)	10.25.4.0	/25	10.25.4.1 - 10.25.4.126	10.25.4.127
Centro de cómputo (SW4)	10.25.5.32	/27	10.25.5.33 - 10.25.5.62	10.25.5.63
R5→R6	10.25.5.64	/30	10.25.5.65 - 10.25.5.66	10.25.5.67

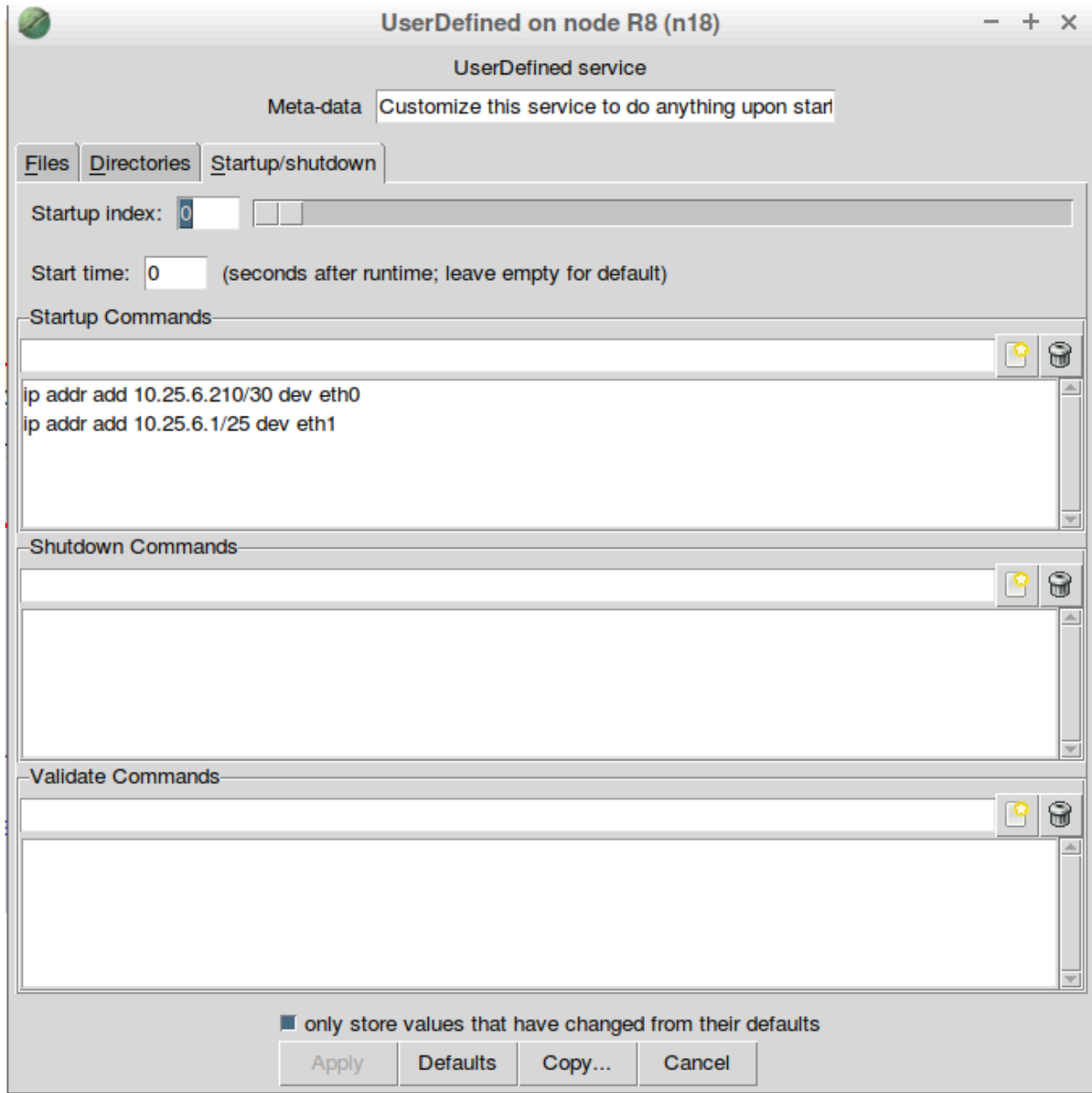
EJERCICIO 1.f)



EJERCICIO 1.g) Configure las interfaces de los router colocando su IP y Máscara.

Una vez creada la red en Vcore, asignamos las ip y máscaras correspondientes ingresando en cada router y host a la pestaña de servicios, UserDefined, Startup/Shutdown commands y agregando el siguiente comando.

ip addr add (número de ip/máscara) **dev** (nombre de la interfaz correspondiente, eth0 por ejemplo) .



EJERCICIO 1 .h) Compruebe el funcionamiento interno de cada red y la asignación de IPs utilizando los comandos ping, traceroute y la aplicación wireshark. Adjunte capturas y explique lo que comprueba en cada caso. Explique cómo se usa cada comando.

Ping: si el ping se envía correctamente verificamos que los dispositivos están bien conectados y pueden enviar y recibir paquetes

Traceroute: comprobamos si la ruta que toman los paquetes al ser enviados es correcta y cuál es el recorrido

Wireshark: comprobamos fuente y destino del paquete, que protocolo es y si es request o reply

Comedor. (Pc7 -> Router7)

Ping R7 (10.25.6.129) a Pc7 (10.25.6.130)

```
vcmd
Archivo  Editar  Pestañas  Ayuda
root@PC7:/tmp/pycore.39749/PC7.conf# ping 10.25.6.129
PING 10.25.6.129 (10.25.6.129) 56(84) bytes of data.
64 bytes from 10.25.6.129: icmp_seq=1 ttl=64 time=0.049 ms
64 bytes from 10.25.6.129: icmp_seq=2 ttl=64 time=0.060 ms
64 bytes from 10.25.6.129: icmp_seq=3 ttl=64 time=0.078 ms
64 bytes from 10.25.6.129: icmp_seq=4 ttl=64 time=0.063 ms
64 bytes from 10.25.6.129: icmp_seq=5 ttl=64 time=0.063 ms
64 bytes from 10.25.6.129: icmp_seq=6 ttl=64 time=0.064 ms
64 bytes from 10.25.6.129: icmp_seq=7 ttl=64 time=0.072 ms
64 bytes from 10.25.6.129: icmp_seq=8 ttl=64 time=0.083 ms
64 bytes from 10.25.6.129: icmp_seq=9 ttl=64 time=0.127 ms
64 bytes from 10.25.6.129: icmp_seq=10 ttl=64 time=0.150 ms
64 bytes from 10.25.6.129: icmp_seq=11 ttl=64 time=0.064 ms
64 bytes from 10.25.6.129: icmp_seq=12 ttl=64 time=0.067 ms
```

Se hace un ping request desde 10.25.6.129(origen) a 10.25.6.130(destino) y este le contesta con un ping reply desde la 10.25.6.130 a 10.25.6.129

No.	Time	Source	Destination	Protocol	Length	Info
24	9.247741774	00:00:00_aa:00:17	00:00:00_aa:00:16	ARP	42	10.25.6.130 is at 00:00:00_aa:00:17
25	10.240952643	10.25.6.130	10.25.6.129	ICMP	98	Echo (ping) request id=0x0039, seq=53/13568, ttl=64 (reply in 26)
26	10.240884448	10.25.6.129	10.25.6.130	ICMP	98	Echo (ping) reply id=0x0039, seq=53/13568, ttl=64 (request in 25)
27	11.264886687	10.25.6.130	10.25.6.129	ICMP	98	Echo (ping) request id=0x0039, seq=54/13824, ttl=64 (reply in 28)
28	11.265051402	10.25.6.129	10.25.6.130	ICMP	98	Echo (ping) reply id=0x0039, seq=54/13824, ttl=64 (request in 27)
29	12.287698436	10.25.6.130	10.25.6.129	ICMP	98	Echo (ping) request id=0x0039, seq=55/14080, ttl=64 (reply in 30)
30	12.287775974	10.25.6.129	10.25.6.130	ICMP	98	Echo (ping) reply id=0x0039, seq=55/14080, ttl=64 (request in 29)
31	13.311955711	10.25.6.130	10.25.6.129	ICMP	98	Echo (ping) request id=0x0039, seq=56/14336, ttl=64 (reply in 32)
32	13.312016588	10.25.6.129	10.25.6.130	ICMP	98	Echo (ping) reply id=0x0039, seq=56/14336, ttl=64 (request in 31)
33	14.335645157	10.25.6.130	10.25.6.129	ICMP	98	Echo (ping) request id=0x0039, seq=57/14592, ttl=64 (reply in 34)
34	14.335678748	10.25.6.129	10.25.6.130	ICMP	98	Echo (ping) reply id=0x0039, seq=57/14592, ttl=64 (request in 33)
35	15.359769425	10.25.6.130	10.25.6.129	ICMP	98	Echo (ping) request id=0x0039, seq=58/14848, ttl=64 (reply in 36)
36	15.359797326	10.25.6.129	10.25.6.130	ICMP	98	Echo (ping) reply id=0x0039, seq=58/14848, ttl=64 (request in 35)
37	16.384371818	10.25.6.130	10.25.6.129	ICMP	98	Echo (ping) request id=0x0039, seq=59/15104, ttl=64 (reply in 38)
38	16.384424941	10.25.6.129	10.25.6.130	ICMP	98	Echo (ping) reply id=0x0039, seq=59/15104, ttl=64 (request in 37)

TraceRoute Router7(10.25.6.130) hacia Pc7(10.25.6.129)

```
vcmd
Archivo  Editar  Pestañas  Ayuda
root@PC7:/tmp/pycore.39749/PC7.conf# traceroute 10.25.6.130
traceroute to 10.25.6.130 (10.25.6.130), 30 hops max, 60 byte packets
 1 10.25.6.130 (10.25.6.130) 0.023 ms 0.004 ms 0.003 ms
root@PC7:/tmp/pycore.39749/PC7.conf#
```

Biblioteca

Ping Server A(10.25.5.1) a Router5(10.25.5.2)

```
vcmd
Archivo  Editar  Pestañas  Ayuda
root@Server-A:/tmp/pycore.39749/Server-A.conf# ping 10.25.5.1
PING 10.25.5.1 (10.25.5.1) 56(84) bytes of data.
64 bytes from 10.25.5.1: icmp_seq=1 ttl=64 time=0.081 ms
64 bytes from 10.25.5.1: icmp_seq=2 ttl=64 time=0.117 ms
64 bytes from 10.25.5.1: icmp_seq=3 ttl=64 time=0.079 ms
64 bytes from 10.25.5.1: icmp_seq=4 ttl=64 time=0.126 ms
64 bytes from 10.25.5.1: icmp_seq=5 ttl=64 time=0.127 ms
64 bytes from 10.25.5.1: icmp_seq=6 ttl=64 time=0.092 ms
```

Se hace un ping request desde 10.25.5.1(origen) a 10.25.5.2(destino) y este le contesta con un ping reply desde la 10.25.5.2 a 10.25.5.1

Capturing from veth1c.0.de

No.	Time	Source	Destination	Protocol	Length	Info
27	12.288364531	10.25.5.1	10.25.5.2	ICMP	98	Echo (ping) reply id=0x001e, seq=46/11776, ttl=64 (request in 26)
28	13.312147827	10.25.5.2	10.25.5.1	ICMP	98	Echo (ping) request id=0x001e, seq=47/12832, ttl=64 (reply in 29)
29	13.312236528	10.25.5.1	10.25.5.2	ICMP	98	Echo (ping) reply id=0x001e, seq=47/12832, ttl=64 (request in 28)
30	14.335977467	10.25.5.2	10.25.5.1	ICMP	98	Echo (ping) request id=0x001e, seq=48/12288, ttl=64 (reply in 31)
31	14.336915369	10.25.5.1	10.25.5.2	ICMP	98	Echo (ping) reply id=0x001e, seq=48/12288, ttl=64 (request in 30)
32	15.360424604	10.25.5.2	10.25.5.1	ICMP	98	Echo (ping) request id=0x001e, seq=49/12544, ttl=64 (reply in 33)
33	15.360466531	10.25.5.1	10.25.5.2	ICMP	98	Echo (ping) reply id=0x001e, seq=49/12544, ttl=64 (request in 32)
34	16.394273233	10.25.5.2	10.25.5.1	ICMP	98	Echo (ping) request id=0x001e, seq=50/12800, ttl=64 (reply in 35)
35	16.384324793	10.25.5.1	10.25.5.2	ICMP	98	Echo (ping) reply id=0x001e, seq=50/12800, ttl=64 (request in 34)
36	17.408196255	10.25.5.2	10.25.5.1	ICMP	98	Echo (ping) request id=0x001e, seq=51/13056, ttl=64 (reply in 37)
37	17.408240255	10.25.5.1	10.25.5.2	ICMP	98	Echo (ping) reply id=0x001e, seq=51/13056, ttl=64 (request in 36)
38	17.535854593	00:00:00:aa:00:21	00:00:00:aa:00:1b	ARP	42	Who has 10.25.5.1? Tell 10.25.5.2
40	18.432435915	10.25.5.2	10.25.5.1	ICMP	98	Echo (ping) request id=0x001e, seq=52/13312, ttl=64 (reply in 41)
41	18.432494037	10.25.5.1	10.25.5.2	ICMP	98	Echo (ping) reply id=0x001e, seq=52/13312, ttl=64 (request in 40)

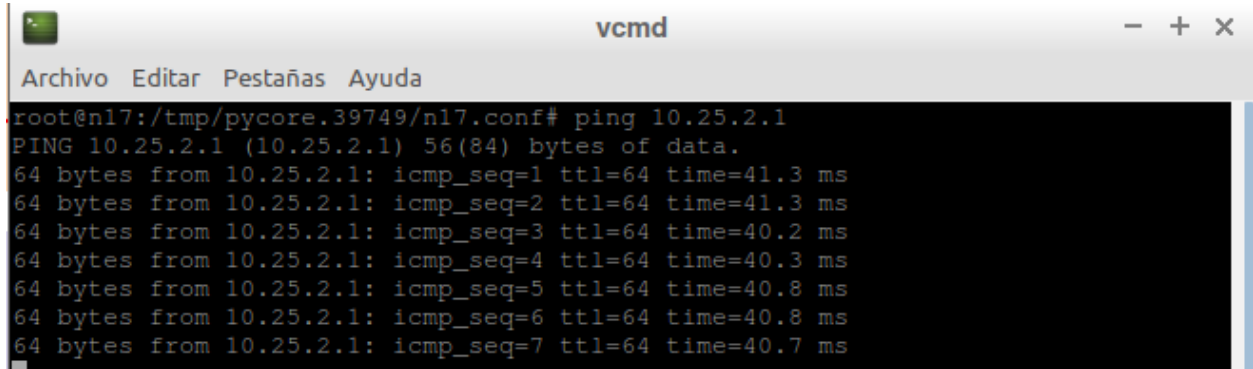
▶ Frame 1: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0
▶ Ethernet II, Src: 00:00:00:aa:00:21 (00:00:00:aa:00:21), Dst: 00:00:00:aa:00:1b (00:00:00:aa:00:1b)
▶ Internet Protocol Version 4, Src: 10.25.5.2, Dst: 10.25.5.1
▶ Internet Control Message Protocol

TraceRoute Router5(10.25.5.1) hacia Pc7(10.25.5.2)

```
vcmd
Archivo  Editar  Pestañas  Ayuda
root@Server-A:/tmp/pycore.39749/Server-A.conf# traceroute 10.25.5.2
traceroute to 10.25.5.2 (10.25.5.2), 30 hops max, 60 byte packets
 1 10.25.5.2 (10.25.5.2) 0.026 ms 0.004 ms 0.003 ms
root@Server-A:/tmp/pycore.39749/Server-A.conf#
```

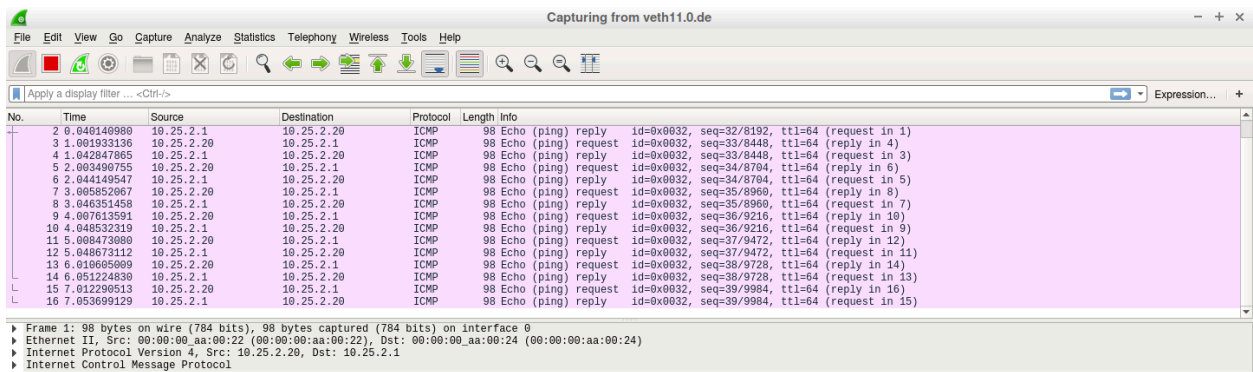

Aulas Comunes

Ping Router10(10.25.2.1) a n17(10.25.2.20)



```
root@n17:/tmp/pycore.39749/n17.conf# ping 10.25.2.1
PING 10.25.2.1 (10.25.2.1) 56(84) bytes of data.
64 bytes from 10.25.2.1: icmp_seq=1 ttl=64 time=41.3 ms
64 bytes from 10.25.2.1: icmp_seq=2 ttl=64 time=41.3 ms
64 bytes from 10.25.2.1: icmp_seq=3 ttl=64 time=40.2 ms
64 bytes from 10.25.2.1: icmp_seq=4 ttl=64 time=40.3 ms
64 bytes from 10.25.2.1: icmp_seq=5 ttl=64 time=40.8 ms
64 bytes from 10.25.2.1: icmp_seq=6 ttl=64 time=40.8 ms
64 bytes from 10.25.2.1: icmp_seq=7 ttl=64 time=40.7 ms
```

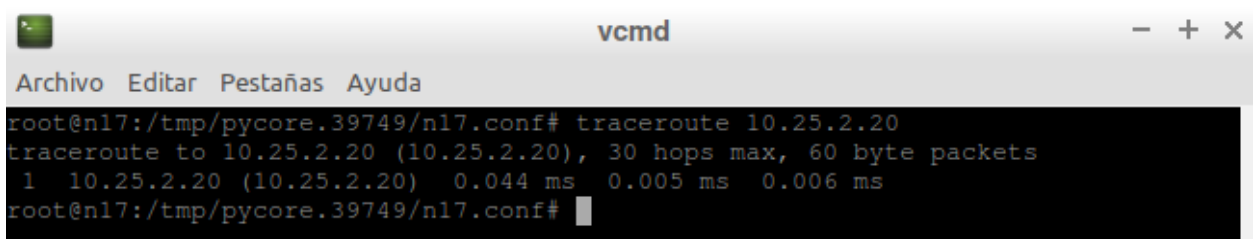
Se hace un ping request desde 10.25.2.1(origen) a 10.25.2.20(destino) y este le contesta con un ping reply desde la 10.25.2.20 a 10.25.2.1



No.	Time	Source	Destination	Protocol	Length	Info
2	0.646146980	10.25.2.1	10.25.2.20	ICMP	98	Echo (ping) request id=0x0032, seq=32/8192, ttl=64 (request in 1)
3	1.001933136	10.25.2.20	10.25.2.1	ICMP	98	Echo (ping) reply id=0x0032, seq=33/8448, ttl=64 (reply in 4)
4	1.042847865	10.25.2.1	10.25.2.20	ICMP	98	Echo (ping) request id=0x0032, seq=33/8448, ttl=64 (request in 3)
5	2.003490755	10.25.2.20	10.25.2.1	ICMP	98	Echo (ping) reply id=0x0032, seq=34/8704, ttl=64 (reply in 6)
6	2.044149547	10.25.2.1	10.25.2.20	ICMP	98	Echo (ping) request id=0x0032, seq=34/8704, ttl=64 (request in 5)
7	3.005852967	10.25.2.20	10.25.2.1	ICMP	98	Echo (ping) reply id=0x0032, seq=35/8960, ttl=64 (reply in 8)
8	3.046351458	10.25.2.1	10.25.2.20	ICMP	98	Echo (ping) request id=0x0032, seq=35/8960, ttl=64 (request in 7)
9	4.007613591	10.25.2.20	10.25.2.1	ICMP	98	Echo (ping) reply id=0x0032, seq=36/9216, ttl=64 (reply in 10)
10	4.048532319	10.25.2.1	10.25.2.20	ICMP	98	Echo (ping) request id=0x0032, seq=36/9216, ttl=64 (request in 9)
11	5.008473980	10.25.2.20	10.25.2.1	ICMP	98	Echo (ping) reply id=0x0032, seq=37/9472, ttl=64 (reply in 12)
12	5.048673112	10.25.2.1	10.25.2.20	ICMP	98	Echo (ping) request id=0x0032, seq=37/9472, ttl=64 (request in 11)
13	6.010605009	10.25.2.20	10.25.2.1	ICMP	98	Echo (ping) reply id=0x0032, seq=38/9728, ttl=64 (reply in 14)
14	6.051224830	10.25.2.1	10.25.2.20	ICMP	98	Echo (ping) request id=0x0032, seq=38/9728, ttl=64 (request in 13)
15	7.012290513	10.25.2.20	10.25.2.1	ICMP	98	Echo (ping) reply id=0x0032, seq=39/9984, ttl=64 (reply in 16)
16	7.053699129	10.25.2.1	10.25.2.20	ICMP	98	Echo (ping) request id=0x0032, seq=39/9984, ttl=64 (request in 15)

► Frame 1: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0
► Ethernet II, Src: 00:00:00:aa:00:22 (00:00:00:aa:00:22), Dst: 00:00:00:aa:00:24 (00:00:00:aa:00:24)
► Internet Protocol Version 4, Src: 10.25.2.20, Dst: 10.25.2.1
► Internet Control Message Protocol

TraceRoute n17(10.25.2.20) hacia Router10(10.25.2.1)



```
root@n17:/tmp/pycore.39749/n17.conf# traceroute 10.25.2.20
traceroute to 10.25.2.20 (10.25.2.20), 30 hops max, 60 byte packets
 1 10.25.2.20 (10.25.2.20) 0.044 ms 0.005 ms 0.006 ms
root@n17:/tmp/pycore.39749/n17.conf#
```

R3 a R4

Ping Router3(10.25.7.130) a Router4(10.25.7.129)

```
root@R3:/tmp/pycore.43255/R3.conf# ping 10.25.7.129
PING 10.25.7.129 (10.25.7.129) 56(84) bytes of data.
64 bytes from 10.25.7.129: icmp_seq=1 ttl=64 time=0.044 ms
64 bytes from 10.25.7.129: icmp_seq=2 ttl=64 time=0.037 ms
64 bytes from 10.25.7.129: icmp_seq=3 ttl=64 time=0.040 ms
64 bytes from 10.25.7.129: icmp_seq=4 ttl=64 time=0.035 ms
64 bytes from 10.25.7.129: icmp_seq=5 ttl=64 time=0.031 ms
64 bytes from 10.25.7.129: icmp_seq=6 ttl=64 time=0.052 ms
^C64 bytes from 10.25.7.129: icmp_seq=7 ttl=64 time=0.035 ms
64 bytes from 10.25.7.129: icmp_seq=8 ttl=64 time=0.035 ms
^C
--- 10.25.7.129 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7149ms
rtt min/avg/max/mdev = 0.031/0.038/0.052/0.009 ms
root@R3:/tmp/pycore.43255/R3.conf#
```

Se hace un ping request desde 10.25.7.130(origen) a 10.25.7.129(destino) y este le contesta con un ping reply desde la 10.25.7.129 a 10.25.7.130.

No.	Time	Source	Destination	Protocol	Length	Info
14	6.143881181	10.25.7.129	10.25.7.130	ICMP	98	Echo (ping) reply id=0x003d, seq=19/4864, ttl=64 (request in 13)
15	7.167985664	10.25.7.130	10.25.7.129	ICMP	98	Echo (ping) request id=0x003d, seq=20/5120, ttl=64 (reply in 16)
16	7.168004754	10.25.7.129	10.25.7.130	ICMP	98	Echo (ping) reply id=0x003d, seq=20/5120, ttl=64 (request in 15)
17	8.191807597	10.25.7.130	10.25.7.129	ICMP	98	Echo (ping) request id=0x003d, seq=21/5376, ttl=64 (reply in 18)
18	8.191827527	10.25.7.129	10.25.7.130	ICMP	98	Echo (ping) reply id=0x003d, seq=21/5376, ttl=64 (request in 17)
19	9.216023460	10.25.7.130	10.25.7.129	ICMP	98	Echo (ping) request id=0x003d, seq=22/5632, ttl=64 (reply in 20)
20	9.216044330	10.25.7.129	10.25.7.130	ICMP	98	Echo (ping) reply id=0x003d, seq=22/5632, ttl=64 (request in 19)
21	10.239710523	10.25.7.130	10.25.7.129	ICMP	98	Echo (ping) request id=0x003d, seq=23/5888, ttl=64 (reply in 22)
22	10.239729053	10.25.7.129	10.25.7.130	ICMP	98	Echo (ping) reply id=0x003d, seq=23/5888, ttl=64 (request in 21)
23	11.263733967	10.25.7.130	10.25.7.129	ICMP	98	Echo (ping) request id=0x003d, seq=24/6144, ttl=64 (reply in 24)
24	11.263751027	10.25.7.129	10.25.7.130	ICMP	98	Echo (ping) reply id=0x003d, seq=24/6144, ttl=64 (request in 23)
25	11.999826514	fe80::3413:e6ff:feff::2	ff02::2	ICMPv6	70	Router Solicitation from d2:06:e0:59:dd:4f
26	12.288011534	10.25.7.130	10.25.7.129	ICMP	98	Echo (ping) request id=0x003d, seq=25/6400, ttl=64 (reply in 27)
27	12.288031644	10.25.7.129	10.25.7.130	ICMP	98	Echo (ping) reply id=0x003d, seq=25/6400, ttl=64 (request in 26)
28	13.311719590	10.25.7.130	10.25.7.129	ICMP	98	Echo (ping) request id=0x003d, seq=26/6656, ttl=64 (reply in 29)
29	13.311738559	10.25.7.129	10.25.7.130	ICMP	98	Echo (ping) reply id=0x003d, seq=26/6656, ttl=64 (request in 28)
30	14.335781517	10.25.7.130	10.25.7.129	ICMP	98	Echo (ping) request id=0x003d, seq=27/6912, ttl=64 (reply in 31)
31	14.335803487	10.25.7.129	10.25.7.130	ICMP	98	Echo (ping) reply id=0x003d, seq=27/6912, ttl=64 (request in 30)
32	15.359759194	10.25.7.130	10.25.7.129	ICMP	98	Echo (ping) request id=0x003d, seq=28/7168, ttl=64 (reply in 33)
33	15.359778074	10.25.7.129	10.25.7.130	ICMP	98	Echo (ping) reply id=0x003d, seq=28/7168, ttl=64 (request in 32)
34	16.383727450	10.25.7.130	10.25.7.129	ICMP	98	Echo (ping) request id=0x003d, seq=29/7424, ttl=64 (reply in 35)
35	16.383746807	10.25.7.129	10.25.7.130	ICMP	98	Echo (ping) reply id=0x003d, seq=29/7424, ttl=64 (request in 34)

TraceRoute Router3(10.25.7.130) hacia Router4(10.25.7.129)

```
vcmd
Archivo Editar Pestañas Ayuda
root@R3:/tmp/pycore.34559/R3.conf# traceroute 10.25.7.129
traceroute to 10.25.7.129 (10.25.7.129), 30 hops max, 60 byte packets
 1 10.25.7.129 (10.25.7.129) 0.048 ms 0.005 ms 0.005 ms
root@R3:/tmp/pycore.34559/R3.conf#
```

Datcenter

Ping Server2(10.25.0.5) a Server1(10.25.0.4)

```
root@Server2:/tmp/pycore.42345/Server2.conf# ping 10.25.0.4
PING 10.25.0.4 (10.25.0.4) 56(84) bytes of data.
64 bytes from 10.25.0.4: icmp_seq=1 ttl=64 time=0.028 ms
64 bytes from 10.25.0.4: icmp_seq=2 ttl=64 time=0.044 ms
64 bytes from 10.25.0.4: icmp_seq=3 ttl=64 time=0.040 ms
64 bytes from 10.25.0.4: icmp_seq=4 ttl=64 time=0.066 ms
64 bytes from 10.25.0.4: icmp_seq=5 ttl=64 time=0.041 ms
64 bytes from 10.25.0.4: icmp_seq=6 ttl=64 time=0.045 ms
64 bytes from 10.25.0.4: icmp_seq=7 ttl=64 time=0.040 ms
64 bytes from 10.25.0.4: icmp_seq=8 ttl=64 time=0.042 ms
```

Se hace un ping request desde 10.25.0.5(origen) a 10.25.0.4(destino) y este le contesta con un ping reply desde la 10.25.0.5 a 10.25.0.4.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.25.0.5	10.25.0.4	ICMP	98	Echo (ping) request id=0x0029, seq=36/9216, ttl=64 (reply in 2)
2	0.000020930	10.25.0.4	10.25.0.5	ICMP	98	Echo (ping) reply id=0x0029, seq=36/9216, ttl=64 (request in 1)
3	0.223741050	10.25.0.5	10.25.0.4	ICMP	98	Echo (ping) request id=0x001e, seq=109/27904, ttl=64 (reply in 4)
4	0.223760770	10.25.0.4	10.25.0.5	ICMP	98	Echo (ping) reply id=0x001e, seq=109/27904, ttl=64 (request in 3)
5	1.023960676	10.25.0.5	10.25.0.4	ICMP	98	Echo (ping) request id=0x0029, seq=37/9472, ttl=64 (reply in 6)
6	1.023981056	10.25.0.4	10.25.0.5	ICMP	98	Echo (ping) reply id=0x0029, seq=37/9472, ttl=64 (request in 5)
7	1.247746213	10.25.0.5	10.25.0.4	ICMP	98	Echo (ping) request id=0x001e, seq=110/28160, ttl=64 (reply in 8)
8	1.247767172	10.25.0.4	10.25.0.5	ICMP	98	Echo (ping) reply id=0x001e, seq=110/28160, ttl=64 (request in 7)
9	2.048132743	10.25.0.5	10.25.0.4	ICMP	98	Echo (ping) request id=0x0029, seq=38/9728, ttl=64 (reply in 10)
10	2.048154313	10.25.0.4	10.25.0.5	ICMP	98	Echo (ping) reply id=0x0029, seq=38/9728, ttl=64 (request in 9)
11	2.271848767	10.25.0.5	10.25.0.4	ICMP	98	Echo (ping) request id=0x001e, seq=111/28416, ttl=64 (reply in 12)
12	2.271878677	10.25.0.4	10.25.0.5	ICMP	98	Echo (ping) reply id=0x001e, seq=111/28416, ttl=64 (request in 11)
13	3.071761728	10.25.0.5	10.25.0.4	ICMP	98	Echo (ping) request id=0x0029, seq=39/9984, ttl=64 (reply in 14)
14	3.071784438	10.25.0.4	10.25.0.5	ICMP	98	Echo (ping) reply id=0x0029, seq=39/9984, ttl=64 (request in 13)
15	3.295766384	10.25.0.5	10.25.0.4	ICMP	98	Echo (ping) request id=0x001e, seq=112/28672, ttl=64 (reply in 16)
16	3.295787264	10.25.0.4	10.25.0.5	ICMP	98	Echo (ping) reply id=0x001e, seq=112/28672, ttl=64 (request in 15)
17	4.095867913	10.25.0.5	10.25.0.4	ICMP	98	Echo (ping) request id=0x0029, seq=40/10240, ttl=64 (reply in 18)
18	4.095889883	10.25.0.4	10.25.0.5	ICMP	98	Echo (ping) reply id=0x0029, seq=40/10240, ttl=64 (request in 17)
19	4.319891824	10.25.0.5	10.25.0.4	ICMP	98	Echo (ping) request id=0x001e, seq=113/28928, ttl=64 (reply in 20)
20	4.319937833	10.25.0.4	10.25.0.5	ICMP	98	Echo (ping) reply id=0x001e, seq=113/28928, ttl=64 (request in 19)
21	5.119922092	10.25.0.5	10.25.0.4	ICMP	98	Echo (ping) request id=0x0029, seq=41/10496, ttl=64 (reply in 22)
22	5.119945482	10.25.0.4	10.25.0.5	ICMP	98	Echo (ping) reply id=0x0029, seq=41/10496, ttl=64 (request in 21)

TraceRoute Server2(10.25.0.5) a Server1(10.25.0.4)

```
root@Server2:/tmp/pycore.42345/Server2.conf# traceroute 10.25.0.4
traceroute to 10.25.0.4 (10.25.0.4), 30 hops max, 60 byte packets
 1 10.25.0.4 (10.25.0.4) 0.037 ms 0.008 ms 0.005 ms
root@Server2:/tmp/pycore.42345/Server2.conf#
```

PARTE 2

Ejercicio 2 .a)

Biblioteca				
Router	Red destino	IP sig router	Interfaz salida	Directa/ indirecta
R5	10.25.7.0/25	-	eth0	D
	10.25.5.0/27	-	eth1	D
	10.25.5.32/27	-	eth2	D
	10.25.5.64/30	-	eth3	D
	10.25.4.0/24	10.25.5.66	eth3	I
	10.25.2.0/23	10.25.7.2	eth0	I
	10.25.6.0/24	10.25.7.4	eth0	I
	DEFAULT	10.25.7.1	eth0	I
R6	10.25.4.0/25	-	eth1	D
	10.25.4.128/25	-	eth0	D
	10.25.5.64/30	-	eth2	D
	DEFAULT	10.25.5.65	eth2	I

R5 (Comando ip route show)

```

root@R5:/tmp/pycore.45635/R5.conf# ip route show
default via 10.25.7.1 dev eth0
10.25.2.0/23 via 10.25.7.2 dev eth0
10.25.4.0/25 via 10.25.5.66 dev eth3
10.25.4.128/25 via 10.25.5.66 dev eth3
10.25.5.0/27 dev eth1 proto kernel scope link src 10.25.5.1
10.25.5.32/27 dev eth2 proto kernel scope link src 10.25.5.33
10.25.5.64/30 dev eth3 proto kernel scope link src 10.25.5.65
10.25.6.0/24 via 10.25.7.4 dev eth0
10.25.7.0/25 dev eth0 proto kernel scope link src 10.25.7.3

```

R6 (Comando ip route show)

```
root@R6:/tmp/pycore.45635/R6.conf# ip route show
default via 10.25.5.65 dev eth2
10.25.4.0/25 dev eth1 proto kernel scope link src 10.25.4.1
10.25.4.128/25 dev eth0 proto kernel scope link src 10.25.4.129
10.25.5.64/30 dev eth2 proto kernel scope link src 10.25.5.66
```

Minimización por sumalización (R5-Biblioteca)

10.25.4.0/25 (4.0 – 4.127) 10.25.4.128/25 (4.128 – 4.255) (128 direcciones + 128 direcciones = 256 direcciones = /24) 10.25.4.0/24 (4.0 – 4.255)

10.25.4.0/25	10.25.4.1	eth3	I
10.25.4.128/25	10.25.4.129	eth3	I
10.25.4.0/24	10.25.5.66	eth3	I

Comedor				
Router	Red destino	IP sig router	Interfaz salida	Directa/ indirecta
R7	10.25.6.208/30	-	eth3	D
	10.25.6.128/26	-	eth2	D
	10.25.6.192/28	-	eth1	D
	10.25.7.0/25	-	eth0	D
	10.25.6.0/25	10.25.6.210	eth3	I
	10.25.2.0/23	10.25.7.2	eth0	I
	10.25.4.0/23	10.25.7.3	eth0	I
	DEFAULT	10.25.7.1	eth0	I
R8	10.25.6.208/30	-	eth1	D
	10.25.6.0/25	-	eth0	D

	DEFAULT	10.25.6.209	eth1	I
--	---------	-------------	------	---

R7 (Comando ip route show)

```

root@R7:/tmp/pycore.35611/R7.conf# ip route show
default via 10.25.7.1 dev eth0
10.25.2.0/23 via 10.25.7.2 dev eth0
10.25.4.0/23 via 10.25.7.3 dev eth0
10.25.6.0/25 via 10.25.6.210 dev eth3
10.25.6.128/26 dev eth2 proto kernel scope link src 10.25.6.129
10.25.6.192/28 dev eth1 proto kernel scope link src 10.25.6.193
10.25.6.208/30 dev eth3 proto kernel scope link src 10.25.6.209
10.25.7.0/25 dev eth0 proto kernel scope link src 10.25.7.4
root@R7:/tmp/pycore.35611/R7.conf# █

```

R8 (Comando ip route show)

```

root@R8:/tmp/pycore.35611/R8.conf# ip route show
default via 10.25.6.209 dev eth1
10.25.6.0/25 dev eth0 proto kernel scope link src 10.25.6.1
10.25.6.208/30 dev eth1 proto kernel scope link src 10.25.6.210
root@R8:/tmp/pycore.35611/R8.conf# █

```

Aulas Comunes				
Router	Red destino	IP sig router	Interfaz salida	Directa/ indirecta
R9	10.25.3.128/26	-	eth2	D
	10.25.3.192/30	-	eth2	D
	10.25.7.0/25	-	eth1	D
	10.25.2.0/24	10.25.3.194	eth0	I
	10.25.3.0/25	10.25.3.1	eth0	I
	10.25.4.0/23	10.25.7.3	eth1	I
	10.25.6.0/24	10.25.7.4	eth1	I
	DEFAULT	10.25.7.1	eth1	I

Aulas Comunes				
Router	Red destino	IP sig router	Interfaz salida	Directa/ indirecta
R10	10.25.2.0/25	-	eth1	D
	10.25.2.128/25	-	eth2	D
	10.25.3.0/25	-	eth3	D
	10.25.3.194/30	-	eth0	D
	DEFAULT	10.25.3.192	eth0	I

R9 (Comando ip route show)

```
root@R9:/tmp/pycore.45635/R9.conf# ip route show
default via 10.25.7.1 dev eth1
10.25.2.0/25 via 10.25.3.194 dev eth0
10.25.2.128/25 via 10.25.3.194 dev eth0
10.25.3.0/25 via 10.25.3.194 dev eth0
10.25.3.128/26 dev eth2 proto kernel scope link src 10.25.3.129
10.25.3.192/30 dev eth0 proto kernel scope link src 10.25.3.193
10.25.4.0/23 via 10.25.7.3 dev eth1
10.25.6.0/24 via 10.25.7.4 dev eth1
10.25.7.0/25 dev eth1 proto kernel scope link src 10.25.7.2
```

R10(Comando ip route show)

```
root@R10:/tmp/pycore.45635/R10.conf# ip route show
default via 10.25.3.193 dev eth0
10.25.2.0/25 dev eth1 proto kernel scope link src 10.25.2.1
10.25.2.128/25 dev eth2 proto kernel scope link src 10.25.2.129
10.25.3.0/25 dev eth3 proto kernel scope link src 10.25.3.1
10.25.3.192/30 dev eth0 proto kernel scope link src 10.25.3.194
```

Minimizaci3n por sumarizaci3n (R9-Aulas Comunes)

10.25.2.0/25 (2.0 – 2.127) 10.25.2.128/25 (2.128 – 2.255) (128 direcciones + 128 direcciones = 256 direcciones = /24) 10.25.2.0/24 (2.0 – 2.255)

10.25.2.0/25	10.25.2.1	eth0	I
10.25.2.128/25	10.25.2.129	eth0	I
10.25.2.0/24	10.25.3.194	eth0	I

Buscamos simplemente sumarizar la red 2.0 que abarca dos bloques de 128 direcciones, aunque podríamos haber sumariado una tercera red (la 3.0) la sumariación será más amplia de lo necesario, incluyendo rangos de IP que no están en las subredes iniciales.

Datacenter				
Router	Red destino	IP sig router	Interfaz salida	Directa/ indirecta
R2	10.25.0.0/23	-	eth1	D
	201.0.0.0/24	-	eth0	D
	10.25.2.0/23	10.25.0.3	eth1	I
	10.25.4.0/23	10.25.0.3	eth1	I
	10.25.6.0/24	10.25.0.3	eth1	I
	10.25.7.0/25	10.25.0.3	eth1	I
	DEFAULT	201.0.0.1	eth0	I

R2(Comando ip route show)

```

root@R2:/tmp/pycore.38493/R2.conf# ip route show
default via 201.0.0.1 dev eth0
10.25.0.0/23 dev eth1 proto kernel scope link src 10.25.0.1
10.25.2.0/23 via 10.25.0.2 dev eth1
10.25.4.0/23 via 10.25.0.2 dev eth1
10.25.6.0/24 via 10.25.0.2 dev eth1
10.25.7.0/25 via 10.25.0.2 dev eth1
201.0.0.0/24 dev eth0 proto kernel scope link src 201.0.0.2
root@R2:/tmp/pycore.38493/R2.conf#

```


Casa				
Router	Red destino	IP sig router	Interfaz salida	Directa/ indirecta
R11	192.168.10.0/24	-	eth0	D
	201.0.2.2	-	eth1	D
	DEFAULT	201.0.2.1	eth1	I

R11(Comando ip route show)

```

root@R11:/tmp/pycore.45635/R11.conf# ip route show
default via 201.0.2.1 dev eth1
192.168.10.0/24 dev eth0 proto kernel scope link src 192.168.10.1
201.0.2.0/24 dev eth1 proto kernel scope link src 201.0.2.2

```

Punto a Punto (R4 ↔ R3)				
Router	Red destino	IP sig router	Interfaz salida	Directa/ indirecta
R4	10.25.7.128/30	-	eth0	D
	201.0.1.0/24	-	eth1	D
	10.25.2.0/23	10.25.7.130	eth0	I
	10.25.4.0/23	10.25.7.130	eth0	I
	10.25.6.0/24	10.25.7.130	eth0	I
	10.25.7.0/25	10.25.7.130	eth0	I
	DEFAULT	201.0.1.1	eth1	I
R3	10.25.7.128/30	-	eth1	D
	10.25.7.0/25	-	eth2	D

	10.25.0.0/23	-	eth0	D
	10.25.4.0/23	10.25.7.3	eth2	I
	10.25.6.0/24	10.25.7.4	eth2	I
	10.25.2.0/23	10.25.7.2	eth2	I
	DEFAULT	10.25.7.129/30	eth1	I

R3(Comando ip route show)

```
root@R3:/tmp/pycore.46301/R3.conf# ip route show
default via 10.25.7.129 dev eth1
10.25.0.0/23 dev eth0 proto kernel scope link src 10.25.0.2
10.25.2.0/23 via 10.25.7.2 dev eth2
10.25.4.0/23 via 10.25.7.3 dev eth2
10.25.6.0/24 via 10.25.7.4 dev eth2
10.25.7.0/25 dev eth2 proto kernel scope link src 10.25.7.1
10.25.7.128/30 dev eth1 proto kernel scope link src 10.25.7.130
201.0.2.0/24 via 10.25.0.1 dev eth0
root@R3:/tmp/pycore.46301/R3.conf#
```

R4(Comando ip route show)

```
root@R4:/tmp/pycore.35611/R4.conf# ip route show
default via 201.0.1.1 dev eth1
10.25.2.0/23 via 10.25.7.130 dev eth0
10.25.4.0/23 via 10.25.7.130 dev eth0
10.25.6.0/24 via 10.25.7.130 dev eth0
10.25.7.0/25 via 10.25.7.130 dev eth0
10.25.7.128/30 dev eth0 proto kernel scope link src 10.25.7.129
201.0.1.0/24 dev eth1 proto kernel scope link src 201.0.1.2
root@R4:/tmp/pycore.35611/R4.conf#
```

Ejercicio 2 .b)

Reglas aplicadas en R4 para denegar el paso de tráfico desde el campus hacia R4, exceptuando el tráfico proveniente de Aulas Comunes y Biblioteca

Iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE → Enmascaramiento por NAT para salir de la red privada(campus)

Iptables -t filter -A FORWARD -s 10.25.2.0/23 -j ACCEPT → Acepta tráfico proveniente de Aulas Comunes

Iptables -t filter -A FORWARD -s 10.25.4.0/23 -j ACCEPT → Acepta tráfico proveniente de Biblioteca

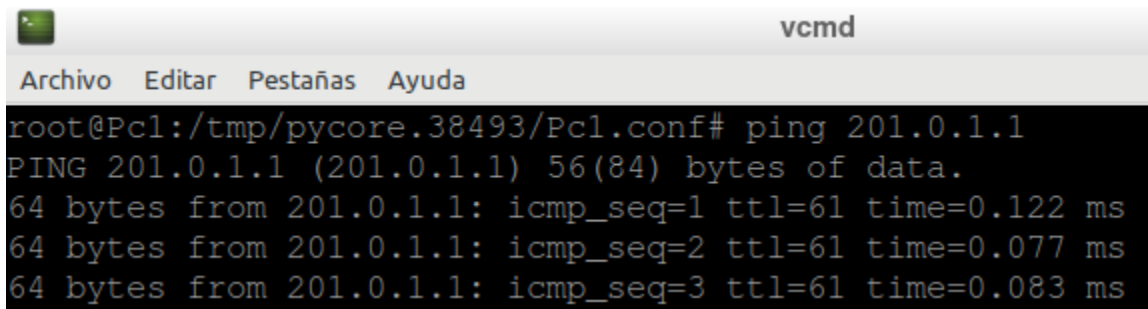
Iptables -t filter -A FORWARD -i eth0 -j REJECT

Capturas:

Comandos Cargados R4

```
iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE
iptables -t filter -A FORWARD -s 10.25.2.0/23 -j ACCEPT
iptables -t filter -A FORWARD -s 10.25.4.0/23 -j ACCEPT
iptables -t filter -A FORWARD -i eth0 -j REJECT
```

Ping desde la pc1 de biblioteca hacia internet

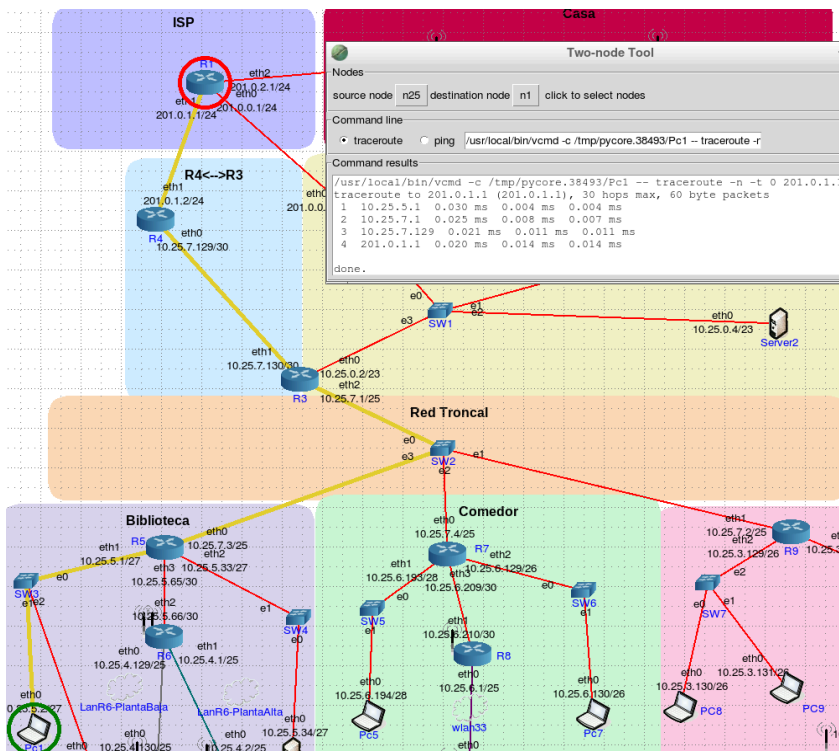


```
root@Pc1:/tmp/pycore.38493/Pc1.conf# ping 201.0.1.1
PING 201.0.1.1 (201.0.1.1) 56(84) bytes of data.
64 bytes from 201.0.1.1: icmp_seq=1 ttl=61 time=0.122 ms
64 bytes from 201.0.1.1: icmp_seq=2 ttl=61 time=0.077 ms
64 bytes from 201.0.1.1: icmp_seq=3 ttl=61 time=0.083 ms
```

Wireshark desde pc1 a internet

5	3.426618...	10.25.5.2	201.0.1.1	ICMP	98 Echo (ping) request	id=0x001c, seq=1/256, ttl=64 (reply in 6)
6	3.426694...	201.0.1.1	10.25.5.2	ICMP	98 Echo (ping) reply	id=0x001c, seq=1/256, ttl=61 (request in 5)
7	4.448119...	10.25.5.2	201.0.1.1	ICMP	98 Echo (ping) request	id=0x001c, seq=2/512, ttl=64 (reply in 8)
8	4.448176...	201.0.1.1	10.25.5.2	ICMP	98 Echo (ping) reply	id=0x001c, seq=2/512, ttl=61 (request in 7)

Two-node tool (traceroute)



Ejercicio 2 .c)

Configurar R2 para que el datacenter salga a Internet por este router. Compruebe el funcionamiento con el comando ping y la aplicación Wireshark y/o traceroute.

Para cumplir con los requisitos, agregamos una regla de filtrado que todo lo que atraviese el R2 con destino a Internet sea permitido desde datacenter, ahora bien si por ejemplo algún equipo de los pabellones quiere salir a internet saldrá por R4 y cuando quiera llegar a R2 el router intermedio (R3) va a obstaculizar ese filtrado llevándolo directamente a la default.

Para salir a internet transformará su ip privada a una ip pública gracias a NAT.

Capturas:

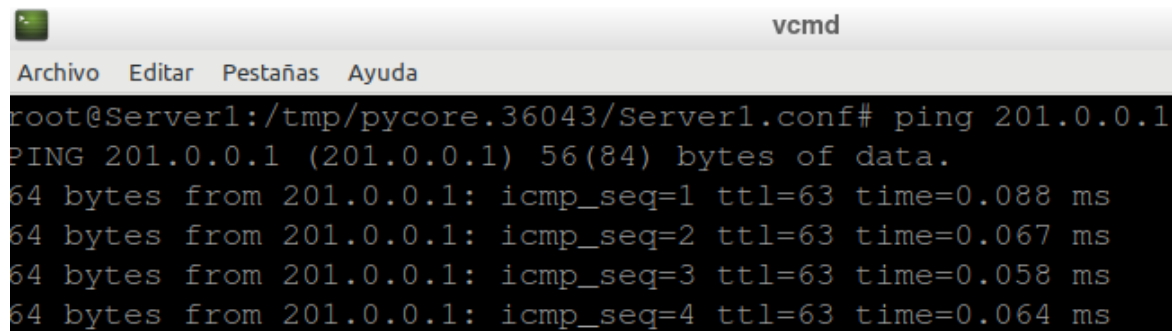
Comandos cargados en R2

```
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
iptables -t filter -A FORWARD -d 201.0.0.1 -i eth1 -j ACCEPT
iptables -t nat -A PREROUTING -p udp -i eth0 --dport 80 -j DNAT --to-destination 10.25.0.3:8080
```

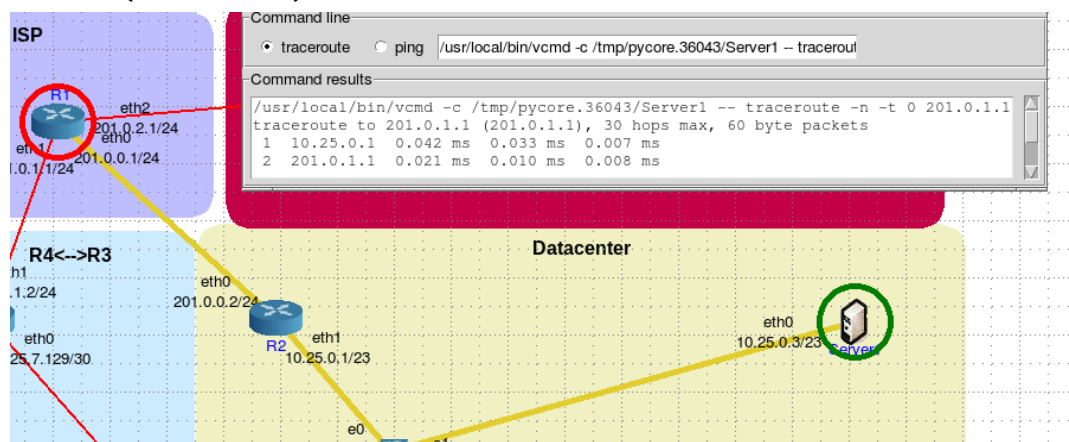
Wireshark (Datacenter -Server 1 hacia internet pasando por R2)

3	0.095870...	10.25.0.3	201.0.0.1	ICMP	98 Echo (ping) request	id=0x001b, seq=6/1536, ttl=64 (reply in 4)
4	0.095912...	201.0.0.1	10.25.0.3	ICMP	98 Echo (ping) reply	id=0x001b, seq=6/1536, ttl=63 (request in 3)
5	1.119784...	10.25.0.3	201.0.0.1	ICMP	98 Echo (ping) request	id=0x001b, seq=7/1792, ttl=64 (reply in 6)
6	1.119826...	201.0.0.1	10.25.0.3	ICMP	98 Echo (ping) reply	id=0x001b, seq=7/1792, ttl=63 (request in 5)

Ping (Server 1 hacia 201.0.0.1)



Traceroute (Two-node tool)



Ejercicio 2 .d)

Configuración en R11 para que PC-CASA pueda acceder a internet:

`iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE` → Enmascaramiento por NAT para salir de la red privada(casa)

Capturas:

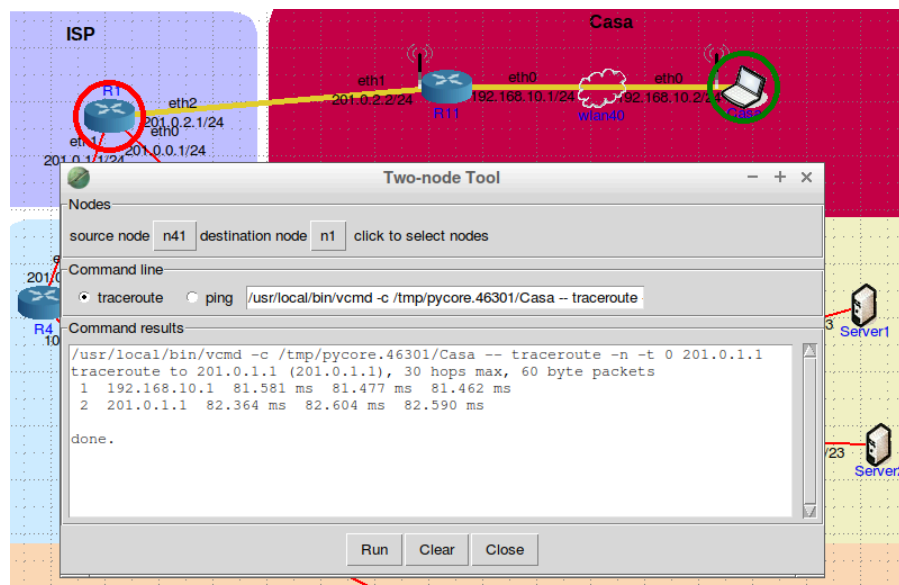
Ping desde la PC-CASA hacia internet

```
root@Casa:/tmp/pycore.46301/Casa.conf# ping 201.0.2.1
PING 201.0.2.1 (201.0.2.1) 56(84) bytes of data.
64 bytes from 201.0.2.1: icmp_seq=1 ttl=63 time=40.1 ms
64 bytes from 201.0.2.1: icmp_seq=2 ttl=63 time=40.9 ms
64 bytes from 201.0.2.1: icmp_seq=3 ttl=63 time=40.3 ms
64 bytes from 201.0.2.1: icmp_seq=4 ttl=63 time=40.7 ms
64 bytes from 201.0.2.1: icmp_seq=5 ttl=63 time=40.9 ms
64 bytes from 201.0.2.1: icmp_seq=6 ttl=63 time=40.1 ms
64 bytes from 201.0.2.1: icmp_seq=7 ttl=63 time=41.2 ms
64 bytes from 201.0.2.1: icmp_seq=8 ttl=63 time=41.6 ms
```

C

No.	Time	Source	Destination	Protocol	Length	Info
1	0.00000...	192.168.10.2	201.0.2.1	ICMP	98	Echo (ping) request id=0x001c, seq=51/13056, ttl=64 (reply in 2)
2	0.04077...	201.0.2.1	192.168.10.2	ICMP	98	Echo (ping) reply id=0x001c, seq=51/13056, ttl=63 (request in 1)
3	1.00185...	192.168.10.2	201.0.2.1	ICMP	98	Echo (ping) request id=0x001c, seq=52/13312, ttl=64 (reply in 4)
4	1.04219...	201.0.2.1	192.168.10.2	ICMP	98	Echo (ping) reply id=0x001c, seq=52/13312, ttl=63 (request in 3)
5	2.00367...	192.168.10.2	201.0.2.1	ICMP	98	Echo (ping) request id=0x001c, seq=53/13568, ttl=64 (reply in 6)
6	2.04409...	201.0.2.1	192.168.10.2	ICMP	98	Echo (ping) reply id=0x001c, seq=53/13568, ttl=63 (request in 5)
7	3.00626...	192.168.10.2	201.0.2.1	ICMP	98	Echo (ping) request id=0x001c, seq=54/13824, ttl=64 (reply in 8)
8	3.04672...	201.0.2.1	192.168.10.2	ICMP	98	Echo (ping) reply id=0x001c, seq=54/13824, ttl=63 (request in 7)

Two-node tool (traceroute)



Ejercicio 2 .e)

Configuración en R3 para dirigir por R2 el tráfico saliente del campus hacia la IP pública de R11:

Intentamos aplicar reglas en R3 para redirigir (DNAT) por R2 aquellos paquetes con dirección destino: 201.0.2.2 (IP pública R11), pero no logramos que funcionara. Una vez que hacíamos el cambio de dirección, por ejemplo a la 10.25.0.1, el paquete continuaba su recorrido a través del datacenter, llegando a R2 pero ahí mismo terminaba su viaje.

Por todo lo explicado anteriormente, y tras debatir con mi compañero, tomamos la decisión de añadir la dirección 201.0.2.2 al router R3, siendo la misma una de las alternativas discutidas durante la clase de práctica para realizar este punto.

```
ip route add 201.0.2.0/24 via 10.25.0.1 dev eth0
iptables -t nat -A PREROUTING -p udp -i eth2 --dport 80 -j DNAT --to-destination 10.25.0.3:8080
```

Ping desde la PC11 (Aulas Comunes) hacia IP pública del R11

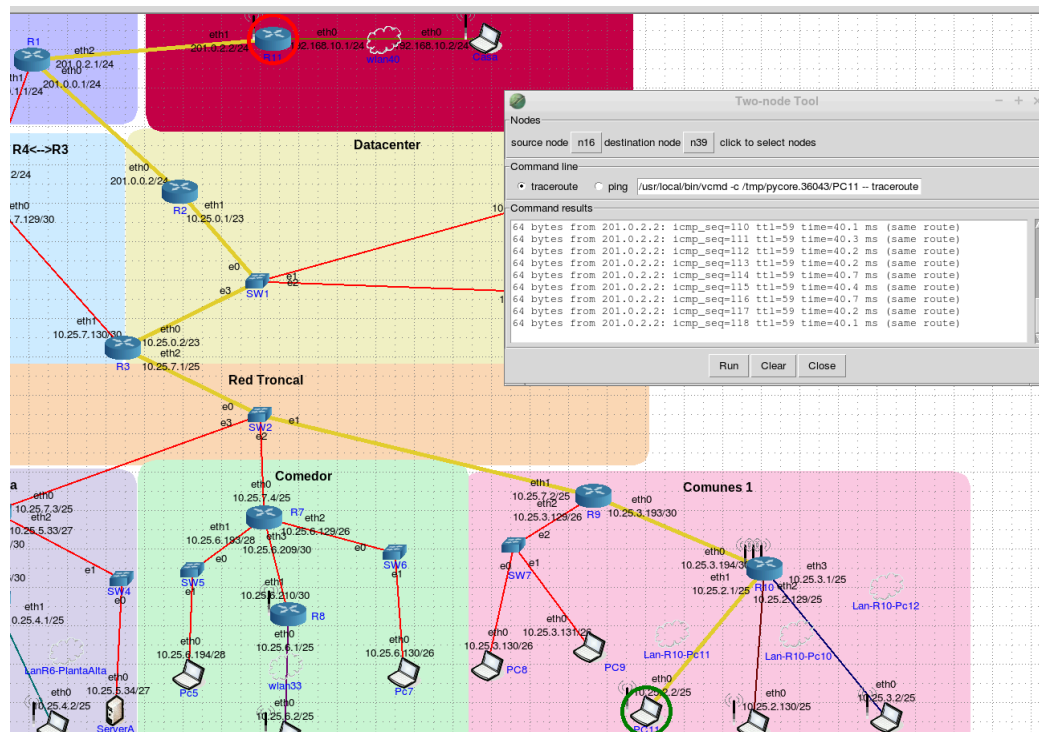
```
/usr/local/bin/vcmd -c /tmp/pycore.36043/PC11 -- ping -R -n 201.0.2.2
PING 201.0.2.2 (201.0.2.2) 56(124) bytes of data.
64 bytes from 201.0.2.2: icmp_seq=1 ttl=59 time=40.1 ms
RR:
 10.25.2.2
 10.25.3.194
 10.25.7.2
 10.25.0.2
 201.0.0.2
 201.0.2.1
 201.0.2.2
 201.0.2.2
 201.0.0.1

64 bytes from 201.0.2.2: icmp_seq=2 ttl=59 time=40.2 ms (same route)
64 bytes from 201.0.2.2: icmp_seq=3 ttl=59 time=40.4 ms (same route)
64 bytes from 201.0.2.2: icmp_seq=4 ttl=59 time=40.2 ms (same route)
64 bytes from 201.0.2.2: icmp_seq=5 ttl=59 time=40.8 ms (same route)
64 bytes from 201.0.2.2: icmp_seq=6 ttl=59 time=41.2 ms (same route)
64 bytes from 201.0.2.2: icmp_seq=7 ttl=59 time=40.1 ms (same route)
64 bytes from 201.0.2.2: icmp_seq=8 ttl=59 time=40.3 ms (same route)
64 bytes from 201.0.2.2: icmp_seq=9 ttl=59 time=40.6 ms (same route)
64 bytes from 201.0.2.2: icmp_seq=10 ttl=59 time=40.4 ms (same route)
64 bytes from 201.0.2.2: icmp_seq=11 ttl=59 time=40.7 ms (same route)
```

Wireshark PC11 (Aulas Comunes) hacia IP pública del R11

3 0.275818...	10.25.2.2	201.0.1.1	ICMP	98 Echo (ping) request	id=0x001c, seq=155/39680, ttl=64 (reply in 4)
4 0.315928...	201.0.1.1	10.25.2.2	ICMP	98 Echo (ping) reply	id=0x001c, seq=155/39680, ttl=60 (request in 3)
5 1.000429...	10.25.2.2	201.0.2.2	ICMP	138 Echo (ping) request	id=0x001d, seq=54/13824, ttl=64 (reply in 6)
6 1.040708...	201.0.2.2	10.25.2.2	ICMP	138 Echo (ping) reply	id=0x001d, seq=54/13824, ttl=59 (request in 5)

Two-node tool (traceroute)



Ejercicio 2 .f)

Configure la red de manera de poder enviar el mensaje “Hola Data Center” desde PC-Casa hasta el Server 1, utilizando Netcat. Tenga en cuenta configurar el reenvío de paquetes en el router R2. Considere que el puerto que está abierto en el router R2 es el 80, mientras que el servicio en el Server 1 está corriendo en el puerto 8080. Indique, ¿qué dirección IP y qué puertos se deben utilizar? Verifique la transformación de los paquetes que atraviesan el router R2 utilizando la aplicación de Wireshark.

Para lograr el objetivo , primero nuestro router R11 debe tener el comando MASQUERADE para salir a internet desde casa con una ip publica. Una vez logrado el cometido procedemos configurar el R2 de tal manera:

```
iptables -t nat -A PREROUTING -p <Protocolo> --dport<Puerto> -i <interfaz> -j DNAT --to <IP destino:Puerto destino>
```

¿Cual es la idea aca? Básicamente que cuando quiera enviar un mensaje desde la CASA hacia el SERVER 1 yo tengo que tener abierto el puerto 8080 (puerto escucha). Entonces cuando venga un mensaje a través del puerto 80 utilizando el protocolo udp redirigiremos el puerto con destino al 8080 que está esperando la respuesta.

Aclaraciones Netcat

Es una herramienta de la línea de comandos que se utiliza para establecer conexiones de tipo TCP y UDP entre dispositivos. para utilizarla se utilizan los siguientes comandos

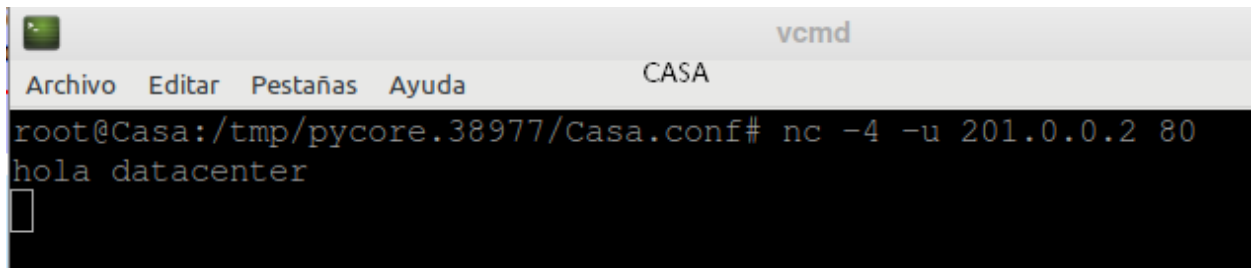
- El Receptor Debe estar Escuchando en un puerto, es decir a la espera de recibir el mensaje

Nc -<versión de IP> -u<protocolo> -l <Puerto de escucha>

- El Emisor debe Enviar el Mensaje a la dirección ip y puerto específico del Receptor
nc -<versión de IP> <Dirección IP del Receptor> <Puerto Del Receptor>

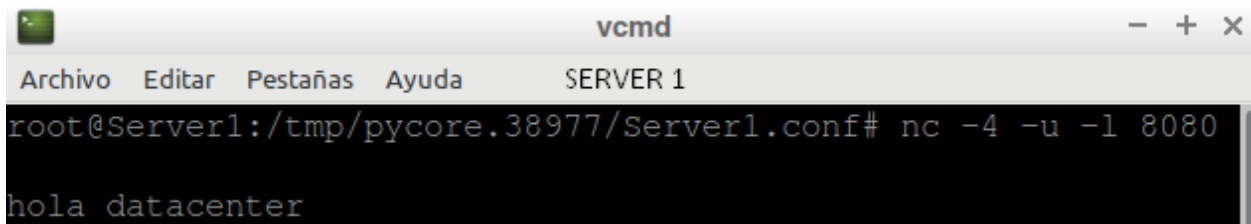
Capturas:

Envío desde CASA → hacia Server 1 (con el mensaje “Hola datacenter”)



```
vcmd
CASA
root@Casa:/tmp/pycore.38977/Casa.conf# nc -4 -u 201.0.0.2 80
hola datacenter
```

Puerto escucha 8080 abierto en Server 1 , para recibir el mensaje desde casa.



```
vcmd
SERVER 1
root@Server1:/tmp/pycore.38977/Server1.conf# nc -4 -u -l 8080
hola datacenter
```

Wireshark protocolo UDP

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000...	201.0.2.2	201.0.0.2	UDP	59	52840 → 80 Len=17

Ejercicio 2 .g)

Replique la conexión del inciso anterior pero desde la Pc1 hacia el Server 1. Indique, ¿Qué dirección IP y qué puertos se deben utilizar? ¿Qué diferencia encuentra con el inciso anterior?

Utilizamos la misma lógica que en el ejercicio 2.f) , Redirigimos el mensaje pero esta vez se lo cargamos el R3 con DNAT. en este caso en vez de ir por la publica , utilizamos la privada del R3 10.25.7.1 para así cuando llegue el mensaje a esa interfaz por el puerto 80, el 8080 escuche y reciba el mensaje. La principal diferencia es que estamos redirigiendo puertos desde la red interna del campus , en contraposición a lo que pedía el inciso anterior que era desde fuera del campus hacia uno de los server internos del datacenter.

Capturas:

Comando cargado R3

```
ip route add 201.0.2.0/24 via 10.25.0.1 dev eth0  
iptables -t nat -A PREROUTING -p udp -i eth2 --dport 80 -j DNAT --to-destination 10.25.0.3:8080
```

Envío desde Pc1 → hacia Server 1 (con el mensaje “Hola datacenter soy la pc 1”)

```
root@Pc1:/tmp/pycore.38493/Pc1.conf# nc -4 -u 10.25.7.1 80  
hola datacenter soy la pc 1
```

Puerto escucha 8080 abierto en Server 1 , para recibir el mensaje desde Pc 1.

```
root@Server1:/tmp/pycore.38493/Server1.conf# nc -4 -u -l 8080  
hola datacenter soy la pc 1
```