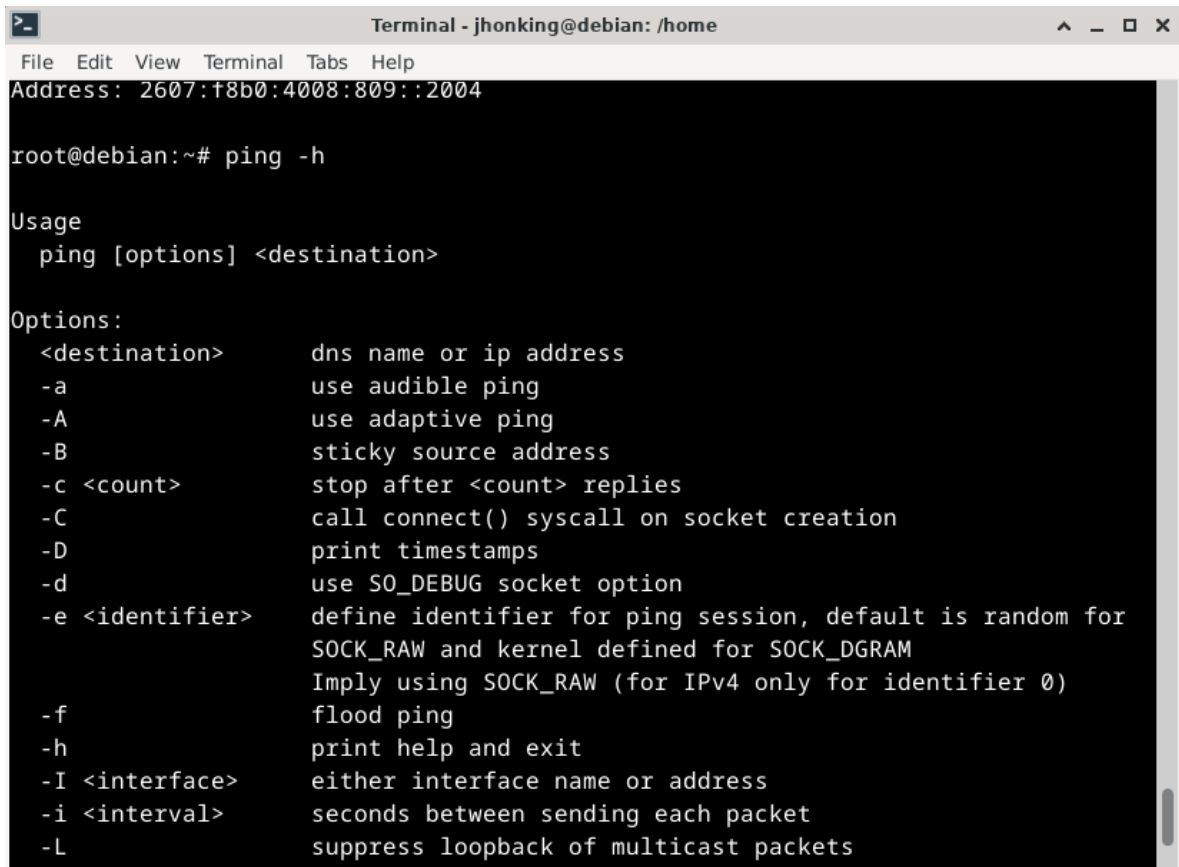


Anotar los comandos necesarios para ejecutar las siguientes instrucciones desde la consola de Ms:

DOS

1-Obtener la ayuda del comando ping



```
Terminal - jhonking@debian: /home
File Edit View Terminal Tabs Help
Address: 2607:f8b0:4008:809::2004

root@debian:~# ping -h

Usage
  ping [options] <destination>

Options:
  <destination>    dns name or ip address
  -a              use audible ping
  -A              use adaptive ping
  -B              sticky source address
  -c <count>      stop after <count> replies
  -C              call connect() syscall on socket creation
  -D              print timestamps
  -d              use SO_DEBUG socket option
  -e <identifier> define identifier for ping session, default is random for
                  SOCK_RAW and kernel defined for SOCK_DGRAM
                  (for IPv4 only for identifier 0)
  -f              flood ping
  -h              print help and exit
  -I <interface>  either interface name or address
  -i <interval>   seconds between sending each packet
  -L              suppress loopback of multicast packets
```

2.- Enviar un ping a 127.0.0.1 aplicando cualquier parametro

```
Terminal - jhonking@debian: /home
File Edit View Terminal Tabs Help
For more details see ping(8).
root@debian:~# ping 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.024 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.047 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.040 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.041 ms
64 bytes from 127.0.0.1: icmp_seq=5 ttl=64 time=0.034 ms
64 bytes from 127.0.0.1: icmp_seq=6 ttl=64 time=0.042 ms
64 bytes from 127.0.0.1: icmp_seq=7 ttl=64 time=0.046 ms
64 bytes from 127.0.0.1: icmp_seq=8 ttl=64 time=0.044 ms
64 bytes from 127.0.0.1: icmp_seq=9 ttl=64 time=0.042 ms
64 bytes from 127.0.0.1: icmp_seq=10 ttl=64 time=0.041 ms
64 bytes from 127.0.0.1: icmp_seq=11 ttl=64 time=0.046 ms
64 bytes from 127.0.0.1: icmp_seq=12 ttl=64 time=0.042 ms
64 bytes from 127.0.0.1: icmp_seq=13 ttl=64 time=0.049 ms
64 bytes from 127.0.0.1: icmp_seq=14 ttl=64 time=0.041 ms
64 bytes from 127.0.0.1: icmp_seq=15 ttl=64 time=0.047 ms
64 bytes from 127.0.0.1: icmp_seq=16 ttl=64 time=0.043 ms
64 bytes from 127.0.0.1: icmp_seq=17 ttl=64 time=0.035 ms
64 bytes from 127.0.0.1: icmp_seq=18 ttl=64 time=0.042 ms
64 bytes from 127.0.0.1: icmp_seq=19 ttl=64 time=0.039 ms
64 bytes from 127.0.0.1: icmp_seq=20 ttl=64 time=0.041 ms
64 bytes from 127.0.0.1: icmp_seq=21 ttl=64 time=0.040 ms
```

3.- Verificar la conectividad del equipo utilizando el comando ping, anotar conclusiones

```
root@debian:~# ping www.google.com
PING www.google.com (142.250.189.132) 56(84) bytes of data.
64 bytes from mia09s26-in-f4.1e100.net (142.250.189.132): icmp_seq=1 ttl=117 time=25.3 ms
64 bytes from mia09s26-in-f4.1e100.net (142.250.189.132): icmp_seq=2 ttl=117 time=24.7 ms
64 bytes from mia09s26-in-f4.1e100.net (142.250.189.132): icmp_seq=3 ttl=117 time=22.7 ms
64 bytes from mia09s26-in-f4.1e100.net (142.250.189.132): icmp_seq=4 ttl=117 time=22.7 ms
64 bytes from mia09s26-in-f4.1e100.net (142.250.189.132): icmp_seq=5 ttl=117 time=24.1 ms
64 bytes from mia09s26-in-f4.1e100.net (142.250.189.132): icmp_seq=6 ttl=117 time=22.7 ms
^C
--- www.google.com ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5000ms
rtt min/avg/max/mdev = 22.654/23.710/25.316/1.055 ms
```

Esto muestra que la conectividad esta correctamente, ya que si envía paquetes correctamente.

4-Obtener la ayuda del comando nslookup

```
root@debian:~# man nslookup
```

```
Terminal - jhonking@debian: ~
File Edit View Terminal Tabs Help
NSLOOKUP(1)                                BIND 9                                NSLOOKUP(1)

NAME
    nslookup - query Internet name servers interactively

SYNOPSIS
    nslookup [-option] [name | -] [server]

DESCRIPTION
    nslookup is a program to query Internet domain name servers. nslookup
    has two modes: interactive and non-interactive. Interactive mode allows
    the user to query name servers for information about various hosts and
    domains or to print a list of hosts in a domain. Non-interactive mode
    prints just the name and requested information for a host or domain.

ARGUMENTS
    Interactive mode is entered in the following cases:

    a. when no arguments are given (the default name server is used);

    b. when the first argument is a hyphen (-) and the second argument is
        the host name or Internet address of a name server.

Manual page nslookup(1) line 1 (press h for help or q to quit)
```

5-Resolver la direccion ip de <https://upqroo.edu.mx/> usando nslookup

```
root@debian:~# nslookup upqroo.edu.mx
Server:      8.8.8.8
Address:     8.8.8.8#53

Non-authoritative answer:
Name:   upqroo.edu.mx
Address: 77.68.126.20
```

6-Hacer ping a la ip obtenida en el paso anterior, anotar conclusiones

```

root@debian:~# ping 77.68.126.20
PING 77.68.126.20 (77.68.126.20) 56(84) bytes of data.
64 bytes from 77.68.126.20: icmp_seq=1 ttl=49 time=125 ms
64 bytes from 77.68.126.20: icmp_seq=2 ttl=49 time=124 ms
64 bytes from 77.68.126.20: icmp_seq=3 ttl=49 time=123 ms
64 bytes from 77.68.126.20: icmp_seq=4 ttl=49 time=136 ms
64 bytes from 77.68.126.20: icmp_seq=5 ttl=49 time=146 ms
64 bytes from 77.68.126.20: icmp_seq=6 ttl=49 time=122 ms
64 bytes from 77.68.126.20: icmp_seq=7 ttl=49 time=121 ms
64 bytes from 77.68.126.20: icmp_seq=8 ttl=49 time=123 ms
64 bytes from 77.68.126.20: icmp_seq=9 ttl=49 time=123 ms
^C
--- 77.68.126.20 ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 8012ms
rtt min/avg/max/mdev = 120.960/127.083/146.221/8.010 ms

```

Que si se conecta.

7-Obtener la ayuda del comando netstat

```

root@debian:~# ss -h
Usage: ss [ OPTIONS ]
        ss [ OPTIONS ] [ FILTER ]
    -h, --help                this message
    -V, --version              output version information
    -n, --numeric              don't resolve service names
    -r, --resolve              resolve host names
    -a, --all                  display all sockets
    -l, --listening            display listening sockets
    -o, --options              show timer information
    -e, --extended             show detailed socket information
    -m, --memory               show socket memory usage
    -p, --processes            show process using socket
    -T, --threads              show thread using socket
    -i, --info                 show internal TCP information
        --tipcinfo             show internal tipc socket information
    -s, --summary              show socket usage summary
        --tos                  show tos and priority information
        --cgroup               show cgroup information
    -b, --bpf                  show bpf filter socket information
    -E, --events               continually display sockets as they are destroyed
    -Z, --context              display task SELinux security contexts
    -z, --contexts             display task and socket SELinux security contexts

```

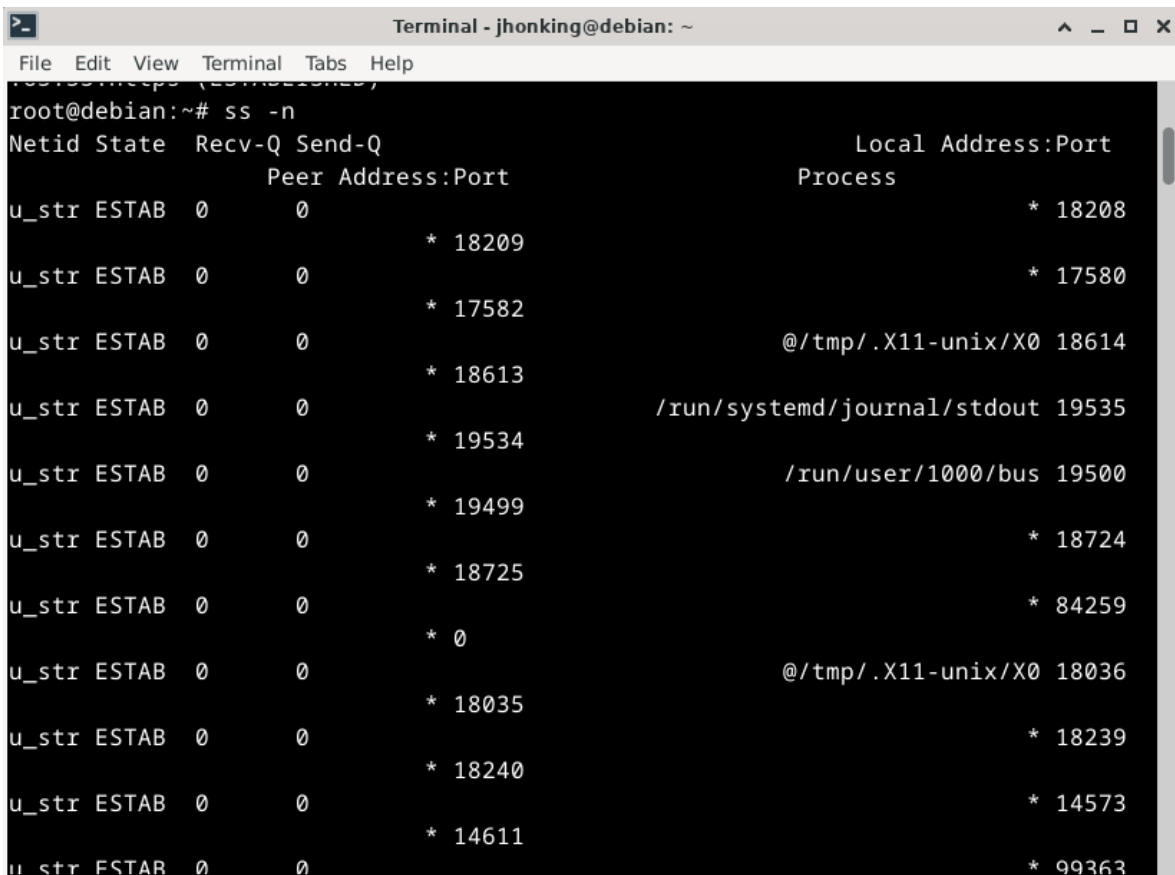
8-Mostrar todas las conexiones y puertos de escucha

```

root@debian:~# lsof -i -n
COMMAND    PID     USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
avahi-dae  411    avahi  12u  IPv4  14617      0t0  UDP *:mdns
avahi-dae  411    avahi  13u  IPv6  14618      0t0  UDP *:mdns
avahi-dae  411    avahi  14u  IPv4  14619      0t0  UDP *:35938
avahi-dae  411    avahi  15u  IPv6  14620      0t0  UDP *:38119
NetworkMa  467     root   26u  IPv4  15236      0t0  UDP 10.0.2.15:bootpc->10.0.
2.2:bootps
cupsd      513     root    6u  IPv6  14988      0t0  TCP [::1]:ipp (LISTEN)
cupsd      513     root    7u  IPv4  14989      0t0  TCP 127.0.0.1:ipp (LISTEN)
sshd       527     root    3u  IPv4  15019      0t0  TCP *:ssh (LISTEN)
sshd       527     root    4u  IPv6  15030      0t0  TCP *:ssh (LISTEN)
cups-brow  595     root    7u  IPv4  15350      0t0  UDP *:631
x-www-bro  2473   jhonking 92u  IPv4  96085      0t0  TCP 10.0.2.15:60008->34.117
.65.55:https (ESTABLISHED)

```

9- Ejecutar netstat sin resolver nombres de dominio o puertos



```

root@debian:~# ss -n
Netid State  Recv-Q Send-Q               Local Address:Port
Peer Address:Port                               Process
u_str  ESTAB  0      0               * 18208
* 18209
* 17580
* 17582
u_str  ESTAB  0      0               @/tmp/.X11-unix/X0 18614
* 18613
/run/systemd/journal/stdout 19535
* 19534
/run/user/1000/bus 19500
* 19499
* 18724
* 18725
* 84259
* 0
u_str  ESTAB  0      0               @/tmp/.X11-unix/X0 18036
* 18035
* 18239
* 18240
u_str  ESTAB  0      0               * 14573
* 14611
u_str  ESTAB  0      0               * 99363

```

10-Mostrar las conexiones TCP

```

root@debian:~# ss -tn
State  Recv-Q  Send-Q      Local Address:Port      Peer Address:Port  Process
ESTAB  0        0          10.0.2.15:56738         34.117.237.239:443
ESTAB  0        0          10.0.2.15:59124         34.120.115.102:443
ESTAB  0        0          10.0.2.15:60008         34.117.65.55:443

```

11-Mostrar las conexiones UDP

```

root@debian:~# ss -un
Recv-Q  Send-Q      Local Address:Port      Peer Address:Port  Process
0        0          10.0.2.15:enp0s3:68      10.0.2.2:67
root@debian:~#

```

12-Utilizar el comando tasklist

```

Terminal - jhonking@debian: ~
File Edit View Terminal Tabs Help
0        0          10.0.2.15:enp0s3:68      10.0.2.2:67
root@debian:~# ps aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.5 102116 11184 ?        Ss   21:45   0:01 /sbin/init
root         2  0.0  0.0      0     0 ?        S    21:45   0:00 [kthreadd]
root         3  0.0  0.0      0     0 ?        I<   21:45   0:00 [rcu_gp]
root         4  0.0  0.0      0     0 ?        I<   21:45   0:00 [rcu_par_gp]
root         5  0.0  0.0      0     0 ?        I<   21:45   0:00 [slub_flushwq]
root         6  0.0  0.0      0     0 ?        I<   21:45   0:00 [netns]
root         8  0.0  0.0      0     0 ?        I<   21:45   0:00 [kworker/0:0H]
root        10  0.0  0.0      0     0 ?        I<   21:45   0:00 [mm_percpu_wq]
root        11  0.0  0.0      0     0 ?        I    21:45   0:00 [rcu_tasks_kt]
root        12  0.0  0.0      0     0 ?        I    21:45   0:00 [rcu_tasks_ru]
root        13  0.0  0.0      0     0 ?        I    21:45   0:00 [rcu_tasks_tr]
root        14  0.0  0.0      0     0 ?        S    21:45   0:02 [ksoftirqd/0]
root        15  0.0  0.0      0     0 ?        I    21:45   0:01 [rcu_preempt]
root        16  0.0  0.0      0     0 ?        S    21:45   0:00 [migration/0]
root        18  0.0  0.0      0     0 ?        S    21:45   0:00 [cpuhp/0]
root        20  0.0  0.0      0     0 ?        S    21:45   0:00 [kdevtmpfs]
root        21  0.0  0.0      0     0 ?        I<   21:45   0:00 [inet_frag_wq]
root        22  0.0  0.0      0     0 ?        S    21:45   0:00 [kauditd]
root        23  0.0  0.0      0     0 ?        S    21:45   0:00 [khungtaskd]
root        24  0.0  0.0      0     0 ?        S    21:45   0:00 [oom_reaper]
root        27  0.0  0.0      0     0 ?        I<   21:45   0:00 [writeback]

```

13-Utilizar el comando taskkill

```

root@debian:~# kill 5099
root@debian:~#

```

14-Utilizar el comando tracert

```

root@debian:~# traceroute www.google.com
traceroute to www.google.com (142.250.189.132), 30 hops max, 60 byte packets
 1  10.0.2.2 (10.0.2.2)  0.169 ms  0.067 ms  0.109 ms^C
root@debian:~#

```

15-Utilizar el comando ARP

```

root@debian:~# arp
Address                  HWtype  HWaddress           Flags Mask            Iface
10.0.2.2                 ether    52:54:00:12:35:02    C                     enp0s3
root@debian:~#

```

B) Contesta con tus propias palabras las siguientes preguntas:

1-Verifica la conexión entre tu computadora y otros dispositivos en una red. Envía paquetes y mide la respuesta para comprobar si un host está disponible y evaluar la velocidad de la red.

2-Se utiliza para buscar y obtener información de nombres de dominio, como la resolución de direcciones IP a partir de nombres de dominio y detalles de servidores de nombres DNS en una red.

3-El comando netstat muestra información sobre conexiones de red, puertos abiertos y estadísticas. Se utiliza para monitorear y solucionar problemas de red, identificar conexiones activas y puertos de escucha.

4-El comando ps muestra información sobre procesos en sistemas Unix y Linux. Permite ver una lista de procesos en ejecución, con detalles como ID de proceso (PID) y recursos utilizados.

5-El comando kill se utiliza para finalizar procesos en sistemas Unix y Linux. Permite detener programas en ejecución enviando señales específicas, como SIGTERM o SIGKILL, para cerrarlos de manera controlada o forzada.

6-El comando traceroute rastrea la ruta de un paquete de datos en una red, mostrando los saltos que atraviesa y el tiempo que lleva para llegar a un destino específico. Se usa para diagnosticar problemas de red y entender la ruta de los datos.

7-Ping verifica la conectividad. Nslookup resuelve nombres de dominio. Netstat muestra conexiones. Combinados, ayudan a diagnosticar problemas de red, como conectividad, resolución DNS y análisis de tráfico.

C) Investigar los siguientes comandos y anotar ejemplos prácticos:

1-Supervisa las conexiones y direcciones registradas por el Administrador de Llamadas atM en una red de modo de transferencia asincrónica (atM). Puede usar **atmadm** para mostrar las estadísticas de las llamadas entrantes y salientes en adaptadores atM. Si lo usa sin parámetros, atmadm muestra estadísticas para supervisar el estado de las conexiones atM activas.

```
atmadm /c
```

2-El comando **bitsadmin** es una herramienta de línea de comandos que se utiliza en sistemas Windows para administrar trabajos de transferencia de archivos en segundo plano. "BITS" significa

```
bitsadmin /transfer miDescarga /download /priority normal http://example.com/archivo.zip C:\ruta\de\destino\archivo.zip
```

"Background Intelligent Transfer Service". BITSAdmin permite crear, modificar, consultar y administrar tareas de transferencia de archivos en el servicio BITS.

3-El comando **cmstp** es una herramienta de línea de comandos que se utiliza en sistemas Windows para instalar y administrar perfiles de conexión de red en un sistema. "CMSTP" significa "Connection Manager Profile Installer". Este comando se utiliza para automatizar la instalación de perfiles de conexión de red, que pueden incluir configuraciones de acceso a Internet o redes VPN.

```
cmstp.exe /au "vpn_profile.inf"
```

4-FTP (File Transfer Protocol) es un protocolo de red que se utiliza para la transferencia de archivos entre un cliente y un servidor a través de una red, como Internet. FTP permite copiar archivos de un lugar a otro de manera eficiente y es ampliamente utilizado para cargar y descargar archivos en servidores web, administrar sitios web y transferir datos entre sistemas.

```
ftp ftp.example.com|
```

5-El comando **getmac** es un comando de Windows que se utiliza para obtener la dirección MAC (Media Access Control) de una interfaz de red en un sistema Windows. La dirección MAC es un identificador único asociado a una tarjeta de red o adaptador de red en una computadora.

```
getmac /v /fo list /nh|
```

6-El comando **hostname** se utiliza para mostrar o configurar el nombre del host de una computadora en un sistema Unix, Linux o Windows. El nombre del host es una etiqueta alfanumérica que se utiliza para identificar de manera única una computadora en una red.

```
C:\Users\jhoan>hostname  
LapKingz
```


7-El comando **nbtstat** es una herramienta de línea de comandos utilizada en sistemas Windows para obtener información relacionada con el protocolo NetBIOS (Sistema de Entrada y Salida Básica de Red). NetBIOS es un conjunto de protocolos que permite la comunicación entre computadoras en una red local. nbtstat proporciona información sobre la resolución de nombres NetBIOS y el estado de NetBIOS en un sistema Windows.

```
C:\Users\jhoan>nbtstat -c

Ethernet 2:
Dirección IP del nodo: [192.168.56.1] Id. de ámbito : []

    No hay nombres en la caché

Ethernet:
Dirección IP del nodo: [0.0.0.0] Id. de ámbito : []

    No hay nombres en la caché
```

8-El comando **net** es una herramienta de línea de comandos en sistemas Windows que proporciona una variedad de funciones relacionadas con la administración de redes y sistemas. Puede utilizarse para llevar a cabo tareas relacionadas con la gestión de usuarios, recursos compartidos, servicios, etc.

```
>net localgroup|
```

9-El comando **net use** es una herramienta de línea de comandos en sistemas Windows que se utiliza para conectar o desconectar unidades de red. Puedes usar este comando para asignar una letra de unidad a una ubicación de red compartida, como una carpeta o un recurso compartido en otro equipo

```
>net use Z: \\servidor\recurso /user:nombre_de_usuario contraseña|
```

10-El comando **netsh** es una herramienta de línea de comandos en sistemas Windows que permite configurar y administrar una amplia variedad de configuraciones y componentes de red. Puedes utilizar netsh para realizar tareas como configurar la red, modificar parámetros de red, diagnosticar problemas de red y administrar servicios relacionados con la red. netsh es una herramienta poderosa y versátil que se utiliza para gestionar aspectos de la configuración de red en sistemas Windows.

```
>netsh interface ipv4 set address "Nombre de la interfaz" static IP Máscara Puerta_de_enlace|
```

11-**pathping** es una herramienta de diagnóstico de red en sistemas Windows que combina la funcionalidad de dos comandos: ping y tracert (tracert). Esta herramienta permite realizar un seguimiento más detallado de la ruta de los paquetes a través de la red y proporcionar estadísticas sobre la calidad de la conexión en cada salto.

```
pathping www.google.com
```

12-**rcp** (Remote Copy Protocol) es un protocolo de transferencia de archivos que se utiliza para copiar archivos entre sistemas Unix y Linux. El protocolo rcp permite la copia de archivos desde una máquina local a una máquina remota o desde una máquina remota a una máquina local de manera similar al comando cp (copia) en sistemas Unix.

```
>rcp archivo.txt usuario@maquina-remota:/ruta/de/destino/|
```

13-El comando **rexec** (Remote Execution) es una herramienta que permite ejecutar comandos en un sistema remoto desde una computadora local. rexec se utiliza en sistemas Unix y Linux y es parte de las herramientas estándar de red en sistemas UNIX. A través de rexec, puedes iniciar procesos o ejecutar comandos en un sistema remoto con el permiso del usuario remoto.

```
>rexec host -l usuario -p puerto comando|
```

14-El comando **route** se utiliza en sistemas Unix, Linux y Windows para mostrar y administrar la tabla de enrutamiento, que es una lista de rutas que determinan cómo se dirigen los paquetes de datos en una red. La tabla de enrutamiento especifica las rutas a través de las cuales los paquetes de datos deben ser enviados para alcanzar su destino.

```
>route -n|
```

15-**rpcping** es una herramienta de diagnóstico utilizada para probar la conectividad entre un cliente y un servidor que utilizan el Protocolo RPC (Remote Procedure Call). RPC es un protocolo utilizado para comunicación entre aplicaciones distribuidas en sistemas Windows y otros sistemas operativos. rpcping permite verificar si un servidor RPC es accesible desde un cliente y si los procedimientos remotos están disponibles.

```
>rpcping -s servidor -o endpoint|
```

16-**rsh** (Remote Shell) es un protocolo y un conjunto de comandos que permiten la ejecución de comandos en un sistema remoto desde una máquina local en una red. El protocolo rsh es parte del conjunto de herramientas de comunicación de red en sistemas Unix y Linux. Sin embargo, es importante destacar que rsh no proporciona cifrado de datos, lo que lo hace inseguro para su uso en redes no confiables o públicas.

```
>rsh servidor-remoto -l usuario-remoto ls -l|
```

17-**tcmssetup**: Configura o deshabilita el cliente TAPI. Para que TAPI funcione correctamente, debe ejecutar este comando para especificar los servidores remotos que usarán los clientes TAPI.

```
tcmsetup [/q] [/x] /c <server1> [<server2> ...]  
tcmsetup  [/q] /c /d
```

18-Telnet es un protocolo de red y una herramienta de línea de comandos que se utiliza para establecer conexiones de terminal a través de una red, como Internet. Permite acceder a una computadora o dispositivo remoto y ejecutar comandos en ese dispositivo como si estuvieras físicamente presente en el lugar. Telnet es ampliamente utilizado en entornos de administración de sistemas y redes.

```
>telnet dirección_ip_o_nombre_de_host puerto|
```

19-**TFTP** (Trivial File Transfer Protocol) es un protocolo de transferencia de archivos simple y liviano utilizado para la transferencia de archivos en una red, especialmente en entornos de arranque de dispositivos y sistemas embebidos. A diferencia de protocolos de transferencia de archivos más complejos como FTP, TFTP es minimalista y no incluye autenticación ni características de seguridad avanzadas, lo que lo hace adecuado para tareas específicas y simples.

```
>tftp -g -r archivo_remoto -l archivo_local dirección_servidor|
```