Práctica 7

1. La función de la capa de red es por tanto tremendamente simple: transporta paquetes desde un host emisor a un host receptor. En la realización de esta tarea podemos identificar dos importantes funciones de la capa de red:
   Servicios:
   Direccionamiento lógico

   - Asigna direcciones únicas (como IP) a los dispositivos para identificarlos en una red global.
     Encaminamiento (routing)
   - Selecciona la mejor ruta para enviar paquetes desde el origen hasta el destino, usando algoritmos de enrutamiento.
     Conmutación de paquetes (packet switching)
   - Transfiere paquetes entre redes intermedias (routers) para alcanzar el destino final.
     Encapsulamiento de datos
   - Toma los datos de la capa de transporte y los encapsula dentro de una **unidad de datos de protocolo (PDU)** para ser enviados por la red.
     Fragmentación y reensamblado (si es necesario)
   - Divide paquetes grandes en fragmentos más pequeños para adaptarse a diferentes tecnologías de enlace.
     La **PDU de la capa de red** se llama **paquete** (o **datagrama**, especialmente cuando se habla de IP). En IPv4, hablamos de **paquete IP (IP packet)** o **datagrama IP (IP datagram)**.

   El dispositivo que opera exclusivamente en la **capa de red** es el **router (enrutador)**.
   - Un **router** examina las **direcciones IP** en los paquetes y determina la mejor ruta para reenviarlos hacia su destino.
   - No accede a los datos de la capa de transporte ni de aplicación.
   - No trabaja con direcciones MAC (eso es de la **capa de enlace de datos**).
2. "**Mejor esfuerzo**" significa que la red **hace todo lo posible** por entregar los paquetes, pero **no promete nada**. En otras palabras:
   - ❌ **No garantiza** que el paquete llegue a su destino.
   - ❌ **No asegura** que los paquetes lleguen en orden.
   - ❌ **No evita duplicados**.
   - ❌ **No garantiza** que los paquetes no se pierdan o se corrompan.

- ✅ Solo **intenta** entregar los paquetes de la mejor manera posible, según la disponibilidad de la red.

3. ✅ *Clase A*
   - **Primer bit del IP:** 0
   - **Rango de direcciones IP:** `0.0.0.0` a `127.255.255.255`
   - **Primer octeto válido:** `1` a `126` (el 127 está reservado para loopback)
   - **Cantidad de redes Clase A:**
     👉 $2^7 - 2$ = **126 redes** (7 bits para la red, se restan 2: 0 y 127)
   - **Hosts por red:**
     👉 $2^{24} - 2$ = **16,777,214 hosts**
     (24 bits para host, se restan 2 direcciones: la de red y la de broadcast)
     ✅ *Clase B*
   - **Primeros bits del IP:** 10
   - **Rango de direcciones IP:** `128.0.0.0` a `191.255.255.255`
   - **Primer octeto válido:** `128` a `191`
   - **Cantidad de redes Clase B:**
     👉 2^{14} = **16,384 redes** (14 bits para red)
   - **Hosts por red:**
     👉 $2^{16} - 2$ = **65,534 hosts**
     ✅ *Clase C*
   - **Primeros bits del IP:** 110
   - **Rango de direcciones IP:** `192.0.0.0` a `223.255.255.255`
   - **Primer octeto válido:** `192` a `223`
   - **Cantidad de redes Clase C:**
     👉 2^{21} = **2,097,152 redes** (21 bits para red)
   - **Hosts por red:**
     👉 $2^8 - 2$ = **254 hosts**

4. Una **subred** (subnet) es una **división lógica de una red IP más grande** en **segmentos más pequeños**, lo que permite una mejor organización, control y uso eficiente de direcciones IP. Se logran mediante el uso de una **máscara de subred** (subnet mask), que define qué parte de la dirección IP pertenece a la **red** y qué parte a los **hosts**.

5. El campo **Protocol** (Protocolo) en la cabecera de un paquete IP (específicamente en **IPv4**) **indica a qué protocolo de la capa de transporte debe entregarse el paquete una vez que llega a su destino**. Este campo le dice al host receptor **cómo interpretar los datos contenidos en el paquete IP**.
   Analogía:
   - El **campo Protocol** en la **capa de red** dice: ‘‘entregar a TCP o UDP’’.

- El **número de puerto** en la **capa de transporte** dice: ‘‘entregar a HTTP, DNS, etc.’’.

6. A. Clases:

I. 172.16.58.223/26 --> B

II. 163.10.5.49/27 --> B

III. 128.10.1.0/23 --> B

IV. 10.1.0.0/24 --> A

V. 8.40.11.179/12 --> A

B. 172.16.58.223 --> Binario: 10101100.00010000.00111010.11011111
/26 --> Binario(máscara):
11111111.11111111.11111111.11000000
AND
IP: 10101100.00010000.00111010.11011111
MÁSCARA: 11111111.11111111.11111111.11000000
RESULTADO:10101100.00010000.00111010.11000000 = **172.16.58.192**

163.10.5.49 --> Binario: 10100011.00001010.00000101.00110001
/27 --> Binario(máscara):
11111111.11111111.11111111.11100000
AND
IP: 10100011.00001010.00000101.00110001
MÁSCARA: 11111111.11111111.11111111.11100000
RESULTADO:10100011.00001010.00000101.00100000 = **163.10.5.32**

128.10.1.0 --> Binario: 10000000.00001010.00000001.00000000
/23 --> Binario(máscara):
11111111.11111111.11111110.00000000
AND
IP: 10000000.00001010.00000001.00000000
MÁSCARA: 11111111.11111111.11111110.00000000
RESULTADO:10000000.00001010.00000000.00000000 = **128.10.0.0**

10.1.0.0 --> Binario: 00001010.00000001.00000000.00000000
/24 --> Binario(máscara):
11111111.11111111.11111111.00000000
AND
IP: 00001010.00000001.00000000.00000000
MÁSCARA: 11111111.11111111.11111111.00000000
RESULTADO:00001010.00000001.00000000.00000000 = **10.1.0.0**

8.40.11.179 --> Binario: 00001000.00101000.00001011.10110011
/12 --> Binario(máscara):
11111111.11110000.00000000.00000000
AND

IP: 00001000.00101000.00001011.10110011
MÁSCARA: 11111111.11110000.00000000.00000000
RESULTADO:00001000.00100000.00000000.00000000 = **8.32.0.0**

C. I. 2^6 - 2 = 62

II. 2^5 - 2 = 30

III. 2^9 - 2 = 254

IV. 2^8 - 2 = 126

V. 2^20 - 2 = 1.048.574

D. I. 172.16.58.255

II. 163.10.5.255

III. 128.10.1.255

IV. 10.1.0.255

V. 8.47.255.255

E. I. 172.16.58.193 - 172.16.58.254

II. 163.10.5.33 - 163.10.5.254

III. 128.10.0.1 - 128.10.1.254

IV. 10.1.0.1 - 10.1.0.254

V. 8.32.0.1 - 8.47.255.254

7. A. De Red. (Porque termina en .0 y no se especifica máscara)

B. Clase B

C. La máscara por clase para clase B es --> /16 por lo tanto la cantidad de hosts posibles es: 2^16 - 2 = 65.534

D. Bits necesarios para 513 subredes --> 10, ya que 2^9 - 2 = 512 < 513 --> 2^10 - 2 = 1024
Nueva máscara: /16 + 10 = /26
Cantidad de subredes asignables (o sea nuevas) = 1024
Cantidad de hosts por red = 2^6 - 2 = 62
Subred 710 (0 - 1023) y cantidad de direcciones por red = 64 (62 hosts + broadcast + red):
Entonces: Dirección = 128.50.10.0 + (710 * 64) = 45.440
45.440 % 256 = 177, resto 128 //para mover 1 octeto
Entonces sumamos 128.50.**10**.0 + **177** y 128.50.10.**0** + **128**
Resultado: 128.50.187.128
Broadcast (/26): 10000000.00110010.10111011.10**111111** = 128.50.187.191

8. A. 2^4 - 2 = 14, necesitaría 4 bits más --> /24 --> /28

B. 195.200.45.0 --> 11000011.11001000.00101101.0000/0000

I. 11000011.11001000.00101101.0000(0)

II. 11000011.11001000.00101101.0001(16)

III. 11000011.11001000.00101101.0010(32)

IV. 11000011.11001000.00101101.0011(48)

V. 11000011.11001000.00101101.0100(64)

VI. 11000011.11001000.00101101.0101(80)

VII. 11000011.11001000.00101101.0110(96)

VIII. 11000011.11001000.00101101.0111(112)

IX. 11000011.11001000.00101101.1000(128)

C. 11000011.11001000.00101101.10001111() -->para la red 195.200.45.0 --> Broadcast = 195.200.45.15

I. Direcciones asignables: 195.200.45.1 - 195.200.45.14

9. A. I. Red A-B (Router A) --> tiene la ip 172.26.22.3 pero debe esa ip se reserva para broadcast. Debe ser 172.26.22.2

II. Red B-C (Router C) --> tiene la ip 172.17.10.17 pero está fuera del rango, podría asignar 172.17.10.13

B. Como la máscara de las Clases A es /8 y en el gráfico se muestra /24 asumo que se tomaron 16 bits (24-8 = 16) y la cantidad de subredes posibles es 2^16 = 65.536

C. Según RFC 1918

10.0.10.0/24|10.0.0.0/8 (IP base)| ✅ Privada

192.168.5.0/24|192.168.0.0/16 (IP base)| ✅ Privada

172.17.10.0/28|172.16.0.0/12 (IP base)| ✅ Privada

172.26.22.0/30|172.16.0.0/12 (IP base)| ✅ Privada

191.26.145.0/24|No reservada (IP base)| ❌ Pública

10. El concepto de CIDR (classless inter-domain routing) se definió en la RFC 1519 como una estrategia para frenar algunos problemas que se habían comenzado a manifestar con el crecimiento de Internet.
Los mismos son:

- Agotamiento del espacio de direcciones de clase B.
- Crecimiento de las tablas de enrutamiento más allá de la capacidad del software y hardware disponibles.
- Eventual agotamiento de las direcciones IP en general.
  CIDR consiste básicamente en permitir máscaras de subred de longitud variable (VLSM) para optimizar la asignación de direcciones IP y utilizar resumen de rutas para disminuir el tamaño de las tablas de enrutamiento.

11. Cuando un router publica o anuncia rutas a otras redes (por ejemplo, a través de un protocolo como BGP), puede agrupar varias subredes contiguas en un solo prefijo CIDR si:

- Las direcciones están contiguas.

- El bloque combinado tiene una cantidad de bits comunes desde la izquierda. Esto reduce el número de rutas que los routers deben manejar → mejora el rendimiento de la red.
  198.10.0.0 = ...00000000 (último octeto)
  198.10.1.0 = ...00000001 (bloques contiguos y alineados)
  198.10.2.0 = ...00000010
  198.10.3.0 = ...00000011
  Hay que evaluar los primeros 24 bits (/24) y comprobar si podemos agregar más bits para resumirlas. --> Comparten los primeros 22 bits, luego cambian los últimos 2. Entonces se pueden agregar como una sola red /22: 198.10.0.0/22

12. A. 200.56.168.0/21 --> Caben, por ej, 8 redes /24 en la red /21. Porque toma 3 bits más, o sea 2^3 direcciones de red posibles.

I. 200.56.168.0
200.56.169.0
200.56.170.0
200.56.171.0
200.56.172.0
200.56.173.0
200.56.174.0
200.56.175.0

B. 195.24.0.0/13 --> Caben, por ej, 8 redes /16.

I. 195.24.0.0/16
195.25.0.0/16
195.26.0.0/16
195.27.0.0/16
195.28.0.0/16
195.29.0.0/16
195.30.0.0/16
195.31.0.0/16

C. 195.24/13 --> Igual que el anterior.

13. A. Incluye las direcciones que corresponden a la clase B --> con los 2 primeros bits fijos (/2): 10xxxxxx.x.x.x rango de direcciones 128.0.0.0 - 191.255.255.255

B. Bloque CIDR para todas las redes de clase A --> con el primer bit fijo (/1): 0xxxxxxx.x.x.x rango de direcciones 0.0.0.0 - 127.255.255.255

14. La técnica de VLSM (variable-length subnet masking) consiste en realizar divisiones en subredes con máscaras de longitud variable y es otra de las técnicas surgidas para frenar el agotamiento de direcciones IPv4. Básicamente, VLSM sugiere hacer varios niveles de división en redes para lograr máscaras más óptimas para cada una de las subredes que se necesiten.

15. El mecanismo consiste en asignar diferentes máscaras de subred según la cantidad de hosts que necesita cada red, comenzando por la que más necesita. Se calcula la cantidad de bits que necesita agregar y luego se subnetea la nueva red, así hasta abarcar todos los requerimientos de direcciones.

16. A. Sin VLSM todas las subredes tendrían la misma máscara, lo suficientemente grande como para cubrir la red más grande (Red C = 1530 hosts = 2^11 IPs). /21 para todas las redes.

Red	Hosts reales	IPs asignadas (/21)	IPs desperdiciadas
C	1530	2048	518
A	128	2048	1920
B	20	2048	2028
D	7	2048	2041
			Total: **6507 IPs**

B. I. Red C (1530 hosts --> /21): Asignación: 205.10.192.0/21 Rango: 205.10.192.0 - 205.10.199.255

II. Red A (128 hosts --> /24): Asignación: 205.10.200.0/24 Rango: 205.10.200.0 - 205.10.200.255

III. Red B (20 hosts --> /27): Asignación: 205.10.201.0/27 Rango. 205.10.201.0 - 205.10.201.31

IV. Red D (7 hosts --> /28): Asignación: 205.10.201.32/28 Rango: 205.10.201.32 - 205.10.201.47

C. Última IP usada: 205.10.201.47 --> Rango libre: 200.10.201.48 - 205.10.223.255

205.10.201.48/28 (16 IPs)
205.10.201.64/26 (64 IPs)
205.10.201.128/25 (128 IPs)
205.10.202.0/23 (512 IPs)
205.10.204.0/22 (1024 IPs)
205.10.208.0/20 (4096 IPs)
205.10.224.0/22 (1024 IPs)

D. I. Red C (/21): 205.10.192.0/21

a. Host: 205.10.192.1

b. Router: 205.10.192.2

II. Red A (/24): 205.10.200.0/24

a. Host: 205.10.200.1

b. Router: 205.10.200.2

III. Red B (/27): 205.10.201.32/28

a. Host: 205.10.201.1

b. Router: 205.10.201.2

IV. Red D (/28): 205.10.201.32/28

a. Host: 205.10.201.33

b. Router: 205.10.201.34

17. A.

Red	Hosts actuales	Crecimiento	Total esperado	Tamaño necesario	Máscara
A	125	+20	145	256	`/24`
X	63	0	63	64	`/26`
B	60	0	60	64	`/26`
Y	46	+18	64	64	`/26`

Conexiones entre routers:

- r1-r2
- r1-r3
- r1-r4
- r2-r9
- r4-r7
- r4-r8

B. I. Red A (/24) --> 200.100.8.0/24 Rango: 200.100.8.0 - 200.100.8.255

II. Red X (/26) --> 200.100.9.0/26 Rango: 200.100.9.0 - 200.100.9.63

III. Red B (/26) --> 200.100.9.64/26 Rango: 200.100.9.64 - 200.100.9.127

IV. Red Y (/26) --> 200.100.9.128/26 Rango: 200.100.9.128 - 200.100.9.191

V.

Enlace	Subred	IPs disponibles
r1-r2	`200.100.9.192/30`	192-195
r1-r3	`200.100.9.196/30`	196-199
r1-r4	`200.100.9.200/30`	200-203
r2-r9	`200.100.9.204/30`	204-207
r4-r7	`200.100.9.208/30`	208-211
r4-r8	`200.100.9.212/30`	212-215

18. | Red |Dirección/Máscara| IPs asignadas |
| --- | --- | --- |
| **Red A** | `200.100.8.0/24` | `n10: .1 - n6: .2 - r1: .254` |
| **Red X** | `200.100.9.0/26` | `n11: .1 - n12: .2 - n19: 3 - r2: 62` |
| **Red B** | `200.100.9.64/26` | `n14: .65 - n15: 66 - n16: 67 - n8: 68 - r4: .126` |
| **Red Y** | `200.100.9.128/26` | `n13: .129 - n7: 130 - r4: .190` |
|Enlaces r1-r2| `200.100.9.192/30` | `.193 - .194` |
|Enlace r1-r3 | `200.100.9.196/30` | `.197 - .198` |
|Enlace r1-r4 | `200.100.9.200/30` | `.201 - .202` |
|Enlace r2-r9 | `200.100.9.204/30` | `.205 - .206` |
|Enlace r4-r7 | `200.100.9.208/30` | `.209 - .210` |
|Enlace r4-r8 | `200.100.9.212/30` | `.213 - .214` |

19. Es un **protocolo auxiliar de la capa de red** (usa IP) que se utiliza para **enviar mensajes de control, error y diagnóstico** entre dispositivos de red. Aunque ICMP **no transporta datos de usuario**, es esencial para:
    - Detectar errores (por ejemplo, "host inalcanzable")
    - Informar problemas de entrega
    - Realizar pruebas de conectividad
      Sirve para:
    - Para **avisar** que un paquete no pudo ser entregado
    - Para **diagnosticar fallos** de red
    - Para enviar **mensajes informativos** o de **error**
    - Para **probar la disponibilidad** de un host (como hace `ping`)

    A. `ping` es una utilidad que utiliza ICMP para verificar si un dispositivo está disponible y cuánto tarda en responder.
    Funcionamiento:
    - El equipo **emisor** envía un mensaje ICMP tipo **Echo Request** (petición de eco)
    - El equipo **receptor** responde con un mensaje ICMP tipo **Echo Reply** (respuesta de eco)
    - El emisor mide el **tiempo de ida y vuelta (RTT)** y confirma si hubo respuesta
    - Solicitud:
      - **Tipo**: `8` → Echo Request
      - **Código**: `0`
    - Respuesta:
      - • **Tipo**: `0` → Echo Reply
      - **Código**: `0`

    B. Tanto `traceroute` (Linux) como `tracert` (Windows) son herramientas de diagnóstico que muestran el **camino (ruteo)** que siguen los paquetes IP desde

tu computadora hasta un destino (como `www.nasa.gov`).

- Funcionamiento Básico: ○ Ambos comandos envían paquetes al destino especificado con incrementos progresivos en el campo TTL (Time to Live) de los encabezados IP. La diferencia entre ellos radica principalmente en el protocolo que utilizan:
  - traceroute en Linux: Generalmente utiliza paquetes UDP.
  - tracert en Windows: Utiliza paquetes ICMP Echo Request (los mismos que ping).
- Concepto del TTL:
  - El TTL es un campo en la cabecera IP que indica la cantidad máxima de saltos (hops) que un paquete puede atravesar antes de ser descartado. Se diseñó para evitar bucles infinitos en la red.
- Proceso de decremento:
  - Cada router que recibe un paquete reduce el TTL en 1. Si el TTL llega a 0, el router descarta el paquete y envía un mensaje ICMP Time Exceeded de vuelta al emisor.

```
C:\Users\camit> tracert -d www.nasa.gov

Traza a la dirección nasa-gov.go-vip.net [2a04:fa87:fffd::c000:426c]
sobre un máximo de 30 saltos:

  1     2 ms    10 ms     9 ms  2800:810:521:1bf:d26e:deff:fe8c:2afa
  2    18 ms    17 ms    18 ms  2800:810:400:108::1
  3    20 ms    18 ms    20 ms  fc00:0:200:8a::1
  4     *        *        *     Tiempo de espera agotado para esta solicitud.
  5     *        *        *     Tiempo de espera agotado para esta solicitud.
  6     *        *        *     Tiempo de espera agotado para esta solicitud.
  7    20 ms    21 ms    19 ms  2800:1e0:1050:4::4a
  8    20 ms    21 ms    19 ms  2800:1e0:1050:4::49
  9     *        *        *     Tiempo de espera agotado para esta solicitud.
 10    49 ms    49 ms    49 ms  2800:1e0:1025::9
 11    57 ms    48 ms    56 ms  2a04:fa87:fffd::c000:426c

Traza completa.
```

C. I. Para que no muestre el nombre del dominio asociado a la IP de cada salto, se puede usar la opción -d para evitar la resolución de nombres.

II. Cuando aparece un asterisco (*) en parte o en toda la respuesta de un salto, significa que ese enrutador o dispositivo de red no respondió a la solicitud de "traceroute" o "tracert". Puede ser una medida de seguridad, configuración o simplemente que el enrutador no responde a las solicitudes ICMP utilizadas por "traceroute" para rastrear la ruta.

```
 1     8 ms     8 ms     9 ms  2800:810:521:1bf:d26e:deff:fe8c:2afa
 2    38 ms    20 ms    19 ms  2800:810:400:108::1
 3    20 ms    19 ms    17 ms  fc00:0:200:8a::1
 4     *        *        *     Tiempo de espera agotado para esta solicitud.
 5     *        *        *     Tiempo de espera agotado para esta solicitud.
 6     *        *        *     Tiempo de espera agotado para esta solicitud.
 7    22 ms    20 ms    20 ms  2001:13c7:6001::157
 8    31 ms    22 ms    22 ms  2001:13c7:6001::12
 9    23 ms    21 ms    16 ms  2001:13c7:6011::1
10    19 ms    23 ms    22 ms  2800:340:199::2
11     *        *        *     Tiempo de espera agotado para esta solicitud.
12     *        *        *     Tiempo de espera agotado para esta solicitud.
13     *        *        *     Tiempo de espera agotado para esta solicitud.
14     *        *        *     Tiempo de espera agotado para esta solicitud.
15     *        *        *     Tiempo de espera agotado para esta solicitud.
16     *        *        *     Tiempo de espera agotado para esta solicitud.
17     *        *        *     Tiempo de espera agotado para esta solicitud.
18     *        *        *     Tiempo de espera agotado para esta solicitud.
19     *        *        *     Tiempo de espera agotado para esta solicitud.
20     *        *        *     Tiempo de espera agotado para esta solicitud.
21     *        *        *     Tiempo de espera agotado para esta solicitud.
22     *        *        *     Tiempo de espera agotado para esta solicitud.
23     *        *        *     Tiempo de espera agotado para esta solicitud.
24     *        *        *     Tiempo de espera agotado para esta solicitud.
25     *        *        *     Tiempo de espera agotado para esta solicitud.
26     *        *        *     Tiempo de espera agotado para esta solicitud.
27     *        *        *     Tiempo de espera agotado para esta solicitud.
28     *        *        *     Tiempo de espera agotado para esta solicitud.
29     *        *        *     Tiempo de espera agotado para esta solicitud.
30     *        *        *     Tiempo de espera agotado para esta solicitud.

Traza completa.
```

D. I. No, no es posible determinar el camino que siguen los servidores de nombres del dominio unlp.edu.ar debido a que todas las respuestas muestran "Tiempo de espera agotado para esta solicitud.". Esto puede ser porque los servidores de nombres de dominio están configurados para no responder a traceroute o porque hay un problema en la red que impide que las respuestas lleguen de vuelta.

20. El bloque de direcciones IP 127.0.0.0/8 está reservado para el uso en la red de loopback. La dirección IP más comúnmente utilizada en este bloque es 127.0.0.1, que se conoce como "localhost". La dirección de loopback se utiliza para permitir que un dispositivo se comunique consigo mismo, lo que es útil en el diagnóstico y la prueba de aplicaciones y servicios de red sin la necesidad de acceder a una red real.

A.

```
C:\Users\camit> ping 127.0.0.1

Haciendo ping a 127.0.0.1 con 32 bytes de datos:
Respuesta desde 127.0.0.1: bytes=32 tiempo<1m TTL=128
Respuesta desde 127.0.0.1: bytes=32 tiempo<1m TTL=128
Respuesta desde 127.0.0.1: bytes=32 tiempo<1m TTL=128
Respuesta desde 127.0.0.1: bytes=32 tiempo<1m TTL=128

Estadísticas de ping para 127.0.0.1:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 0ms, Máximo = 0ms, Media = 0ms
```

B.

```
C:\Users\camit> ping 127.0.54.43

Haciendo ping a 127.0.54.43 con 32 bytes de datos:
Respuesta desde 127.0.54.43: bytes=32 tiempo<1m TTL=128
Respuesta desde 127.0.54.43: bytes=32 tiempo<1m TTL=128
Respuesta desde 127.0.54.43: bytes=32 tiempo<1m TTL=128
Respuesta desde 127.0.54.43: bytes=32 tiempo<1m TTL=128

Estadísticas de ping para 127.0.54.43:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 0ms, Máximo = 0ms, Media = 0ms
```

21. **A.** Comando **ifconfig**
    - Función: Se utiliza para configurar interfaces de red en sistemas Linux. Permite asignar o cambiar direcciones IP, habilitar o deshabilitar interfaces, y mostrar información detallada sobre ellas.
      - Ejemplo: *ifconfig eth0 192.168.1.10 netmask 255.255.255.0 up*
    - Reemplazo: ip (paquete iproute2) ha reemplazado a ifconfig.
      - Ejemplo equivalente: *ip addr add 192.168.1.10/24 dev eth0 ip link set eth0 up*

    B. Comando **route**

    - Función: Se utiliza para manipular la tabla de rutas, mostrando o modificando las rutas estáticas del sistema.
    - Ejemplo: *route add default gw 192.168.1.1*
    - Reemplazo: ip route es el reemplazo moderno.
    - Ejemplo equivalente: *ip route add default via 192.168.1.1*

    Práctica con CORE (Common Open Research Emulator)

- Iniciar CORE: Al abrir CORE, puedes crear una topología básica:
  - Agrega un nodo, que simulará una máquina.
  - Configurar interfaces y conexiones en este nodo.
- Configuración de IP:
  - Abre la terminal del nodo.
- Configura una dirección IP con ifconfig o ip:
  - *ifconfig eth0 10.0.0.1 netmask 255.255.255.0 up*
  - *# o con ip ip*
  - *addr add 10.0.0.1/24 dev eth0*
- Para eliminar la dirección IP:
  - *ifconfig eth0 down*
  - *# o con ip*
  - *ip addr del 10.0.0.1/24 dev eth0*
- Ver la tabla de ruteo:
  - *route -n*
  - *# o con ip*
  - *ip route show*