



UNIVERSIDAD SANTO TOMÁS

Facultad de Ingeniería Electrónica

Laboratorio 4: Explorando el mundo de las comunicaciones

Estudiantes: Yojan Contreras - Cristian Losada

Docente: Ing. Diego Alejandro Barragán Vargas

Bogotá D.C., Octubre 2025

1. Objetivo

Configurar y verificar una conexión básica entre un PC, un switch Cisco 2960 y una Raspberry Pi, estableciendo comunicación mediante interfaz serial y Ethernet. Explorar comandos de red en Linux y comprobar la conectividad mediante `ping`, `nmap` e intercambio de archivos con `scp` usando un servidor `ssh`.

2. Materiales y Herramientas

- Switch Cisco 2960
- Raspberry Pi
- Dos laptops con sistema operativo Linux
- Un PC del laboratorio con Windows
- Cables Ethernet y adaptador USB–Serial
- Software: `minicom`, `screen`, `picocom`, `nmap`, `ifconfig`, `ip`, `scp`, `openssh-server`

3. Procedimiento

3.1. Configuración inicial del switch por conexión serial

Antes de establecer la red, se realizó la conexión al switch Cisco 2960 mediante el puerto serial. Primero se instaló el software de terminal `minicom`:

```
sudo apt install minicom  
sudo minicom -s
```

Dentro del menú de configuración se seleccionó el puerto serial detectado:

```
ls /dev/ttyUSB0
```

Se configuraron los parámetros:

- Baud Rate: 9600
- Data Bits: 8
- Stop Bits: 1

- Paridad: None
- Flow Control: None

Posteriormente se accedió con **screen**:

```
sudo screen /dev/ttyUSB0 9600
```

Dentro de la consola del switch:

- Se verificaron los comandos disponibles con ?
- Se revisaron los puertos con:

```
show interfaces status  
show vlan brief  
show mac address-table
```

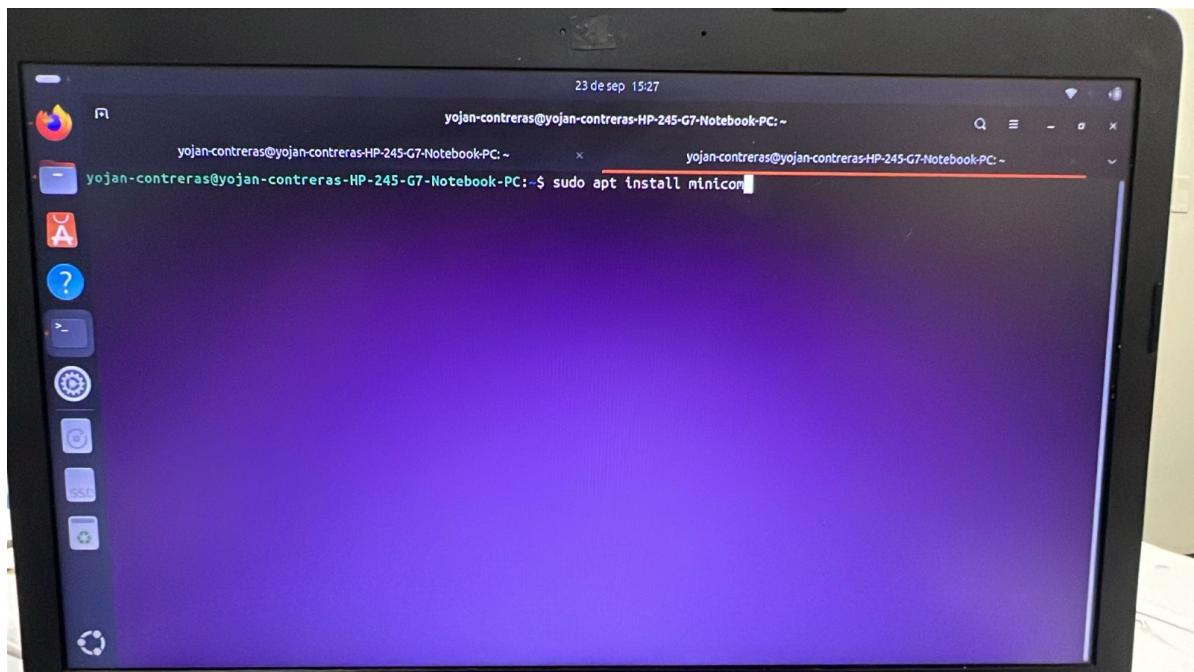


Figure 1: Instalar minicom

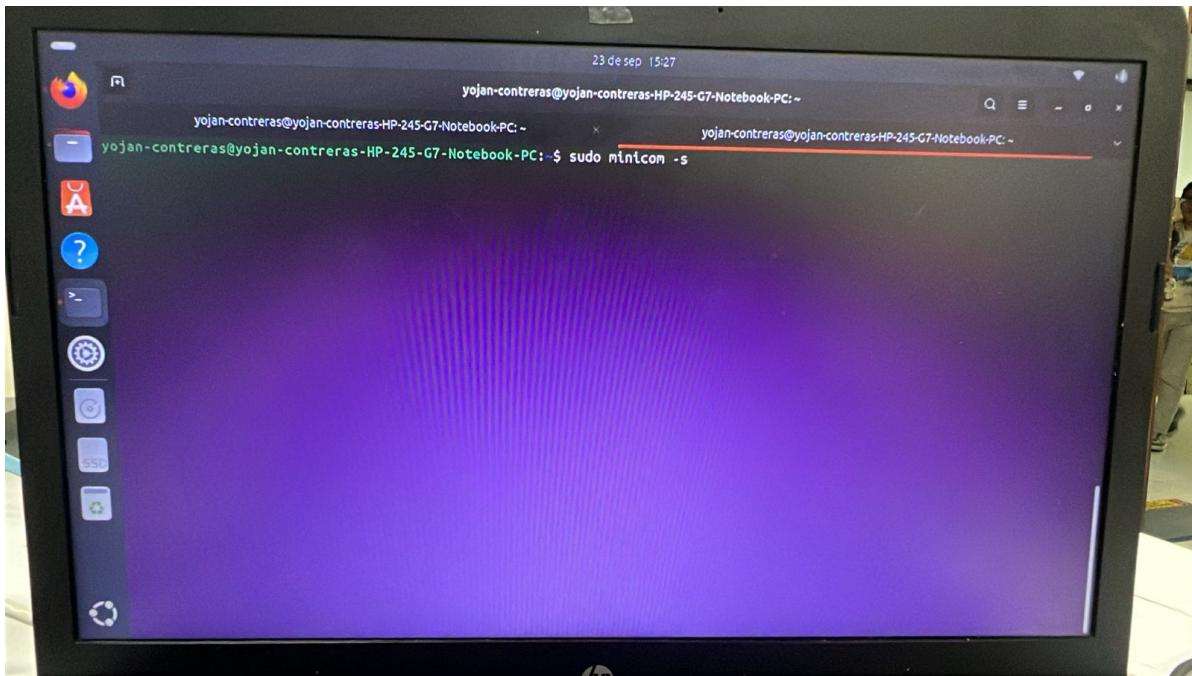


Figure 2: Acceso a minicom

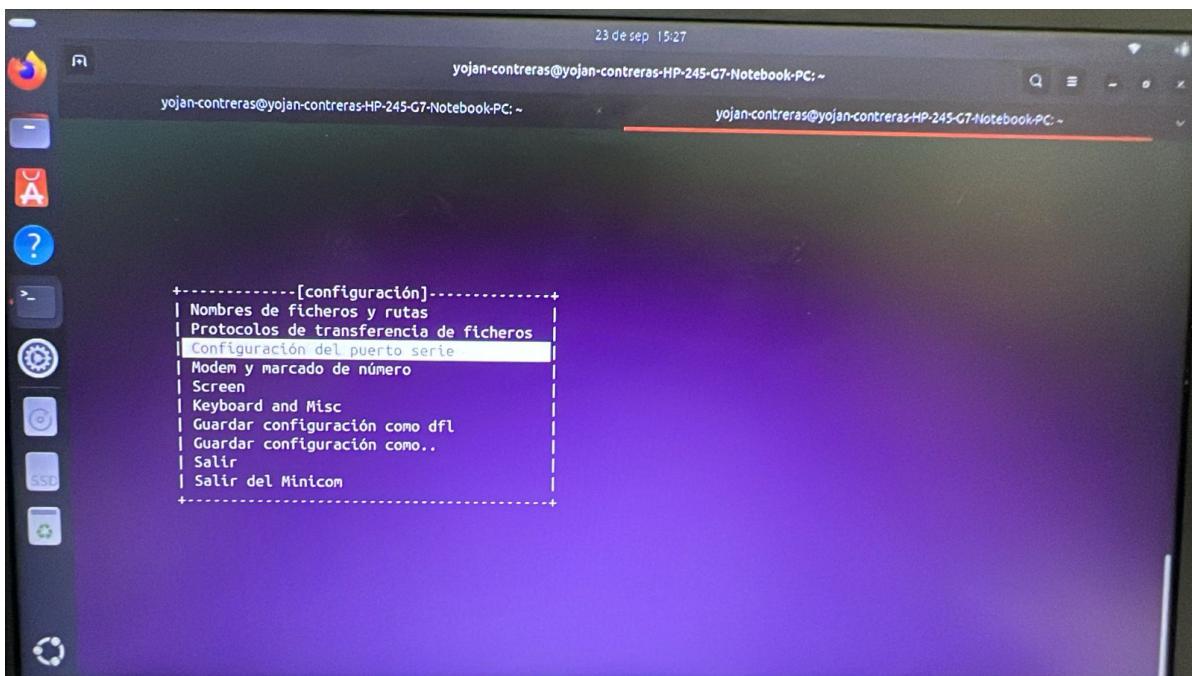


Figure 3: Seleccionamos Configuracion puertos

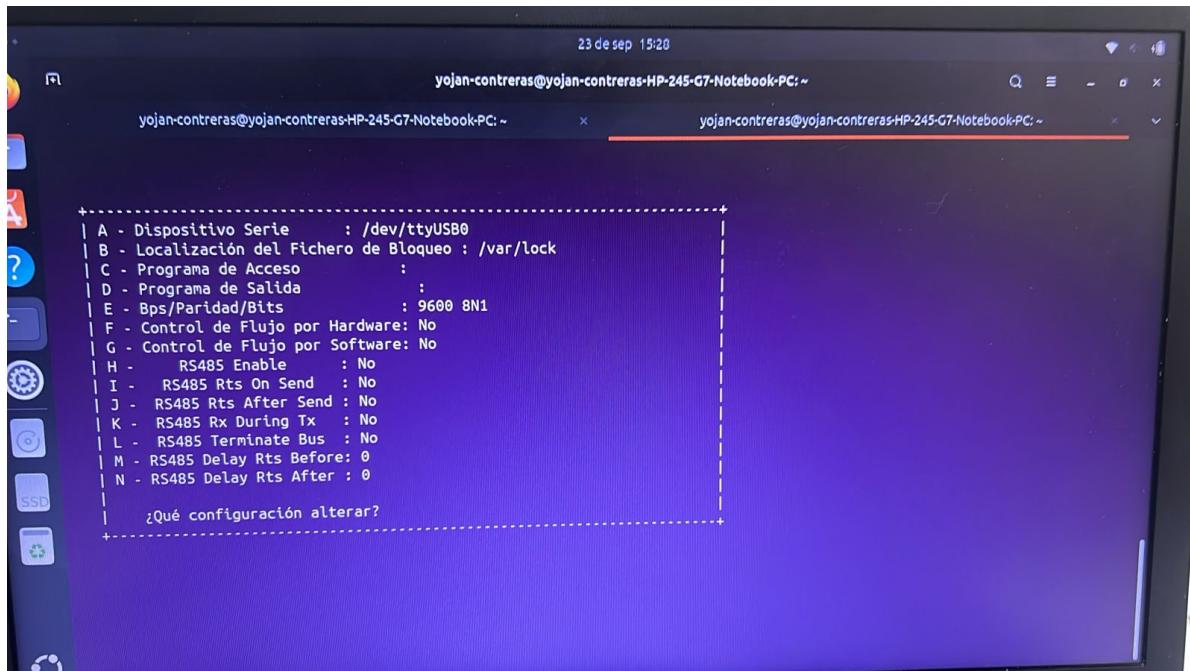


Figure 4: Configuramos

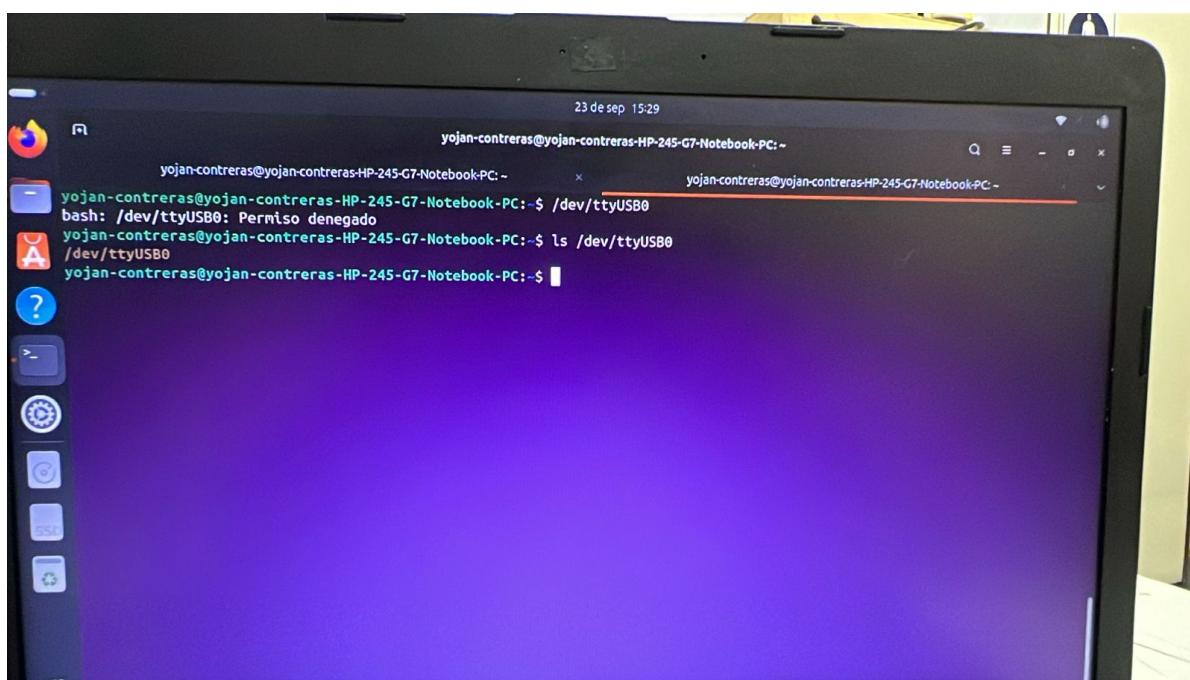
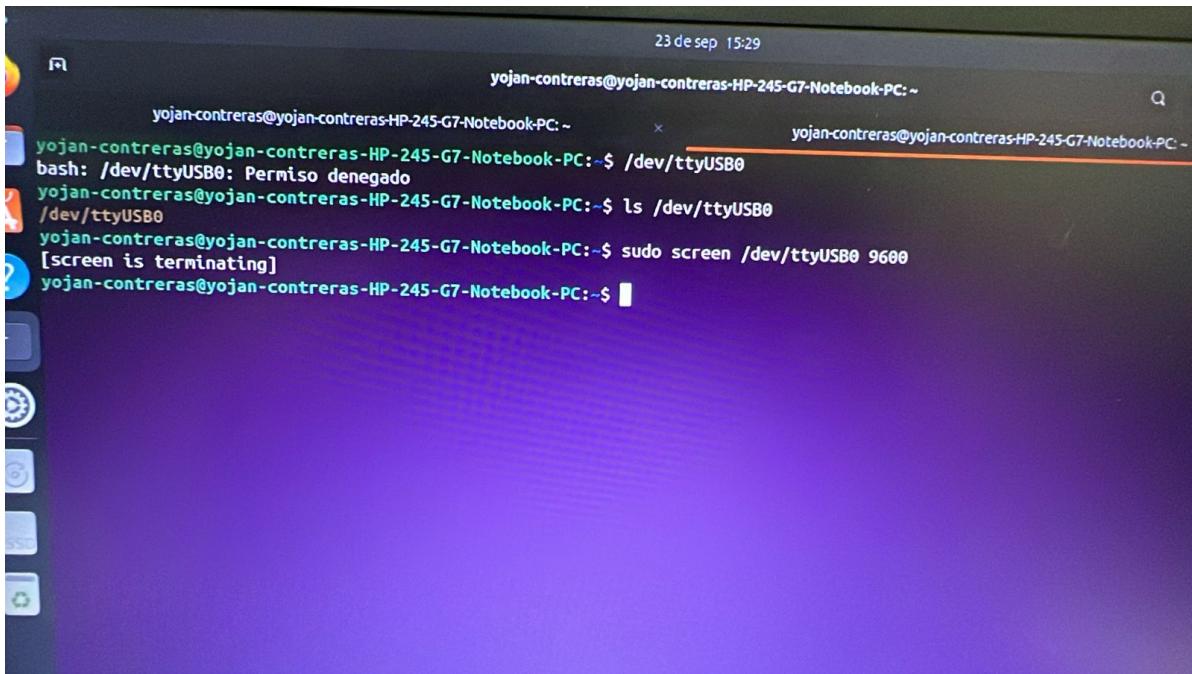
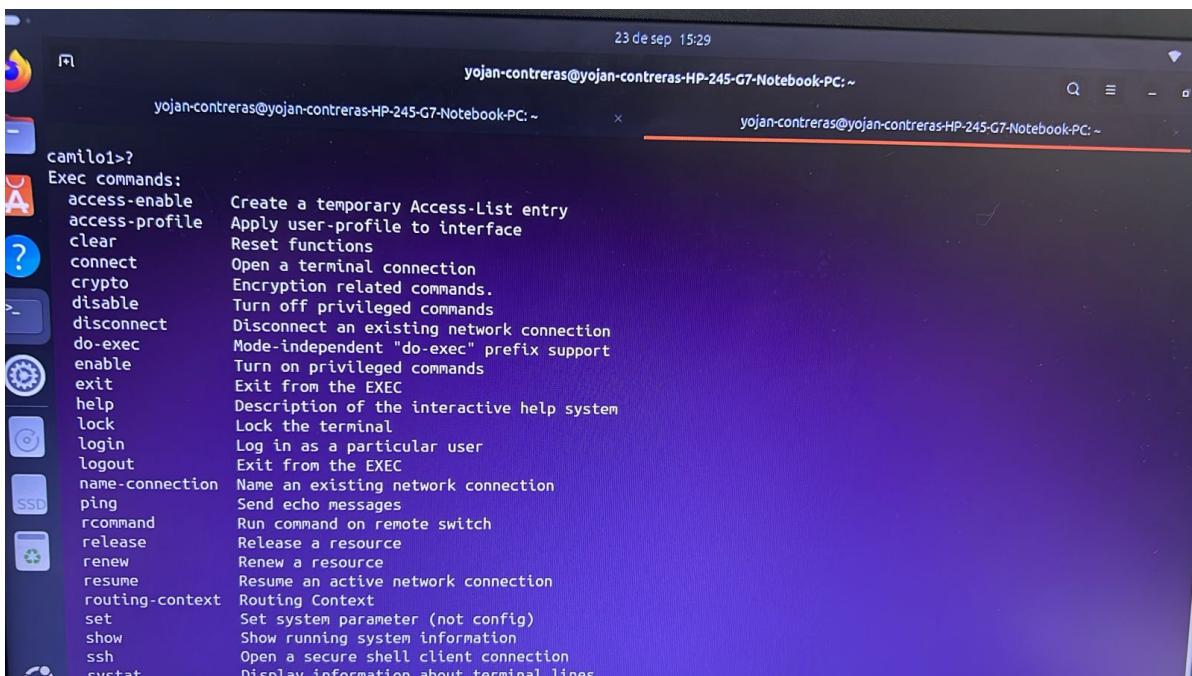


Figure 5: Verificamos el puerto



```
23 de sep 15:29
yojan-contreras@yojan-contreras-HP-245-G7-Notebook-PC: ~
```

Figure 6: Accedemos al switch



```
23 de sep 15:29
yojan-contreras@yojan-contreras-HP-245-G7-Notebook-PC: ~
yojan-contreras@yojan-contreras-HP-245-G7-Notebook-PC: ~
camilo1>?
Exec commands:
access-enable      Create a temporary Access-List entry
access-profile     Apply user-profile to interface
clear              Reset functions
connect             Open a terminal connection
crypto              Encryption related commands.
disable             Turn off privileged commands
disconnect          Disconnect an existing network connection
do-exec             Mode-independent "do-exec" prefix support
enable              Turn on privileged commands
exit                Exit from the EXEC
help                Description of the interactive help system
lock                Lock the terminal
login               Log in as a particular user
logout              Exit from the EXEC
name-connection     Name an existing network connection
ping                Send echo messages
rcommand            Run command on remote switch
release             Release a resource
renew               Renew a resource
resume              Resume an active network connection
routing-context    Routing Context
set                 Set system parameter (not config)
show                Show running system information
ssh                 Open a secure shell client connection
svstat              Display information about terminal lines
```

Figure 7: Observamos los posibles comandos

```
yojan.contreras@yojan-contreras:~$ show ip interface brief
Interface          IP-Address      OK? Method Status      Protocol
Vlan1              192.168.1.10   YES NVRAM up        down
FastEthernet0/1    unassigned     YES unset down     down
FastEthernet0/2    unassigned     YES unset down     down
FastEthernet0/3    unassigned     YES unset down     down
FastEthernet0/4    unassigned     YES unset down     down
FastEthernet0/5    unassigned     YES unset down     down
FastEthernet0/6    unassigned     YES unset down     down
FastEthernet0/7    unassigned     YES unset down     down
FastEthernet0/8    unassigned     YES unset down     down
FastEthernet0/9    unassigned     YES unset down     down
FastEthernet0/10   unassigned     YES unset up       up
FastEthernet0/11   unassigned     YES unset up       up
FastEthernet0/12   unassigned     YES unset up       up
FastEthernet0/13   unassigned     YES unset up       up
FastEthernet0/14   unassigned     YES unset up       up
FastEthernet0/15   unassigned     YES unset down     down
FastEthernet0/16   unassigned     YES unset down     down
FastEthernet0/17   unassigned     YES unset down     down
FastEthernet0/18   unassigned     YES unset down     down
FastEthernet0/19   unassigned     YES unset down     down
FastEthernet0/20   unassigned     YES unset down     down
--More--
```

Figure 8: Observamos el estado de los puertos

```
FastEthernet0/14    unassigned     YES unset up       up
FastEthernet0/15    unassigned     YES unset down     up
FastEthernet0/16    unassigned     YES unset down     down
FastEthernet0/17    unassigned     YES unset down     down
FastEthernet0/18    unassigned     YES unset down     down
FastEthernet0/19    unassigned     YES unset down     down
FastEthernet0/20    unassigned     YES unset down     down
camilo1>show vlan brief
VLAN Name          Status    Ports
-----  

1    default        active   Fa0/23, Fa0/24, Gi0/1, Gi0/2
2    dos            active   Fa0/2, Fa0/3, Fa0/4, Fa0/5
10   VLAN10         active   Fa0/6, Fa0/7, Fa0/8, Fa0/9
                           Fa0/10
30   VLAN30         active   Fa0/11, Fa0/12, Fa0/13, Fa0/14
                           Fa0/15
40   VLAN40         active   Fa0/16, Fa0/17, Fa0/18, Fa0/19
                           Fa0/20
50   VLAN50         active   Fa0/21, Fa0/22
1002 fddi-default  act/unsup
1003 token-ring-default  act/unsup
1004 fddinet-default  act/unsup
1005 trnet-default   act/unsup
camilo1>
camilo1>
```

Figure 9: Resumen rapido de las Vlans

The screenshot shows a terminal window with the following content:

```
yojan-contreras@yojan-contreras-HP-245-G7-Notebook-PC:~  
camilo1>show mac address-table  
Mac Address Table  
-----  
Vlan   Mac Address      Type    Ports  
---  
All    0100.0ccc.cccc  STATIC  CPU  
All    0100.0ccc.cccd  STATIC  CPU  
All    0180.c200.0000  STATIC  CPU  
All    0180.c200.0001  STATIC  CPU  
All    0180.c200.0002  STATIC  CPU  
All    0180.c200.0003  STATIC  CPU  
All    0180.c200.0004  STATIC  CPU  
All    0180.c200.0005  STATIC  CPU  
All    0180.c200.0006  STATIC  CPU  
All    0180.c200.0007  STATIC  CPU  
All    0180.c200.0008  STATIC  CPU  
All    0180.c200.0009  STATIC  CPU  
All    0180.c200.000a  STATIC  CPU  
All    0180.c200.000b  STATIC  CPU  
All    0180.c200.000c  STATIC  CPU  
All    0180.c200.000d  STATIC  CPU  
All    0180.c200.000e  STATIC  CPU  
All    0180.c200.000f  STATIC  CPU  
All    0180.c200.0010  STATIC  CPU  
All    ffff.ffff.ffff  STATIC  CPU  
Total Mac Addresses for this criterion: 20  
camilo1>
```

Figure 10: Visualización de la dirección Mac que ha aprendido el switch y en qué puerto físico lo está viendo conectado.

Port	Name	Status	Vlan	Duplex	Speed	Type
Fa0/1		notconnect	1	auto	auto	10/100BaseTX
Fa0/2		notconnect	10	auto	auto	10/100BaseTX
Fa0/3		notconnect	10	auto	auto	10/100BaseTX
Fa0/4		notconnect	10	auto	auto	10/100BaseTX
Fa0/5		notconnect	10	auto	auto	10/100BaseTX
Fa0/6		notconnect	20	auto	auto	10/100BaseTX
Fa0/7		notconnect	20	auto	auto	10/100BaseTX
Fa0/8		notconnect	20	auto	auto	10/100BaseTX
Fa0/9		notconnect	20	auto	auto	10/100BaseTX
Fa0/10		notconnect	20	auto	auto	10/100BaseTX
Fa0/11		connected	30	a-full	a-100	10/100BaseTX
Fa0/12		connected	30	a-full	a-100	10/100BaseTX
Fa0/13		connected	30	a-full	a-100	10/100BaseTX
Fa0/14		connected	30	a-full	a-10	10/100BaseTX
Fa0/15		notconnect	30	auto	auto	10/100BaseTX
Fa0/16		notconnect	40	auto	auto	10/100BaseTX
Fa0/17		notconnect	40	auto	auto	10/100BaseTX
Fa0/18		notconnect	40	auto	auto	10/100BaseTX
Fa0/19		notconnect	40	auto	auto	10/100BaseTX
Fa0/20		notconnect	40	auto	auto	10/100BaseTX
Fa0/21		notconnect	50	auto	auto	10/100BaseTX
Fa0/22		notconnect	50	auto	auto	10/100BaseTX

Figure 11: Verificación de la conexión

3.2. Configuración de red en Laptop 1

```
ip addr show
sudo ip addr add 192.168.X.X/24 dev eno1
sudo ip link set dev eno1 up
ping -c 4 192.168.X.X
```

```
yojan-contreras@yojan-contreras-HP-245-G7-Notebook-PC: ~
yojan-contreras@yojan-contreras-HP-245-G7-Notebook-PC: ~
yojan-contreras@yojan-contreras-HP-245-G7-Notebook-PC: ~ $ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: eno1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether f8:0d:ac:49:b9:b0 brd ff:ff:ff:ff:ff:ff
    altname enp2s0
    inet 192.168.10.10/24 scope global eno1
        valid_lft forever preferred_lft forever
3: wlp3s0: <BROADCAST,MULTICAST> mtu 1500 qdisc noqueue state DOWN group default qlen 1000
    link/ether 5c:3a:45:5d:ba:83 brd ff:ff:ff:ff:ff:ff
4: virbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default qlen 1000
    link/ether 52:54:00:38:77:28 brd ff:ff:ff:ff:ff:ff
    inet 192.168.122.1/24 brd 192.168.122.255 scope global virbr0
        valid_lft forever preferred_lft forever
yojan-contreras@yojan-contreras-HP-245-G7-Notebook-PC: ~
```

Figure 12: Visualizamos la Ip del Laptop1

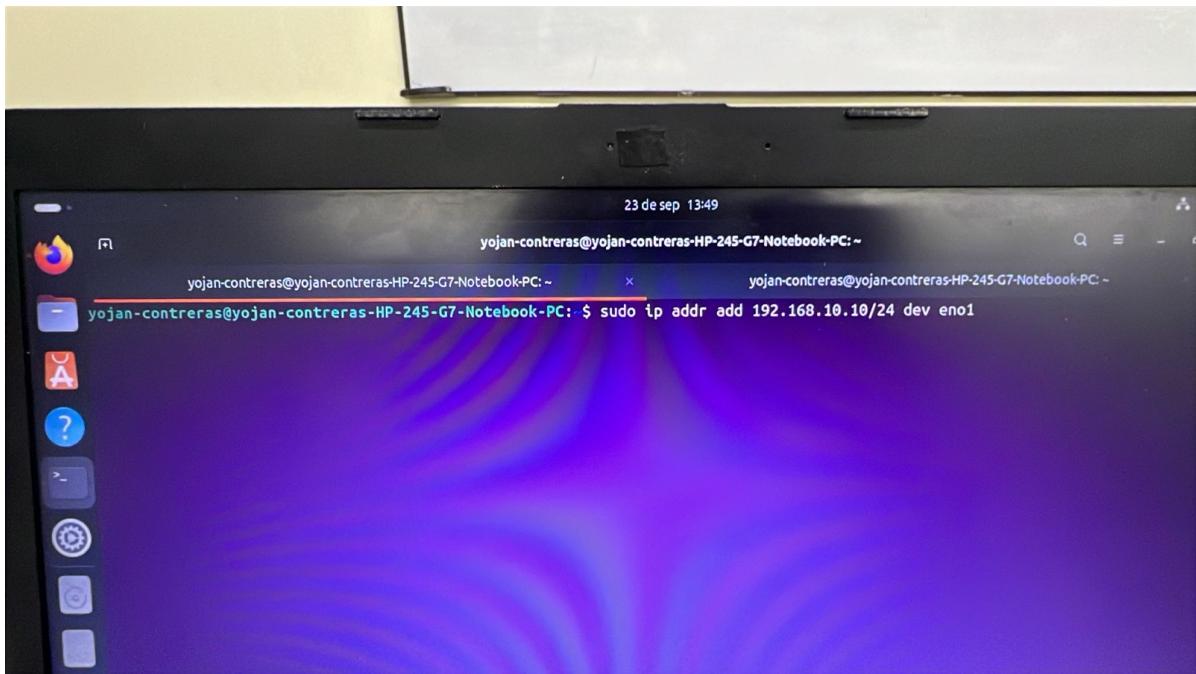


Figure 13: Asiganmos Ip al Laptop1

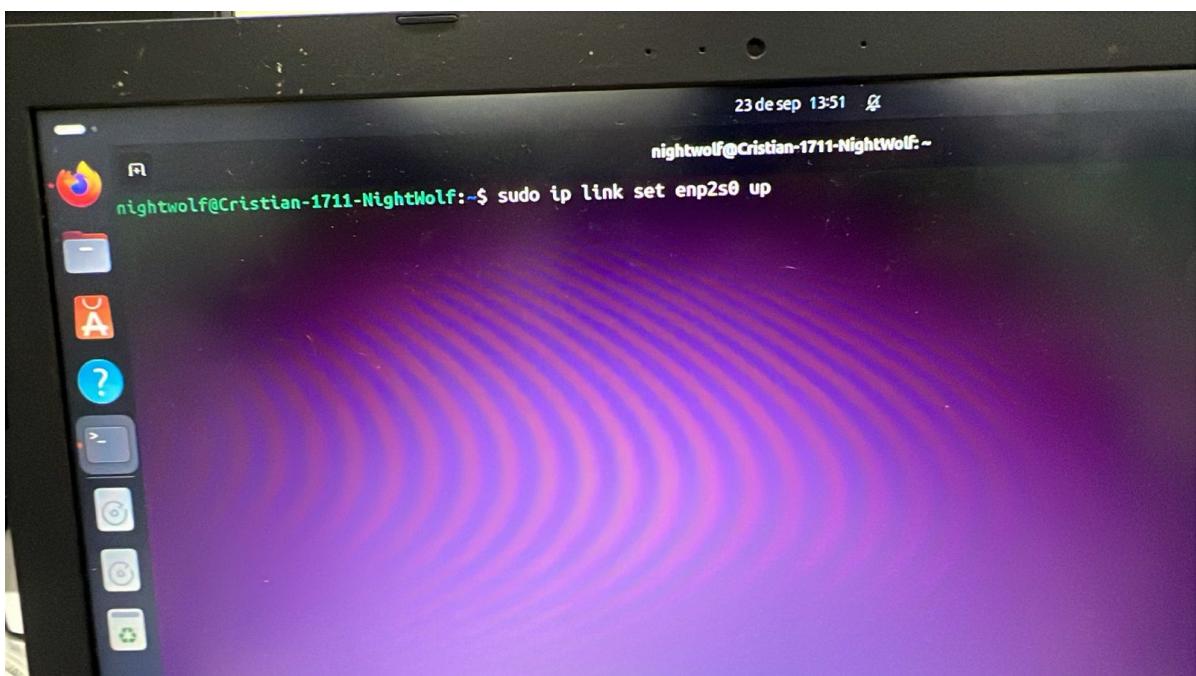


Figure 14: Activamos el puerto que usaremos

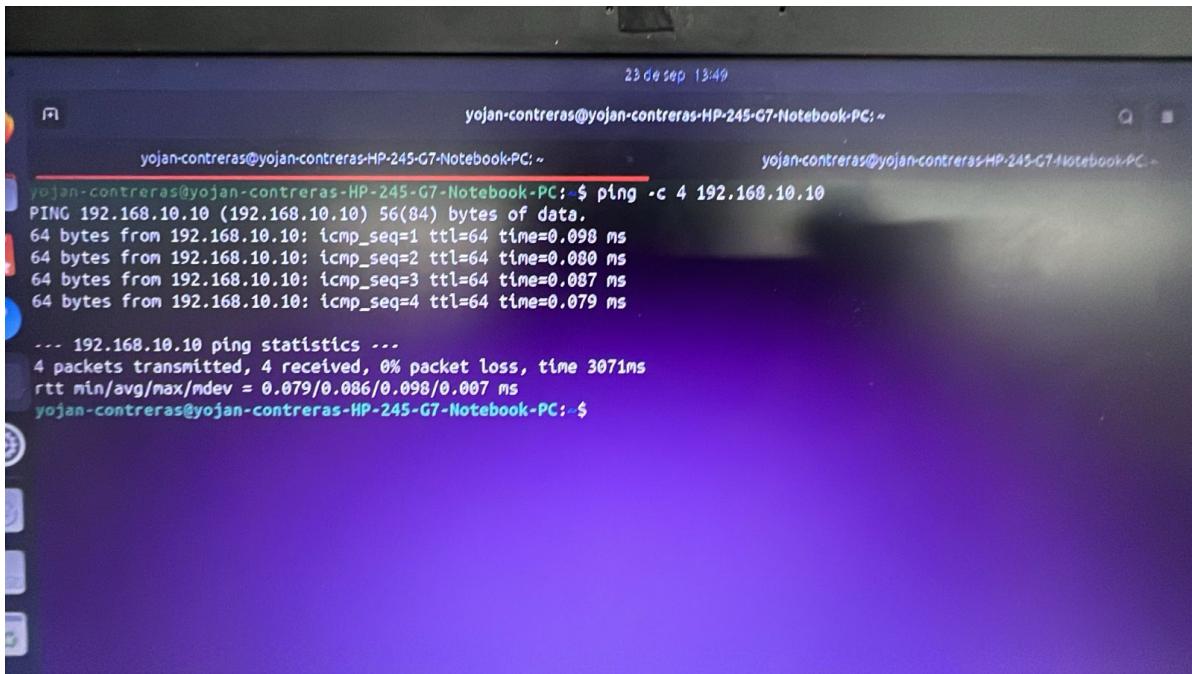


Figure 15: Probamos con Ping

3.3. Configuración de red en Laptop 2

```
ip addr show
sudo ip addr add 192.168.X.X/24 dev enp2s0
sudo ip link set dev enp2s0 up
ping -c 4 192.168.X.X
```

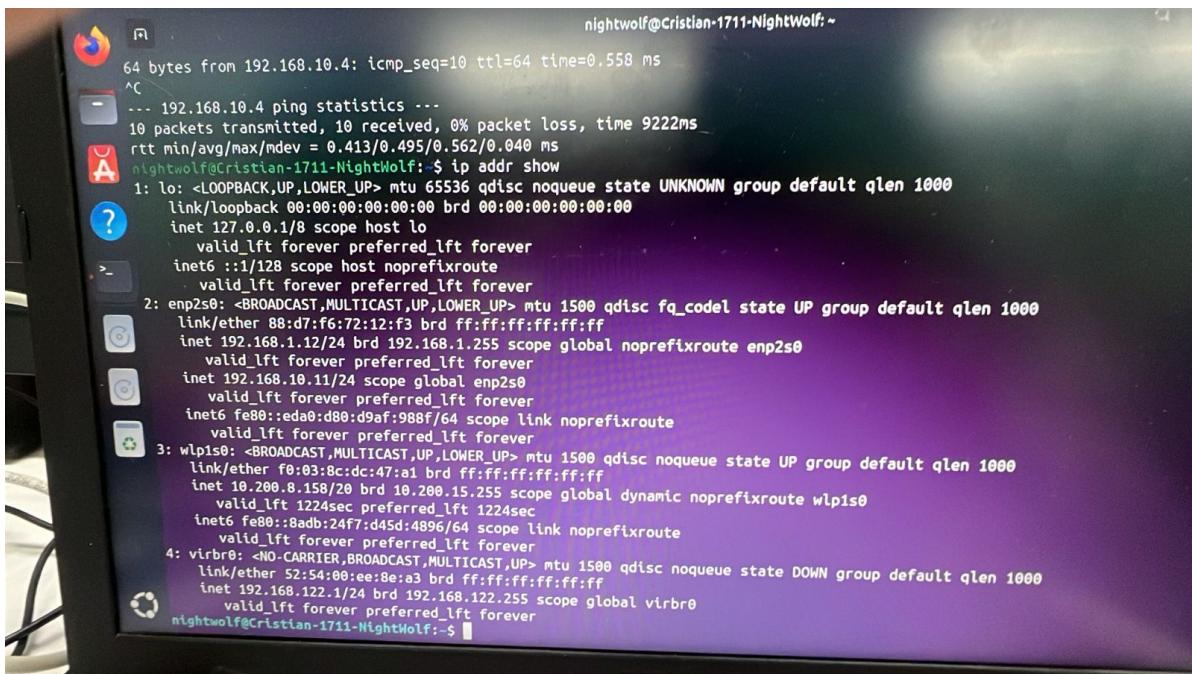


Figure 16: Ip del Laptop2

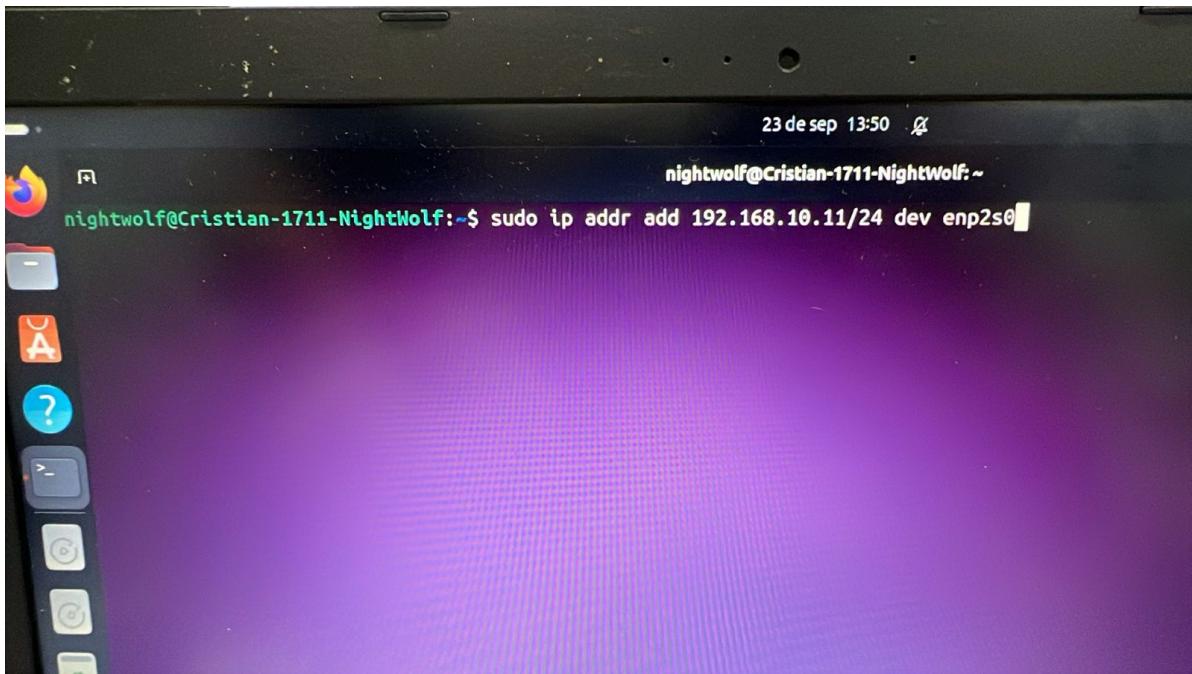


Figure 17: Asignamos Ip a el Laptop2

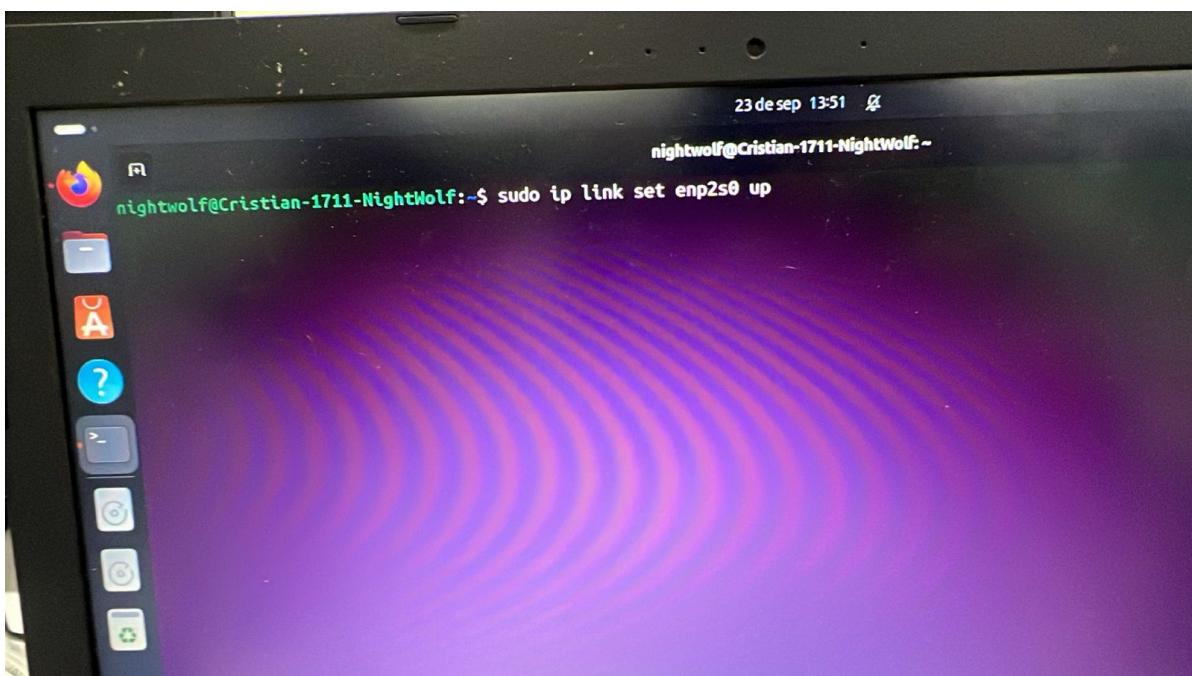


Figure 18: Activamos el puerto que usaremos del laptop2

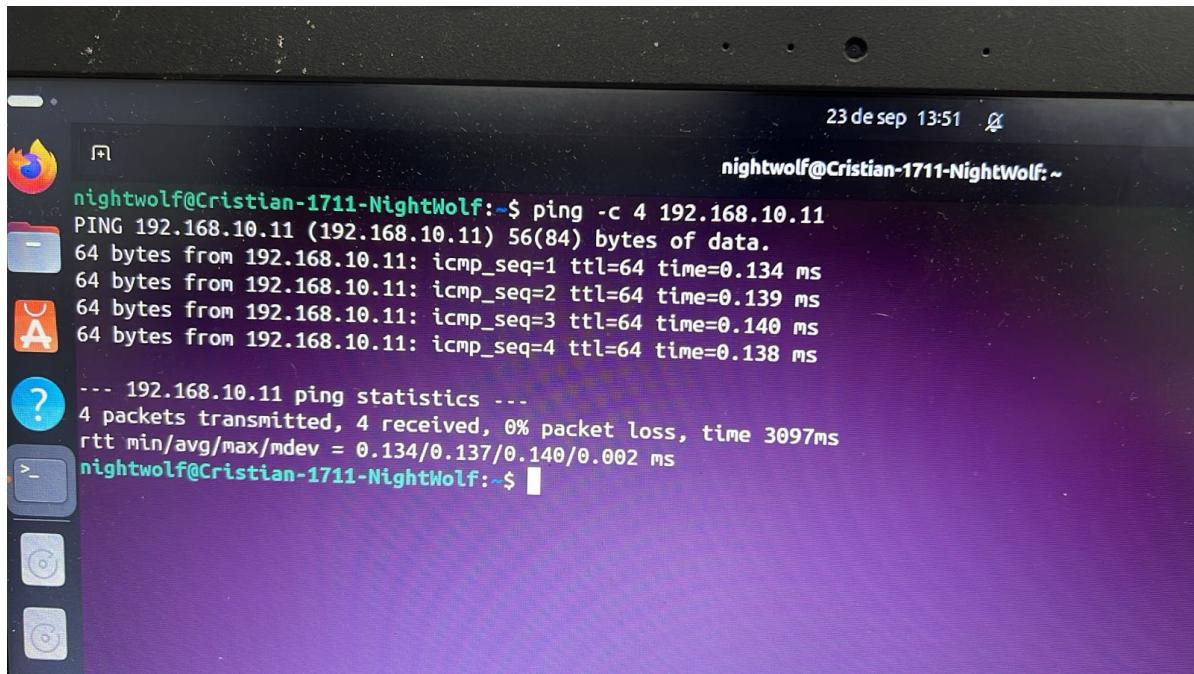


Figure 19: Probamos con Ping del laptop2

3.4. Configuración de red en la Raspberry Pi

```

ip addr show
sudo ip addr add 192.168.X.X/24 dev eth0
sudo ip link set dev eth0 up
ping -c 4 192.168.X.X

```

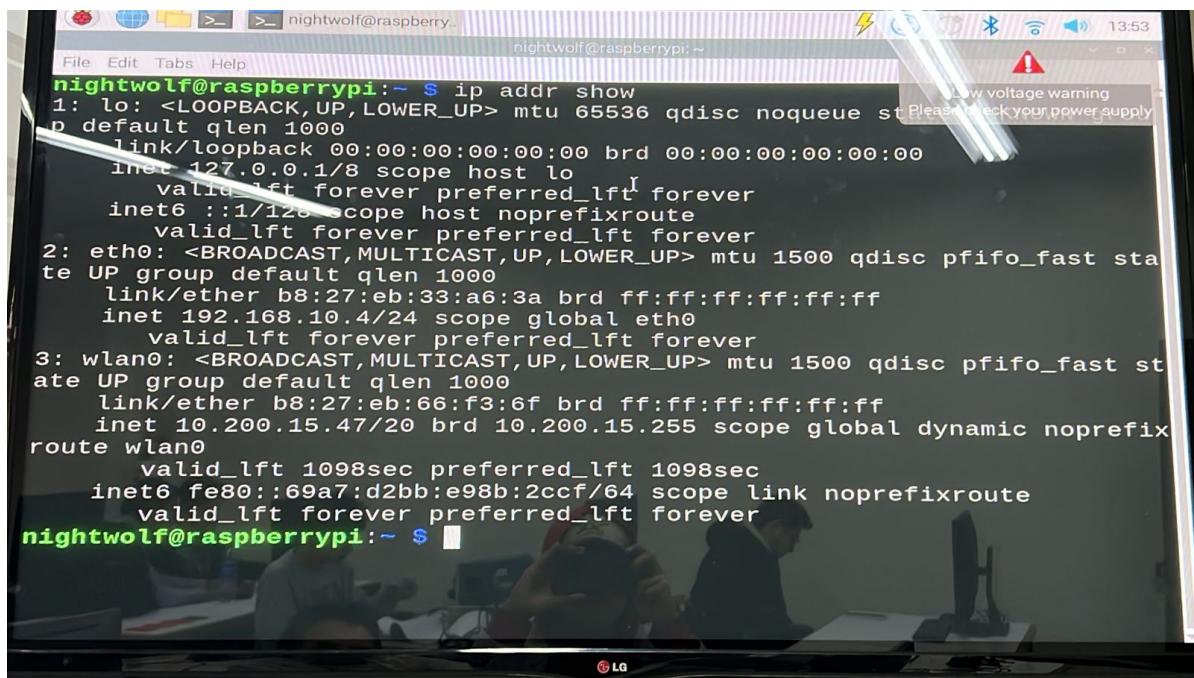


Figure 20: Visualizamos la Ip de la raspberry

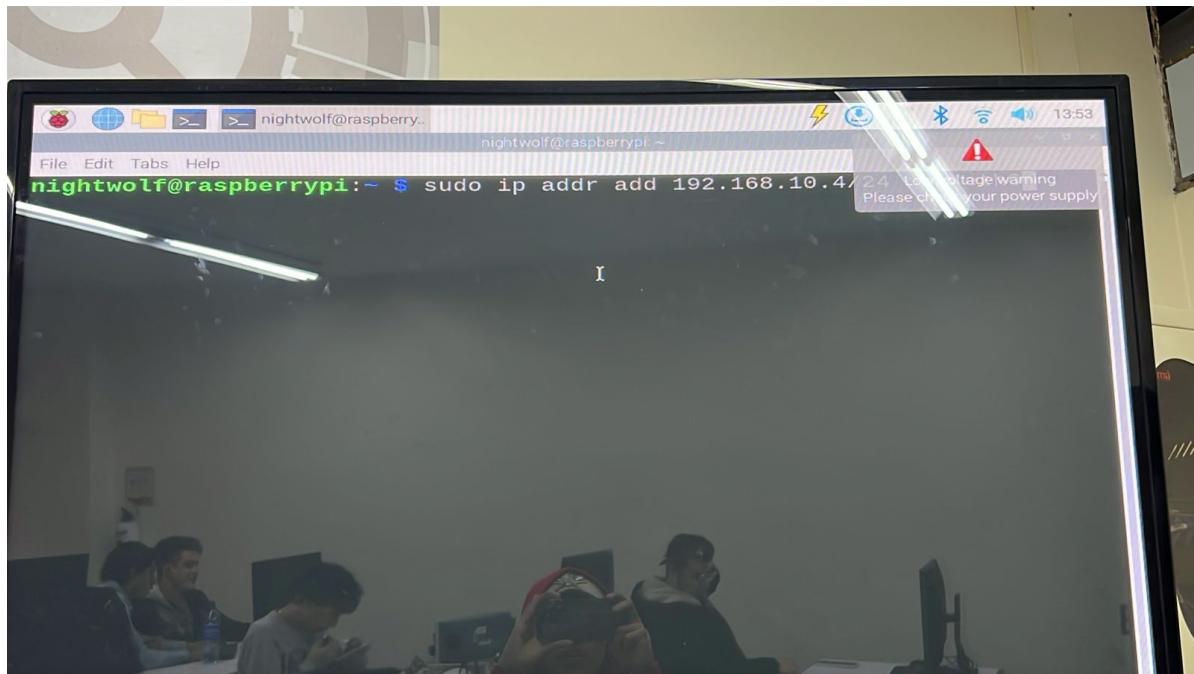


Figure 21: Asignamos Ip a la Raspberri

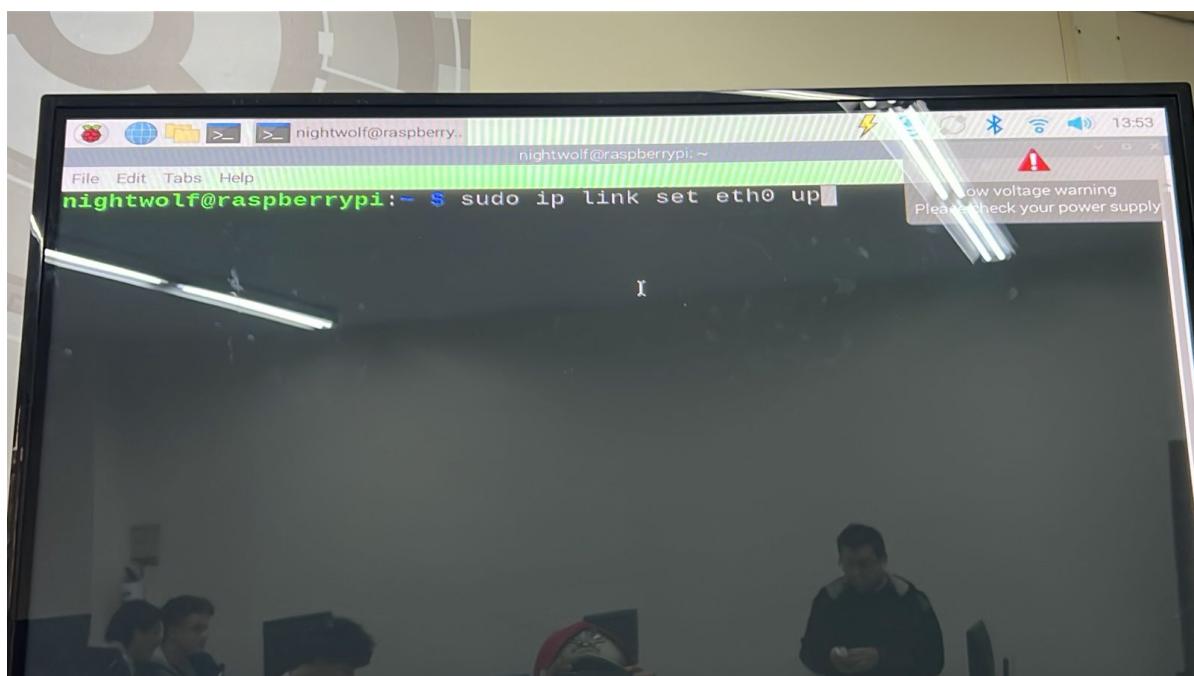


Figure 22: Activamos puerto que usaremos de la Raspberri

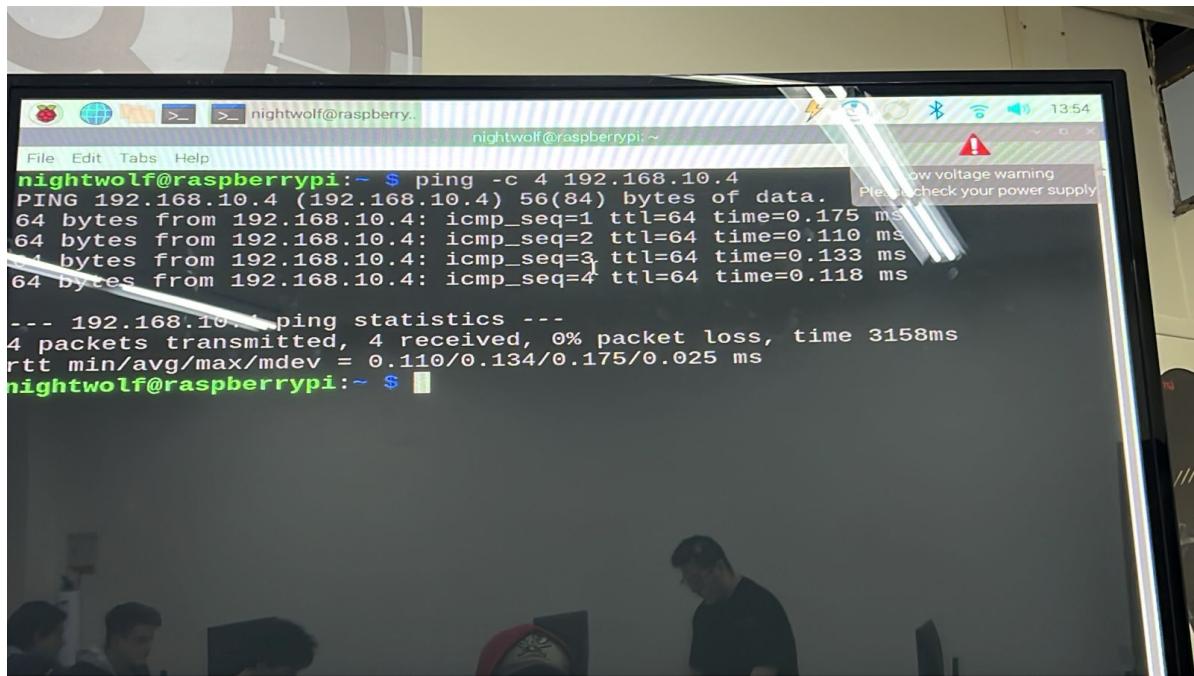


Figure 23: Verificamos con ping la raspberry

3.5. Configuración del PC del laboratorio (Windows)

1. Abrir Panel de control → Centro de redes y recursos compartidos.
2. Seleccionar la conexión por cable y abrir Propiedades.
3. Seleccionar “Protocolo de Internet versión 4 (TCP/IPv4)” y establecer IP manual.

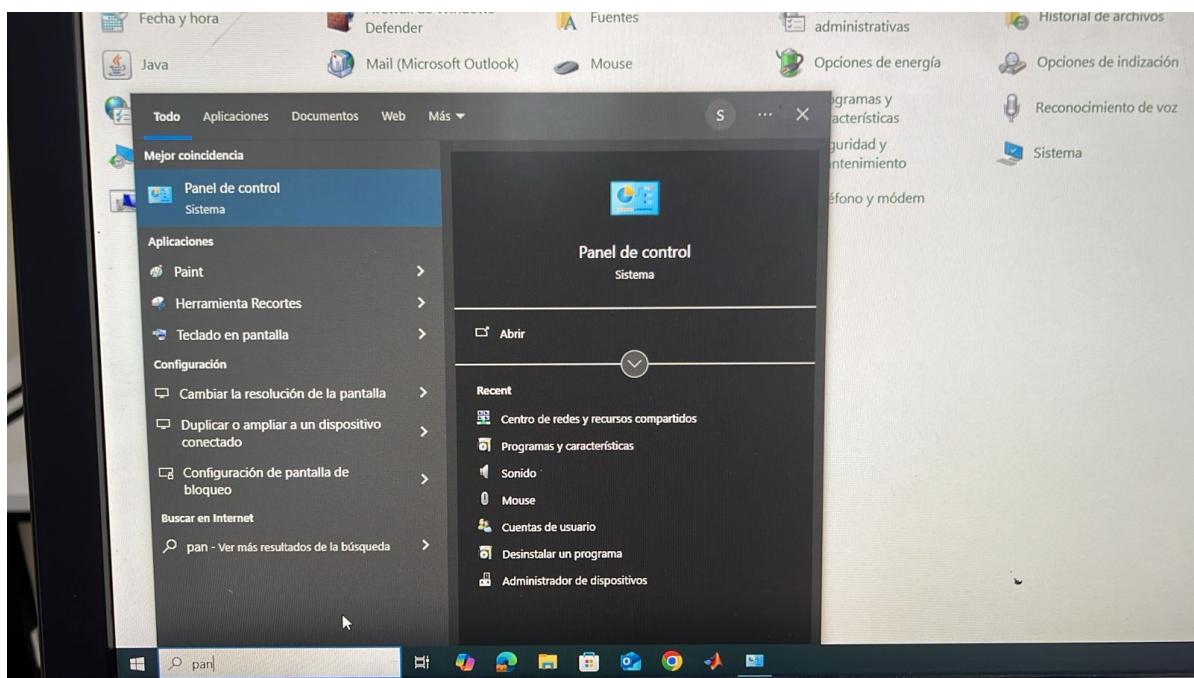


Figure 24: Accedemos al panel de control

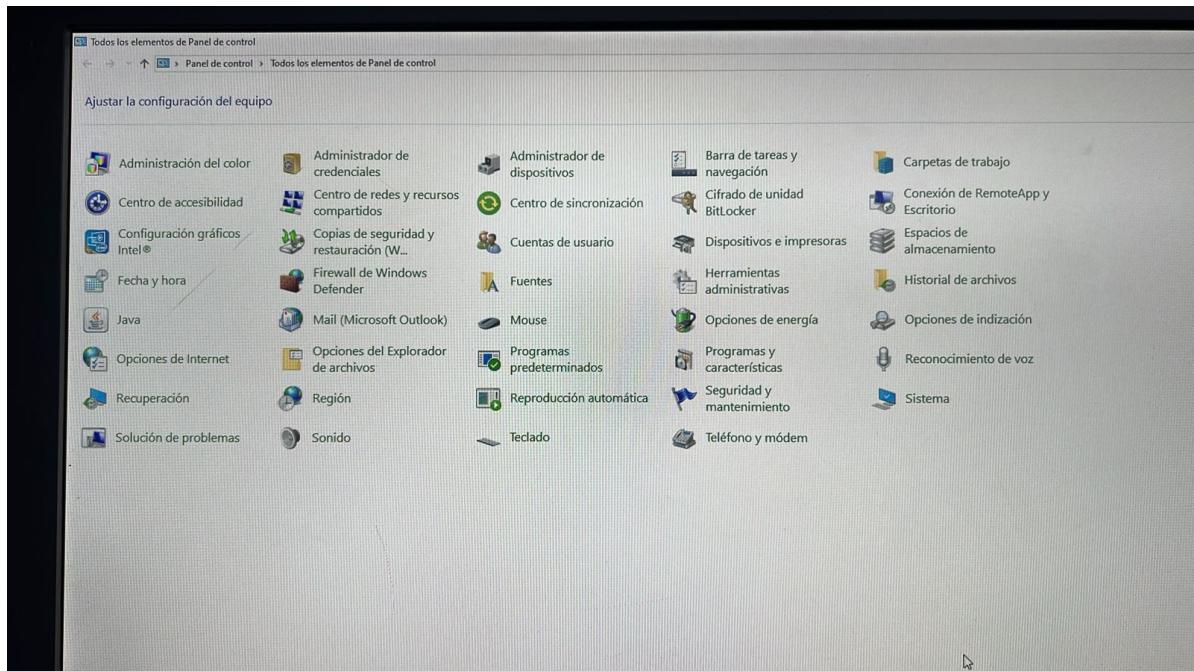


Figure 25: Accedemos a redes y recursos

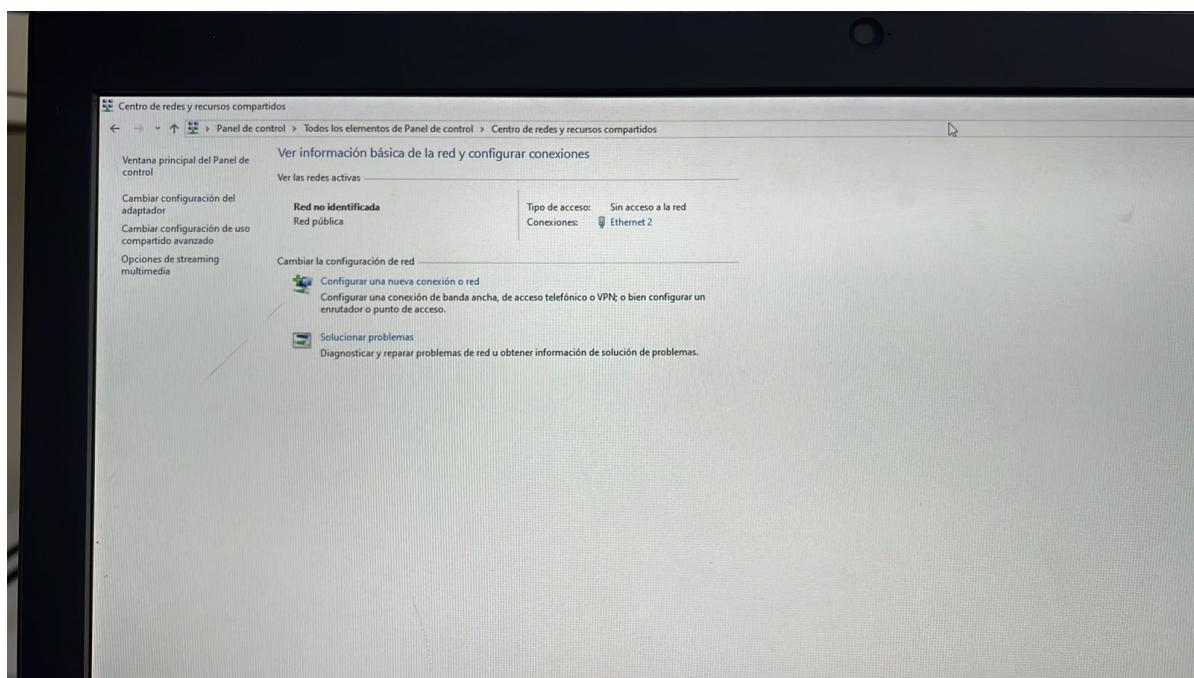


Figure 26: Seleccionamos Cable

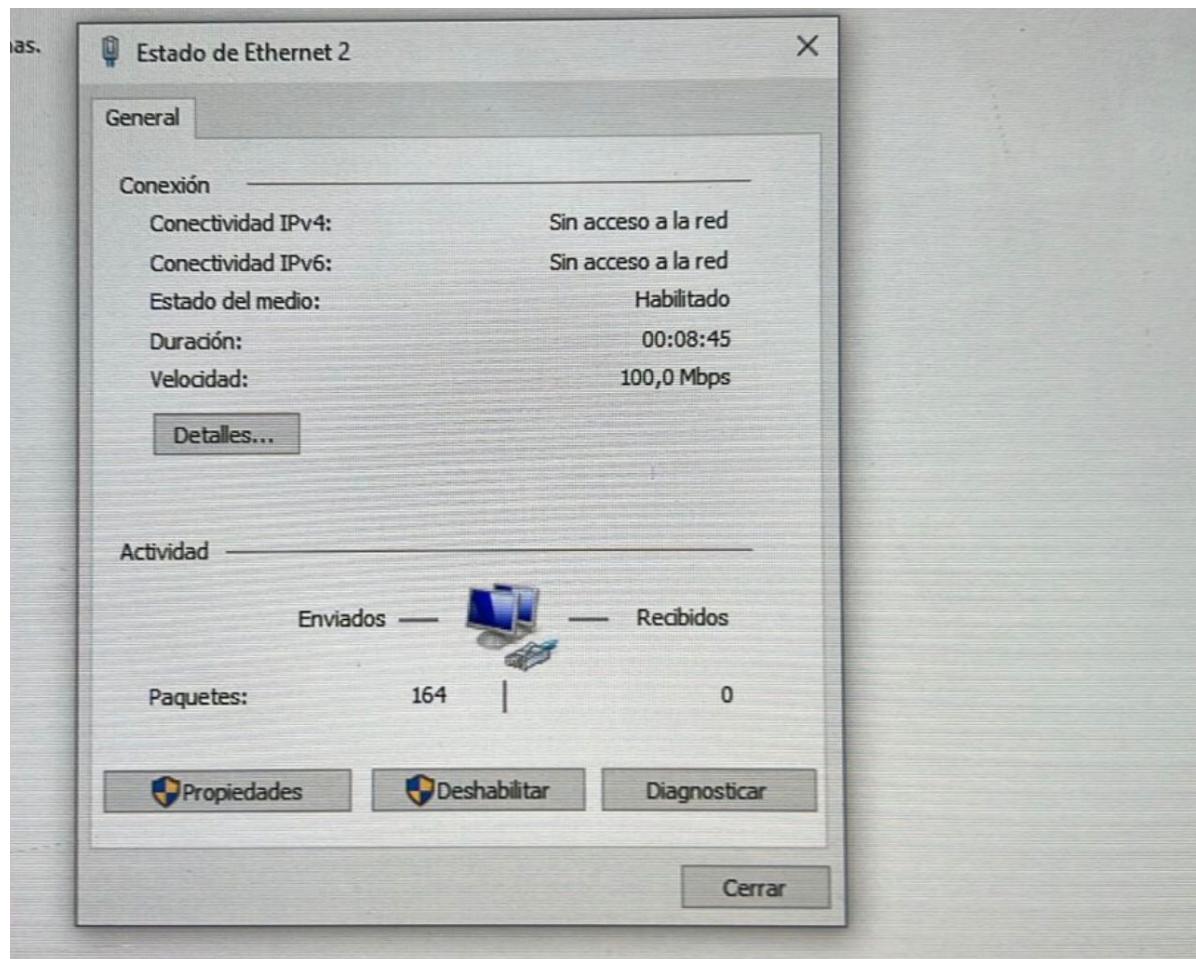


Figure 27: Accedemos a Propiedades

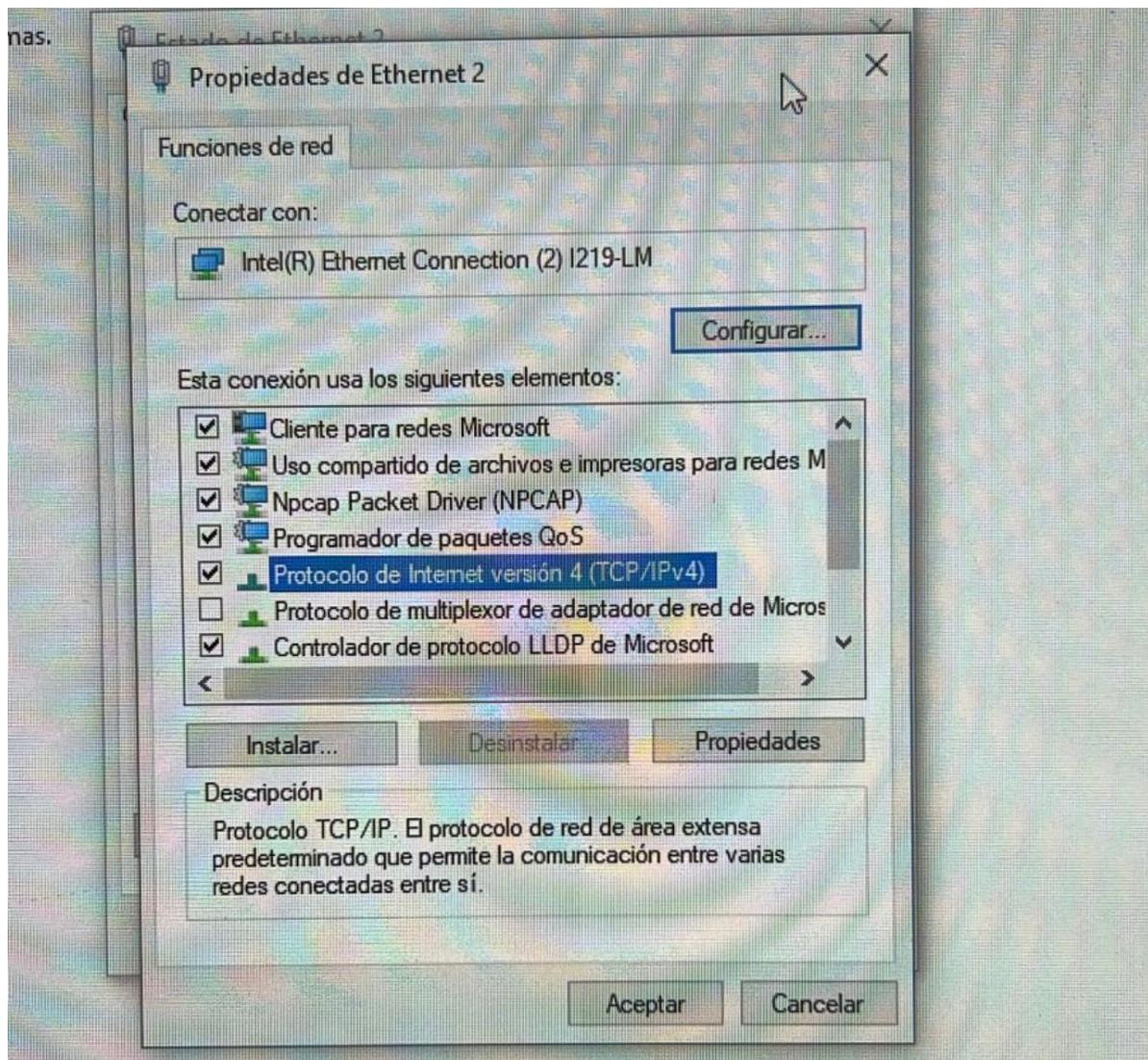


Figure 28: Seleccionamos Protocolo de internet version 4 y accedemos a propiedades

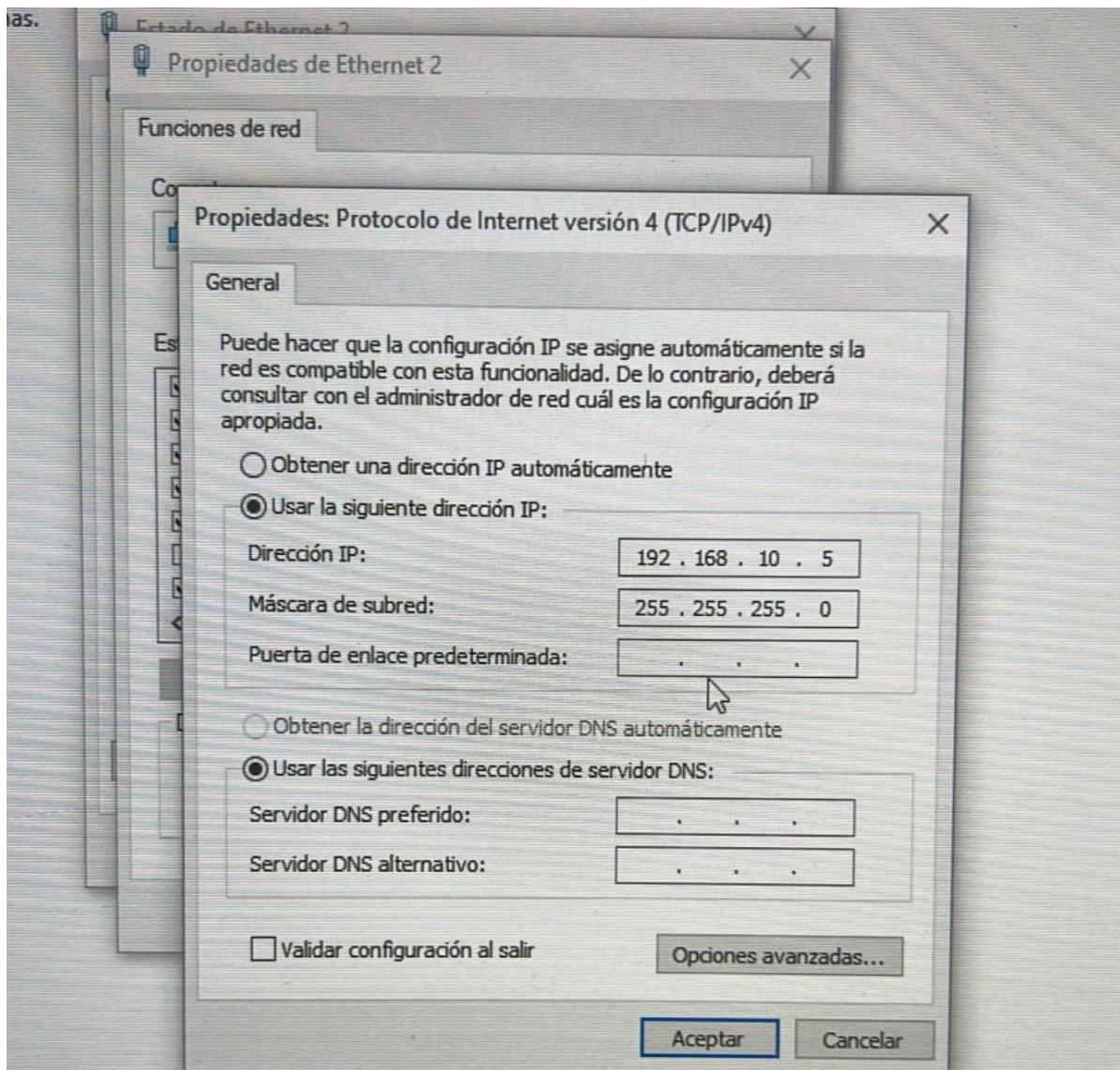


Figure 29: Le asignamos una Ip al Pc-ethm

Para permitir respuesta a pings:

- Abrir wf.msc.
- Ir a Reglas de entrada.
- Habilitar “Archivos e impresoras compartidos”.

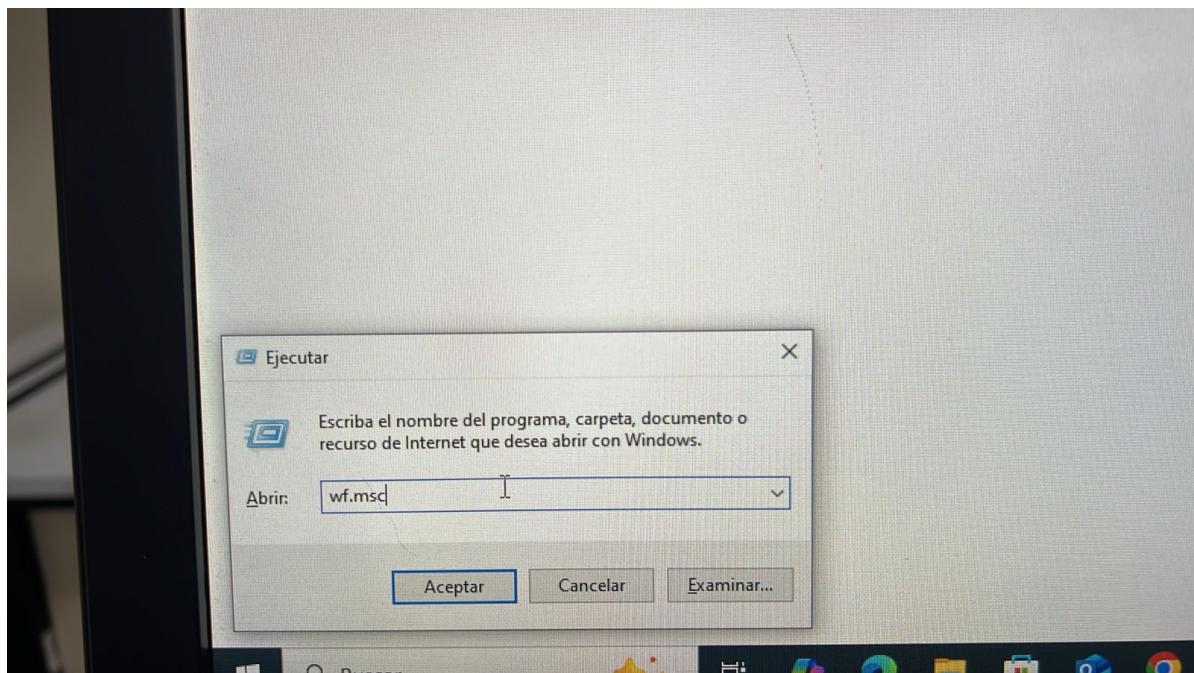


Figure 30: Ingresamos a reglas de entrada

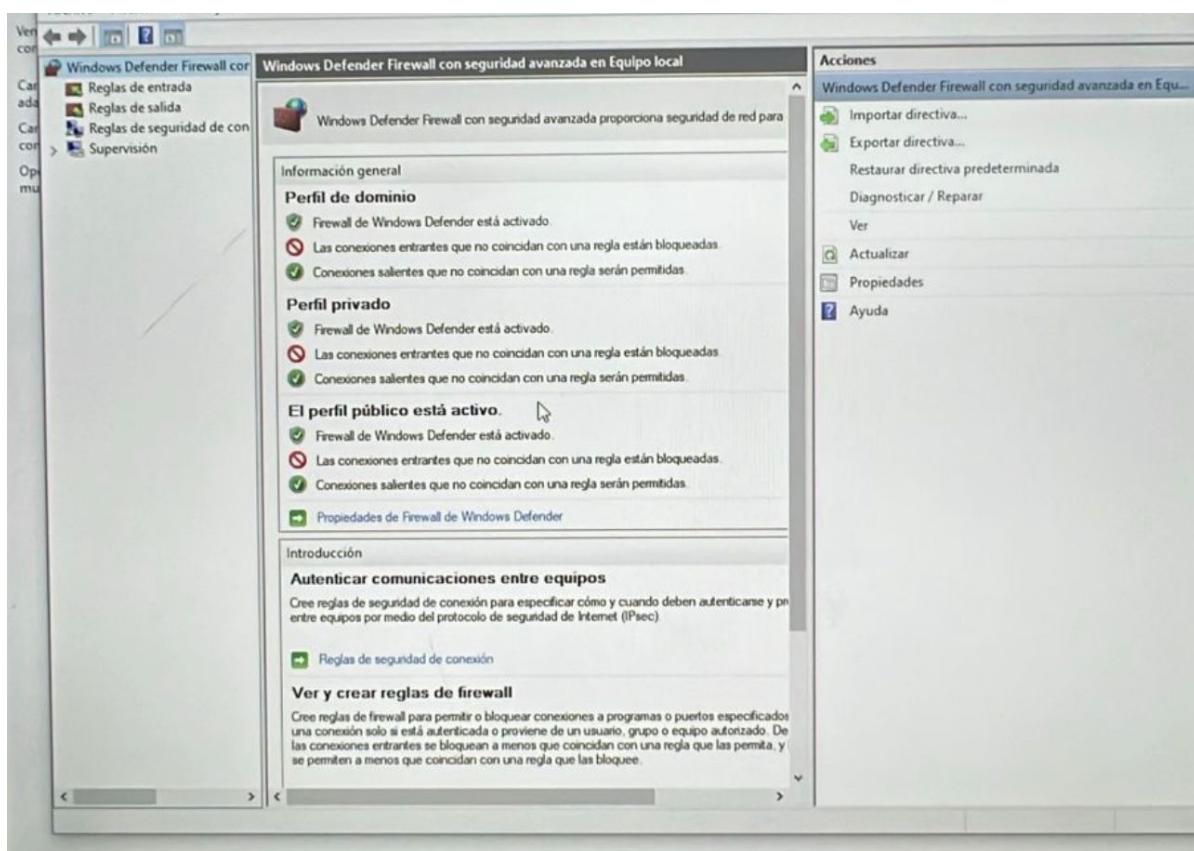


Figure 31: Le asignamos una Ip al Pc-etm

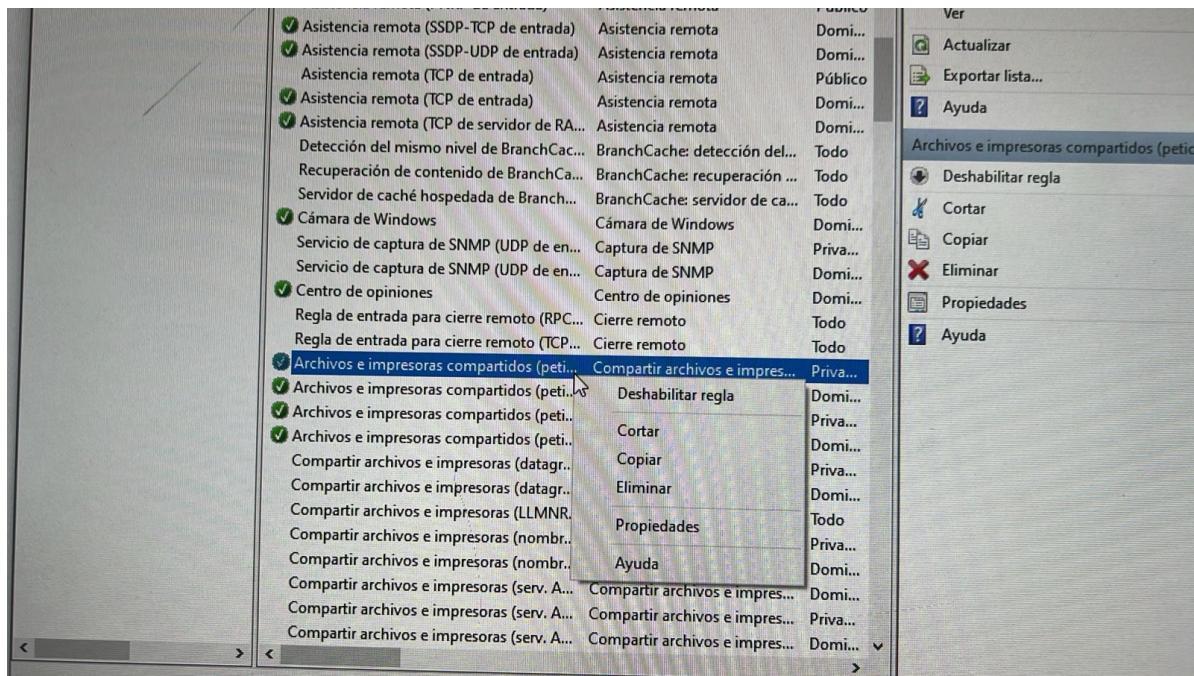


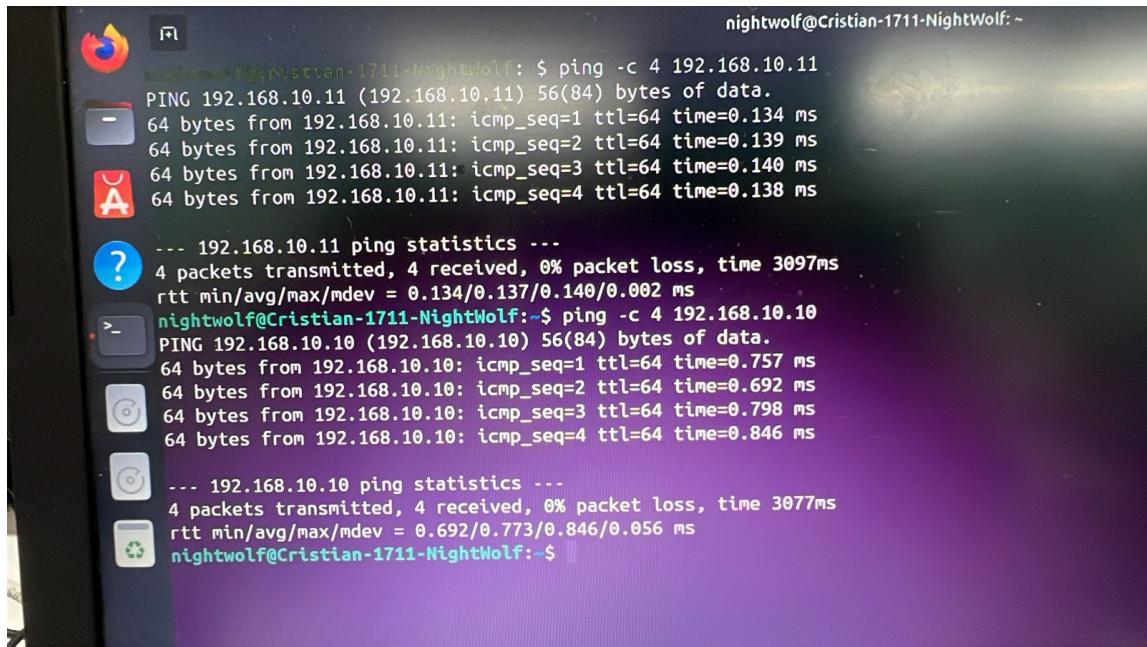
Figure 32: Habilitamos las 4 opciones de Archivos e impresoras compartidos

3.6. Prueba de conectividad entre todos los dispositivos

Una vez configuradas las direcciones IP de todos los equipos, se realizaron pruebas de conectividad entre ellos mediante el comando `ping`. Cada equipo fue probado contra los demás para comprobar que todos los nodos en la red se comunicaban correctamente.

```
ping -c 4 192.168.X.X
```

A continuación se muestran las pruebas de conectividad entre todos los dispositivos:

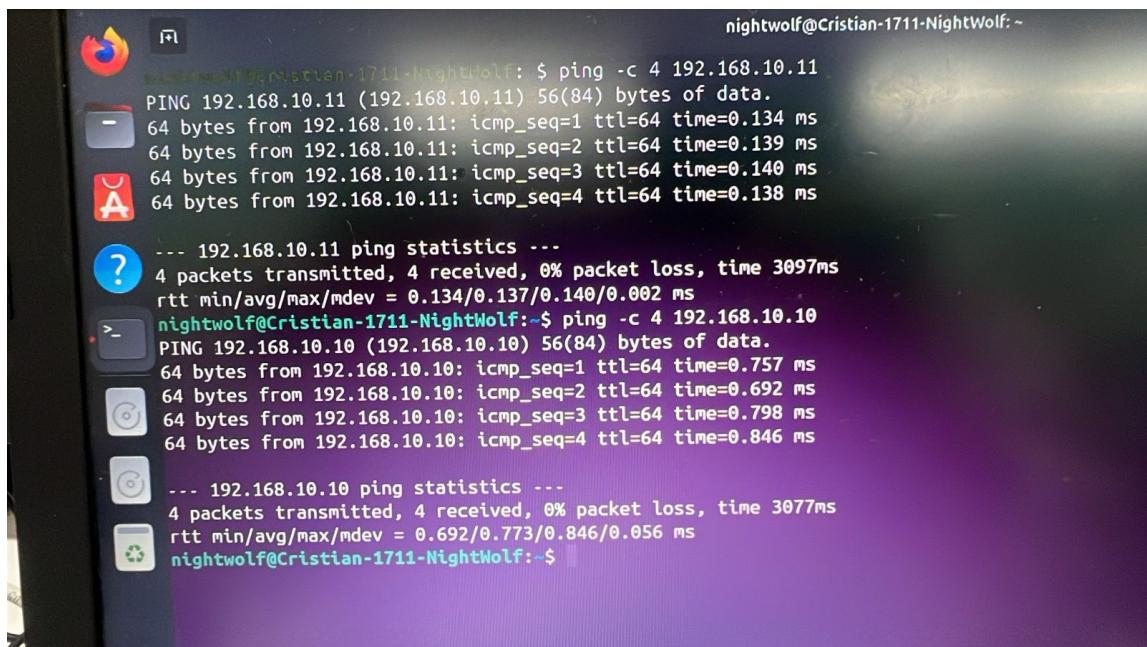


```
nightwolf@Cristian-1711-NightWolf: ~
christian@cristian-1711-NightWolf: $ ping -c 4 192.168.10.11
PING 192.168.10.11 (192.168.10.11) 56(84) bytes of data.
64 bytes from 192.168.10.11: icmp_seq=1 ttl=64 time=0.134 ms
64 bytes from 192.168.10.11: icmp_seq=2 ttl=64 time=0.139 ms
64 bytes from 192.168.10.11: icmp_seq=3 ttl=64 time=0.140 ms
64 bytes from 192.168.10.11: icmp_seq=4 ttl=64 time=0.138 ms

--- 192.168.10.11 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3097ms
rtt min/avg/max/mdev = 0.134/0.137/0.140/0.002 ms
nightwolf@Cristian-1711-NightWolf:~$ ping -c 4 192.168.10.10
PING 192.168.10.10 (192.168.10.10) 56(84) bytes of data.
64 bytes from 192.168.10.10: icmp_seq=1 ttl=64 time=0.757 ms
64 bytes from 192.168.10.10: icmp_seq=2 ttl=64 time=0.692 ms
64 bytes from 192.168.10.10: icmp_seq=3 ttl=64 time=0.798 ms
64 bytes from 192.168.10.10: icmp_seq=4 ttl=64 time=0.846 ms

--- 192.168.10.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3077ms
rtt min/avg/max/mdev = 0.692/0.773/0.846/0.056 ms
nightwolf@Cristian-1711-NightWolf:~$
```

Figure 33: Ping de Laptop 1 a Laptop 2



```
nightwolf@Cristian-1711-NightWolf: ~
christian@cristian-1711-NightWolf: $ ping -c 4 192.168.10.11
PING 192.168.10.11 (192.168.10.11) 56(84) bytes of data.
64 bytes from 192.168.10.11: icmp_seq=1 ttl=64 time=0.134 ms
64 bytes from 192.168.10.11: icmp_seq=2 ttl=64 time=0.139 ms
64 bytes from 192.168.10.11: icmp_seq=3 ttl=64 time=0.140 ms
64 bytes from 192.168.10.11: icmp_seq=4 ttl=64 time=0.138 ms

--- 192.168.10.11 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3097ms
rtt min/avg/max/mdev = 0.134/0.137/0.140/0.002 ms
nightwolf@Cristian-1711-NightWolf:~$ ping -c 4 192.168.10.10
PING 192.168.10.10 (192.168.10.10) 56(84) bytes of data.
64 bytes from 192.168.10.10: icmp_seq=1 ttl=64 time=0.757 ms
64 bytes from 192.168.10.10: icmp_seq=2 ttl=64 time=0.692 ms
64 bytes from 192.168.10.10: icmp_seq=3 ttl=64 time=0.798 ms
64 bytes from 192.168.10.10: icmp_seq=4 ttl=64 time=0.846 ms

--- 192.168.10.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3077ms
rtt min/avg/max/mdev = 0.692/0.773/0.846/0.056 ms
nightwolf@Cristian-1711-NightWolf:~$
```

Figure 34: Ping de Laptop 2 a Laptop 1



Figure 35: Ping de Raspberry Pi a Laptop 1

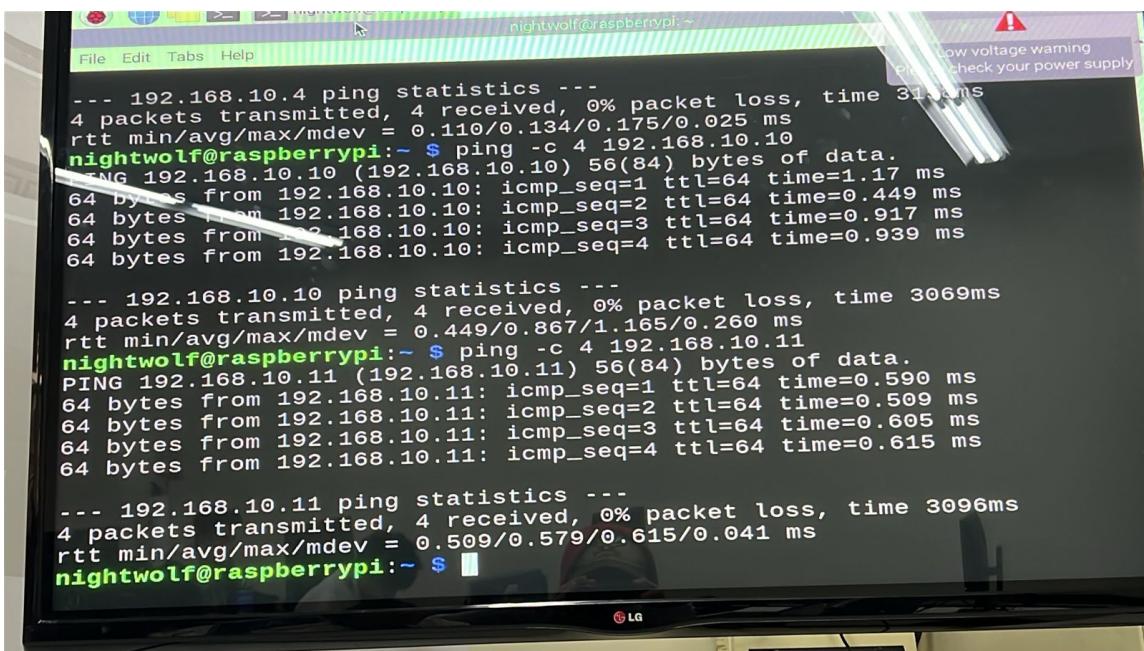


Figure 36: Ping de Raspberry Pi a Laptop 2

```

A 64 bytes from 192.168.10.10: icmp_seq=3 ttl=64 time=0.087 ms
A 64 bytes from 192.168.10.10: icmp_seq=4 ttl=64 time=0.079 ms

? --- 192.168.10.10 ping statistics ---
? 4 packets transmitted, 4 received, 0% packet loss, time 3071ms
? rtt min/avg/max/mdev = 0.079/0.086/0.098/0.007 ms
>_ yojan-contreras@yojan-contreras-HP-245-G7-Notebook-PC:~$ ping -c 4 192.168.10.11
PING 192.168.10.11 (192.168.10.11) 56(84) bytes of data.
64 bytes from 192.168.10.11: icmp_seq=1 ttl=64 time=0.703 ms
64 bytes from 192.168.10.11: icmp_seq=2 ttl=64 time=0.414 ms
64 bytes from 192.168.10.11: icmp_seq=3 ttl=64 time=0.319 ms
64 bytes from 192.168.10.11: icmp_seq=4 ttl=64 time=0.310 ms

? --- 192.168.10.11 ping statistics ---
? 4 packets transmitted, 4 received, 0% packet loss, time 3107ms
? rtt min/avg/max/mdev = 0.310/0.436/0.703/0.159 ms
SSD yojan-contreras@yojan-contreras-HP-245-G7-Notebook-PC:~$ ping -c 4 192.168.10.4
PING 192.168.10.4 (192.168.10.4) 56(84) bytes of data.
64 bytes from 192.168.10.4: icmp_seq=1 ttl=64 time=0.390 ms
64 bytes from 192.168.10.4: icmp_seq=2 ttl=64 time=0.463 ms
64 bytes from 192.168.10.4: icmp_seq=3 ttl=64 time=0.471 ms
64 bytes from 192.168.10.4: icmp_seq=4 ttl=64 time=0.490 ms

? --- 192.168.10.4 ping statistics ---
? 4 packets transmitted, 4 received, 0% packet loss, time 3085ms
? rtt min/avg/max/mdev = 0.390/0.453/0.490/0.037 ms
yojan-contreras@yojan-contreras-HP-245-G7-Notebook-PC:~$
```

Figure 37: Ping de Laptop 1 a Raspberry Pi

```

A 64 bytes from 192.168.10.11: icmp_seq=3 ttl=64 time=0.137 ms
A 64 bytes from 192.168.10.11: icmp_seq=4 ttl=64 time=0.138 ms
A 64 bytes from 192.168.10.11: icmp_seq=5 ttl=64 time=0.138 ms

? --- 192.168.10.11 ping statistics ---
? 4 packets transmitted, 4 received, 0% packet loss, time 3097ms
? rtt min/avg/max/mdev = 0.134/0.137/0.140/0.002 ms
>_ nightwolf@Cristian-1711-NightWolf:~$ ping -c 4 192.168.10.10
PING 192.168.10.10 (192.168.10.10) 56(84) bytes of data.
64 bytes from 192.168.10.10: icmp_seq=1 ttl=64 time=0.757 ms
64 bytes from 192.168.10.10: icmp_seq=2 ttl=64 time=0.692 ms
64 bytes from 192.168.10.10: icmp_seq=3 ttl=64 time=0.798 ms
64 bytes from 192.168.10.10: icmp_seq=4 ttl=64 time=0.846 ms

? --- 192.168.10.10 ping statistics ---
? 4 packets transmitted, 4 received, 0% packet loss, time 3077ms
? rtt min/avg/max/mdev = 0.692/0.773/0.846/0.056 ms
nightwolf@Cristian-1711-NightWolf:~$ ping -c 4 192.168.10.4
PING 192.168.10.4 (192.168.10.4) 56(84) bytes of data.
64 bytes from 192.168.10.4: icmp_seq=1 ttl=64 time=0.549 ms
64 bytes from 192.168.10.4: icmp_seq=2 ttl=64 time=0.504 ms
64 bytes from 192.168.10.4: icmp_seq=3 ttl=64 time=0.613 ms
64 bytes from 192.168.10.4: icmp_seq=4 ttl=64 time=0.486 ms

? --- 192.168.10.4 ping statistics ---
? 4 packets transmitted, 4 received, 0% packet loss, time 3109ms
? rtt min/avg/max/mdev = 0.486/0.538/0.613/0.049 ms
nightwolf@Cristian-1711-NightWolf:~$
```

Figure 38: Ping de Laptop 2 a Raspberry Pi

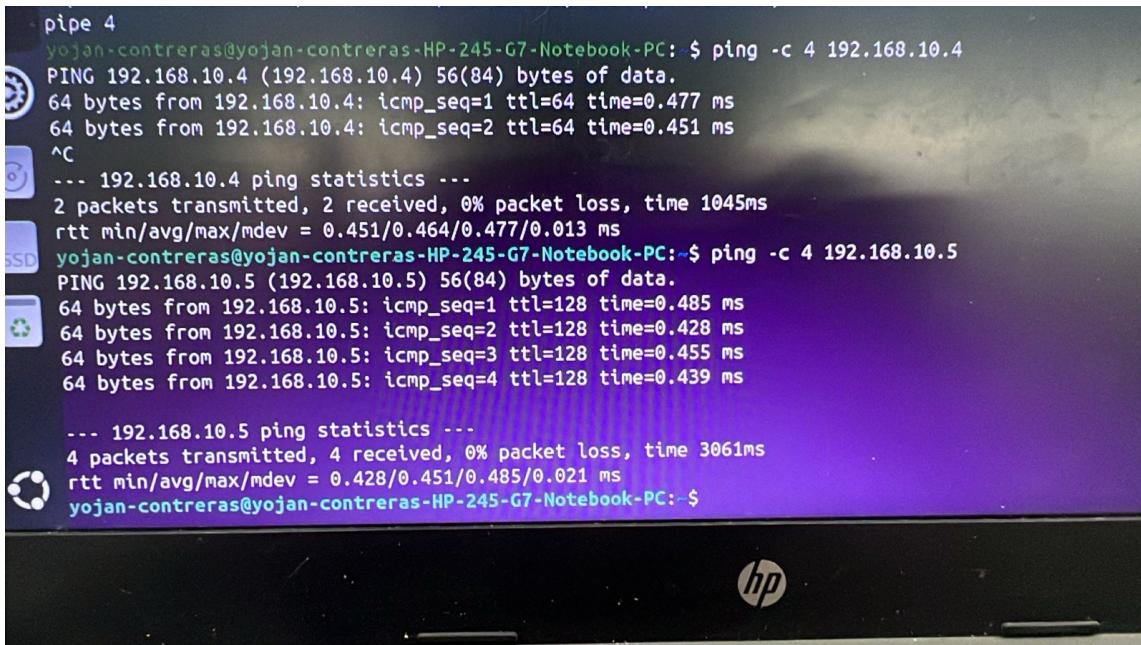


Figure 39: Ping de Laptop 1 a PC del Laboratorio

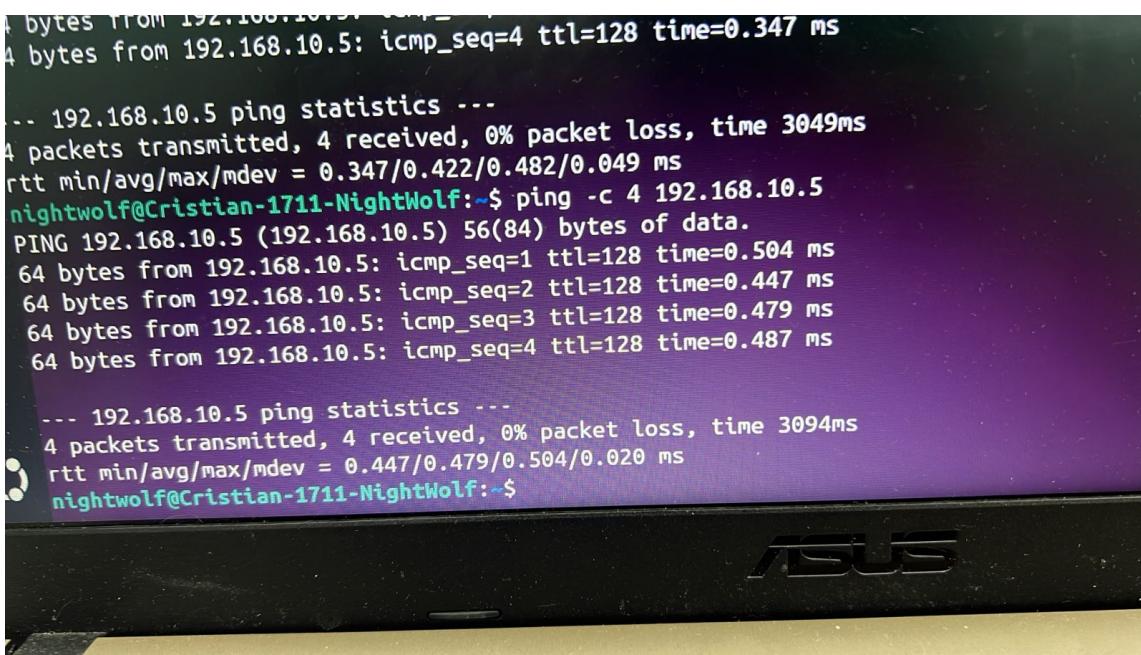


Figure 40: Ping de Laptop 2 a PC del Laboratorio

```
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 0ms, Máximo = 0ms, Media = 0ms

C:\Users\ETM3-01>

C:\Users\ETM3-01>netsh advfirewall set allprofiles state off
La operación solicitada requiere elevación (Ejecutar como administrador).

C:\Users\ETM3-01>ping 192.168.10.4

Haciendo ping a 192.168.10.4 con 32 bytes de datos:
Respuesta desde 192.168.10.4: bytes=32 tiempo<1m TTL=64

Estadísticas de ping para 192.168.10.4:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
                (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 0ms, Máximo = 0ms, Media = 0ms

C:\Users\ETM3-01>
```

Figure 41: Ping de PC del Laboratorio a Raspberry Pi

```
Estadísticas de ping para 192.168.10.10:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
                (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 0ms, Máximo = 0ms, Media = 0ms

C:\Users\ETM3-01>ping 192.168.10.10

Haciendo ping a 192.168.10.10 con 32 bytes de datos:
Respuesta desde 192.168.10.10: bytes=32 tiempo<1m TTL=64

Estadísticas de ping para 192.168.10.10:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
                (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 0ms, Máximo = 0ms, Media = 0ms

C:\Users\ETM3-01>
```

Figure 42: Ping de PC del Laboratorio a Laptop 1

The screenshot shows a Windows Command Prompt window with the following output:

```
tido avanzado  
es de streaming  
edia  
C:\Users\ETM3-01>ping 192.168.10.10  
Haciendo ping a 192.168.10.10 con 32 bytes de datos:  
Respueta desde 192.168.10.10: bytes=32 tiempo<1m TTL=64  
Estadísticas de ping para 192.168.10.10:  
Paquetes: enviados = 4, recibidos = 4, perdidos = 0  
(0% perdidos),  
Tiempos aproximados de ida y vuelta en milisegundos:  
Mínimo = 0ms, Máximo = 0ms, Media = 0ms  
C:\Users\ETM3-01>ping 192.168.10.11  
Haciendo ping a 192.168.10.11 con 32 bytes de datos:  
Respueta desde 192.168.10.11: bytes=32 tiempo<1m TTL=64  
Estadísticas de ping para 192.168.10.11:  
Paquetes: enviados = 4, recibidos = 4, perdidos = 0  
(0% perdidos),  
Tiempos aproximados de ida y vuelta en milisegundos:  
Mínimo = 0ms, Máximo = 0ms, Media = 0ms  
C:\Users\ETM3-01>
```

Figure 43: Ping de PC del Laboratorio a Laptop 2

Los resultados mostraron comunicación exitosa entre todos los equipos, evidenciando que la red fue configurada correctamente y que las interfaces se encuentran activas y dentro del mismo rango de direcciones.

3.7. Pruebas de conectividad y análisis de red

Se verificó la comunicación entre todos los equipos mediante `ping` y se escanearon los puertos de la red con `nmap`.

```
nmap 192.168.1.0/24
```

```

yojan-contreras@yojan-contreras-HP-245-G7-Notebook-PC: $ ping -c 4 192.168.10.5
PING 192.168.10.5 (192.168.10.5) 56(84) bytes of data.
64 bytes from 192.168.10.5: icmp_seq=1 ttl=128 time=0.466 ms
64 bytes from 192.168.10.5: icmp_seq=2 ttl=128 time=0.458 ms
64 bytes from 192.168.10.5: icmp_seq=3 ttl=128 time=0.443 ms
64 bytes from 192.168.10.5: icmp_seq=4 ttl=128 time=0.427 ms
...
--- 192.168.10.5 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3052ms
rtt min/avg/max/mdev = 0.427/0.448/0.466/0.014 ms

yojan-contreras@yojan-contreras-HP-245-G7-Notebook-PC: $ nmap 192.168.10.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-09-23 14:28 -05
Nmap scan report for 192.168.10.4
Host is up (0.00037s latency).
Not shown: 999 closed tcp ports (conn-refused)
PORT      STATE SERVICE
22/tcp    open  ssh

Nmap scan report for yojan-contreras-HP-245-G7-Notebook-PC (192.168.10.10)
Host is up (0.00028s latency).
Not shown: 996 closed tcp ports (conn-refused)
PORT      STATE SERVICE
25/tcp    open  smtp
3000/tcp  open  ppp
9090/tcp  open  zeus-admin
9100/tcp  open  jetdirect

Nmap done: 256 IP addresses (2 hosts up) scanned in 3.19 seconds
yojan-contreras@yojan-contreras-HP-245-G7-Notebook-PC: $

```

Figure 44: Nmap todos los dispositivos

```

yojan-contreras@yojan-contreras-HP-245-G7-Notebook-PC: ~
yojan-contreras@yojan-contreras-HP-245-G7-Notebook-PC: $ ip config
Object "config" is unknown, try "ip help".
yojan-contreras@yojan-contreras-HP-245-G7-Notebook-PC: ~
yojan-contreras@yojan-contreras-HP-245-G7-Notebook-PC: $ ifconfig
eno1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.10.10 netmask 255.255.255.0 broadcast 0.0.0.0
                ether f8:0d:ac:49:b9:b0 txqueuelen 1000 (Ethernet)
                RX packets 3593 bytes 298623 (298.6 KB)
                RX errors 0 dropped 0 overruns 0 frame 0
                TX packets 5740 bytes 358980 (358.9 KB)
                TX errors 0 dropped 128 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
        inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
loop txqueuelen 1000 (Bucle local)
        RX packets 30385 bytes 27116003 (27.1 MB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 30385 bytes 27116003 (27.1 MB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

virbr0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
        inet 192.168.122.1 netmask 255.255.255.0 broadcast 192.168.122.255
                ether 52:54:00:38:77:28 txqueuelen 1000 (Ethernet)
                RX packets 0 bytes 0 (0.0 B)
                RX errors 0 dropped 0 overruns 0 frame 0
                TX packets 0 bytes 0 (0.0 B)
                TX errors 0 dropped 47 overruns 0 carrier 0 collisions 0

yojan-contreras@yojan-contreras-HP-245-G7-Notebook-PC: ~

```

Figure 45: Ifconfig Laptop1

```
lightwolfgcristian-1711-NightWolf: ~ ifconfig
enp2s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.1.12 netmask 255.255.255.0 broadcast 192.168.1.255
        inet6 fe80::eda0:80ff:fe98:98ff prefixlen 64 scopeid 0x20<link>
          ether 88:07:f6:72:12:f3 txqueuelen 1000 (Ethernet)
            RX packets 2899 bytes 262105 (262.1 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 635 bytes 74081 (74.0 KB)
            TX errors 0 dropped 28 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
        inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
          loop txqueuelen 1000 (Bucle local)
            RX packets 1336 bytes 151808 (151.8 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 1336 bytes 151808 (151.8 KB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

virbr0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
        inet 192.168.122.1 netmask 255.255.255.0 broadcast 192.168.122.255
          ether 52:54:00:ee:8e:a3 txqueuelen 1000 (Ethernet)
            RX packets 0 bytes 0 (0.0 B)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 0 bytes 0 (0.0 B)
            TX errors 0 dropped 48 overruns 0 carrier 0 collisions 0

wlp1s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 10.200.8.158 netmask 255.255.240.0 broadcast 10.200.15.255
        inet6 fe80::8adb:24f7:d45d:4896 prefixlen 64 scopeid 0x20<link>
          ether f0:03:8c:dc:47:a1 txqueuelen 1000 (Ethernet)
```

Figure 46: Ifconfig Laptop2

Haciendo ping a 192.168.10.11 con 32 bytes de datos:
Respuesta desde 192.168.10.11: bytes=32 tiempo<1m TTL=64
Respuesta desde 192.168.10.11: bytes=32 tiempo<1m TTL=64
Respuesta desde 192.168.10.11: bytes=32 tiempo<1m TTL=64
Respuesta desde 192.168.10.11: bytes=32 tiempo<1m TTL=64

Estadísticas de ping para 192.168.10.11:
Paquetes: enviados = 4, recibidos = 4, perdidos = 0
(0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
Mínimo = 0ms, Máximo = 0ms, Media = 0ms

C:\Users\ETM3-01>nmap 192.168.10.5
"nmap" no se reconoce como un comando interno o externo,
programa o archivo por lotes ejecutable.

C:\Users\ETM3-01>ipconfig

Configuración IP de Windows

Adaptador de Ethernet Ethernet 2:

Sufijo DNS específico para la conexión. . . :
Vínculo: dirección IPv6 local. . . : fe80::8f33:1b3f:b1b0:1937%15
Dirección IPv4. : 192.168.10.5
Máscara de subred : 255.255.255.0
Puerta de enlace predeterminada :

Figure 47: Ipconfig PC-ethm

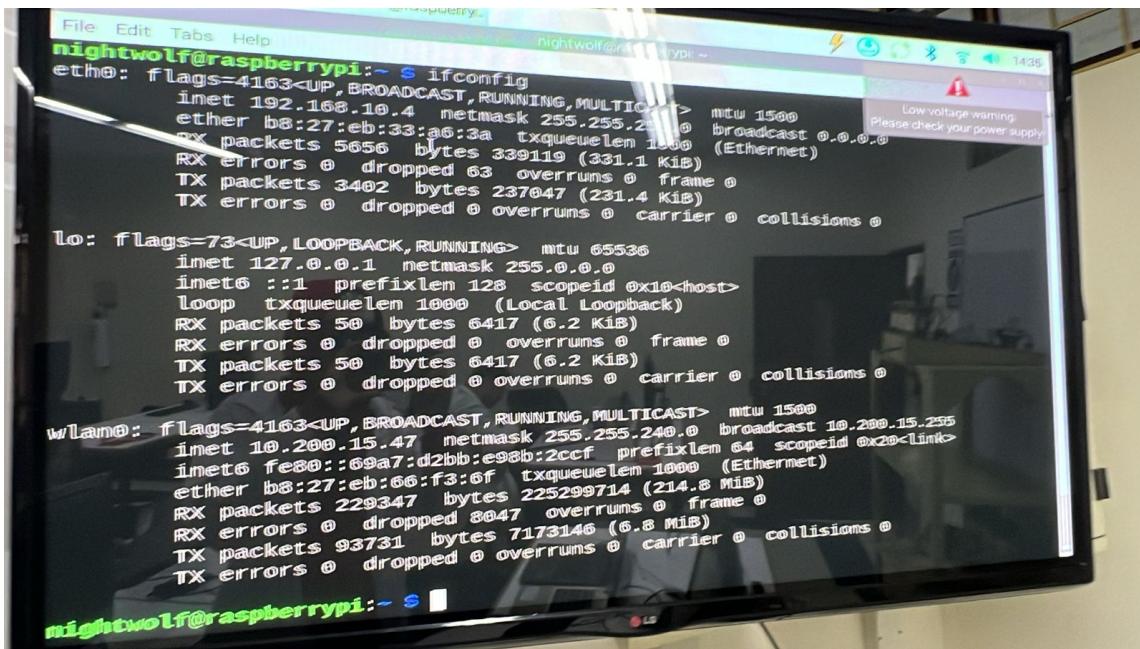


Figure 48: Ipconfig Raspberri

3.8. Transferencia de archivos mediante SSH y SCP

Se configuró un servidor SSH en los equipos Linux para el envío de archivos.

```

sudo apt update
sudo apt install openssh-server
sudo systemctl start ssh
sudo systemctl status ssh

```

Se envió el archivo al otro equipo con:

```
scp d24.txt usuario@192.168.X.X:/home/usuario/
```

Al ingresar la contraseña, el archivo se transfirió correctamente. Finalmente, se verificó en el equipo receptor que el archivo se encontraba en la ruta destino.

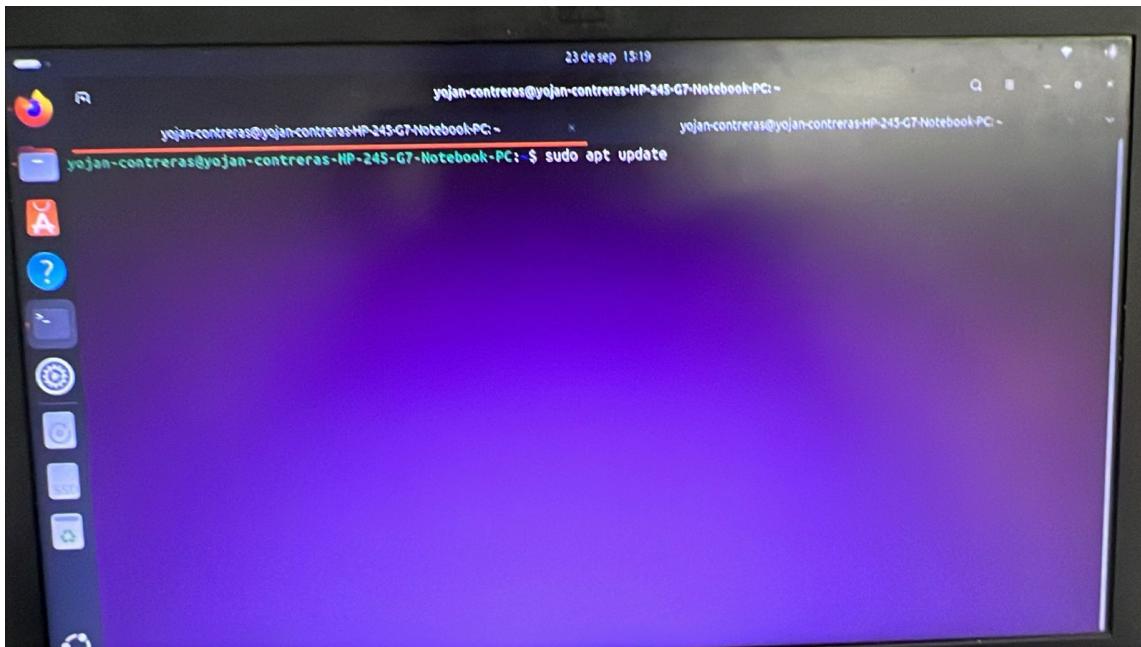


Figure 49: Actualizamos paquetes

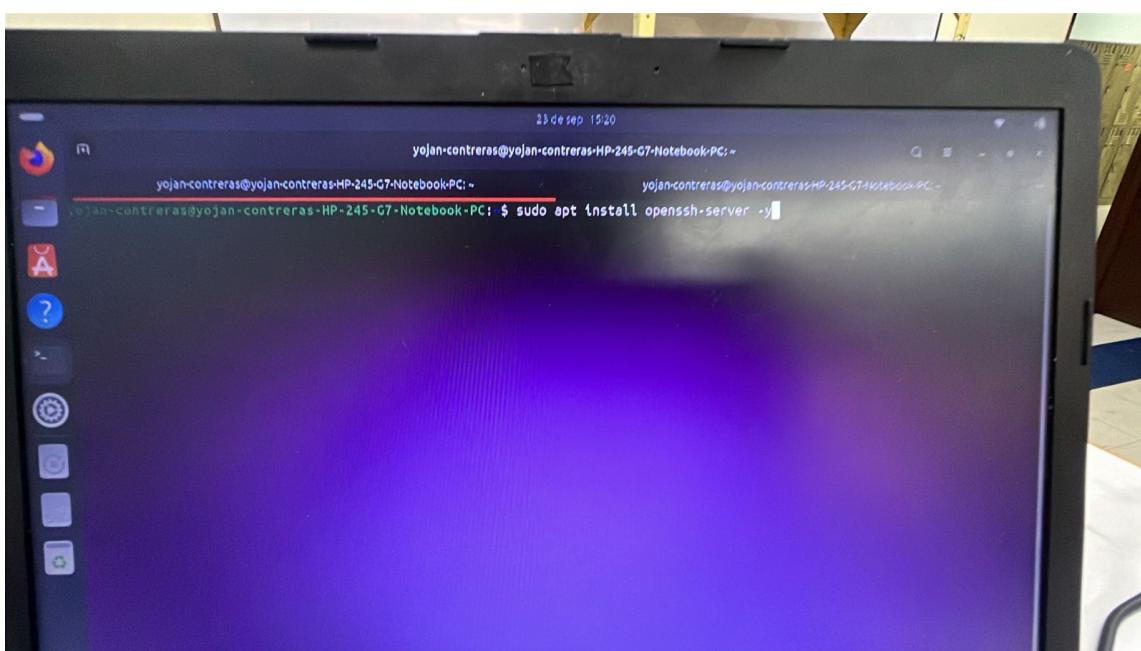


Figure 50: Instalamos servidor ssh

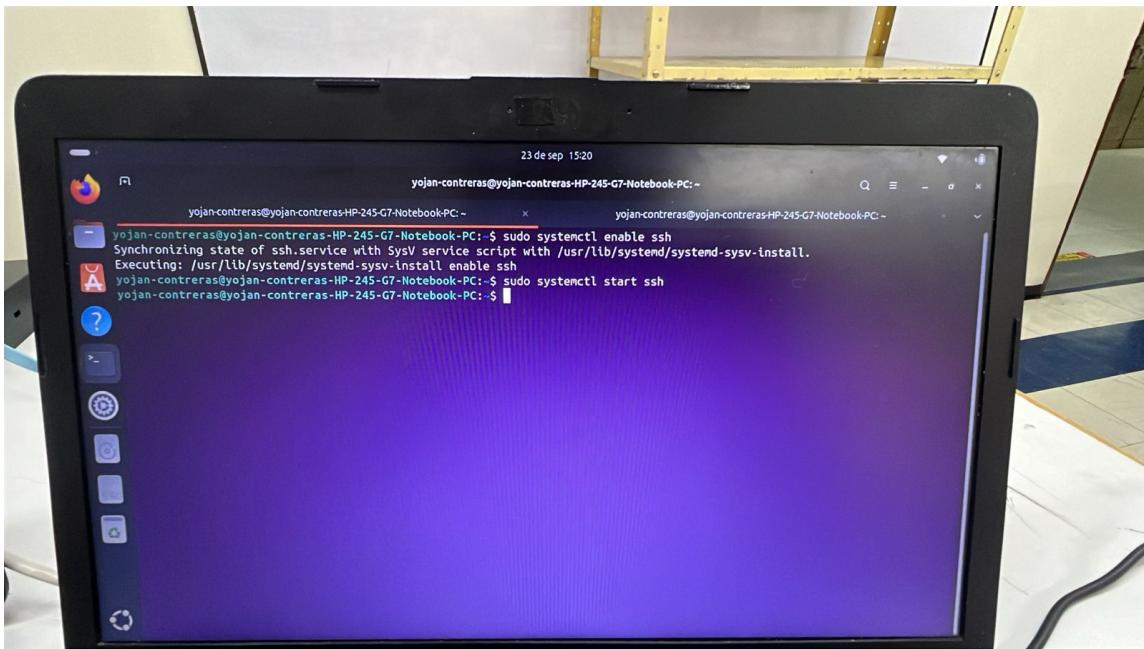


Figure 51: Activamos e iniciamos el servidor

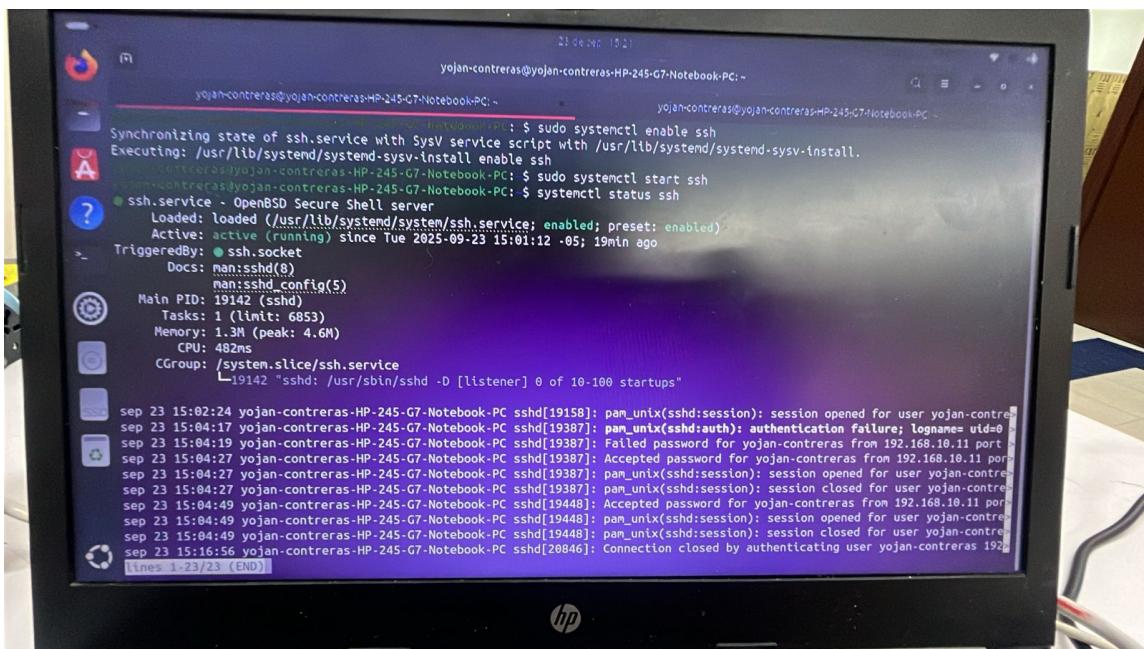


Figure 52: Verificamos que este activo el servidor en Laptop1



Figure 53: Verificamos que este activo el servidor en Laptop2

El mismo procedimiento se realizó en el equipo receptor para habilitar el servicio SSH. Luego, se creó un archivo de prueba:

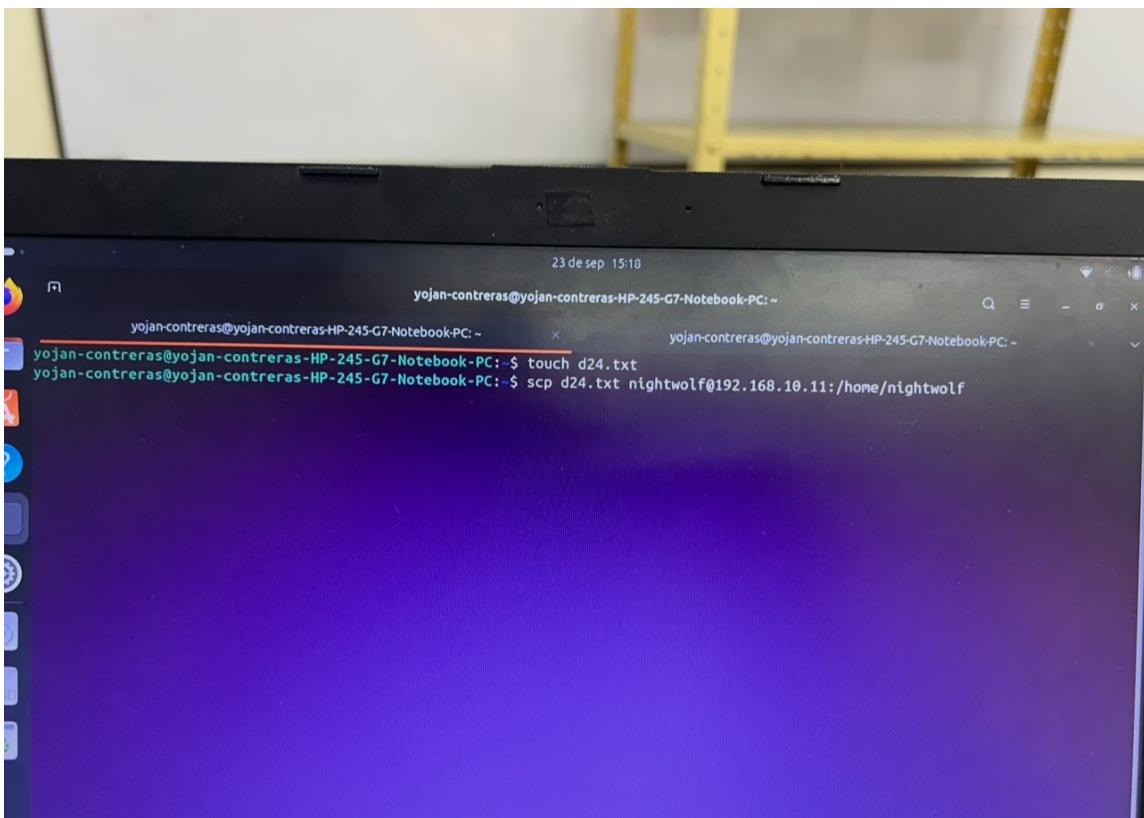


Figure 54: Enviamos el archivo (d24.txt)

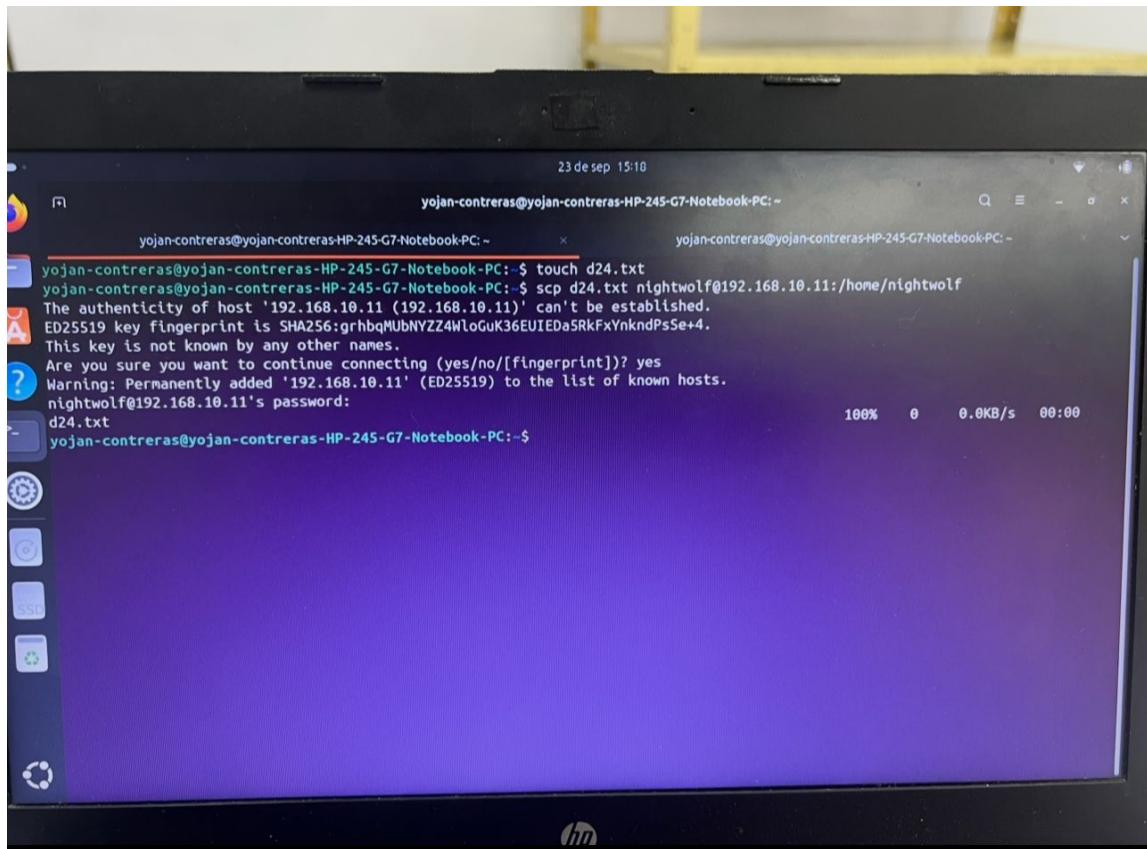


Figure 55: Ponemos la contraseña para poder recibirla



```

inet6 ::1/128 scope host noprefixroute
  valid_lft forever preferred_lft forever
:enp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
  link/ether 88:d7:f6:72:12:f3 brd ff:ff:ff:ff:ff:ff
  inet 192.168.1.12/24 brd 192.168.1.255 scope global noprefixroute enp2s0
    valid_lft forever preferred_lft forever
  inet 192.168.10.11/24 scope global enp2s0
    valid_lft forever preferred_lft forever
  inet6 fe80::eda0:d80::d9af:988f/64 scope link noprefixroute
    valid_lft forever preferred_lft forever
:wlpis0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
  link/ether f0:03:8c:dc:47:a1 brd ff:ff:ff:ff:ff:ff
  inet 10.200.8.158/20 brd 10.200.15.255 scope global dynamic noprefixroute wlpis0
    valid_lft 1795sec preferred_lft 1795sec
  inet6 fe80::8adb:24f7:d45d:4896/64 scope link noprefixroute
    valid_lft forever preferred_lft forever
4: virbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default qlen 1000
  link/ether 52:54:00:ee:8e:a3 brd ff:ff:ff:ff:ff:ff
  inet 192.168.122.1/24 brd 192.168.122.255 scope global virbr0
    valid_lft forever preferred_lft forever
nightwolf@Cristian-1711-NightWolf:~$ scp p23.txt yojan-contreras@192.168.10.10:/home/yojan-contreras/
yojan-contreras@192.168.10.10's password:
nightwolf@Cristian-1711-NightWolf:~$ sudo ufw allow ssh
[sudo] password para nightwolf:
Regla añadida
Regla añadida (v6)
nightwolf@Cristian-1711-NightWolf:~$ ls
d24.txt  ejemplo_dashboard  jcarrito  logs      p23.txt  prueba  'Tarea 4.pdf'  venv
Descargas Escritorio       juegol   nidesktop  Plantillas  Público  'Tarea 4.pdf.txt'  Vídeos
Documentos Imágenes        juego_carro Música  Práctica_Linux  snap   tts-env
nightwolf@Cristian-1711-NightWolf:~$
```

Figure 56: Verificamos

4. Resultados

- Comunicación establecida entre todos los dispositivos.
- Escaneo de red nmap muestra todos los hosts activos.
- Transferencia exitosa de archivos mediante scp.

5. Punto 2: Probando las Máquinas Virtuales

5.1. Objetivo

Instalar diferentes máquinas virtuales y documentar el proceso de instalación en QEMU/KVM, explicando paso a paso cómo crear, configurar y ejecutar cada sistema operativo (Ubuntu, CentOS, Alpine y Scientific Linux) dentro de un entorno de virtualización.

5.2. Preparación Común

Antes de instalar cualquier sistema operativo, se deben preparar las herramientas y directorios necesarios en el sistema host (Linux).

1. Actualizar los repositorios e instalar los paquetes requeridos:

```
sudo apt update  
sudo apt install qemu-kvm libvirt-daemon-system libvirt-clients virtinst bridge-...
```

2. Verificar el estado de KVM:

```
sudo systemctl status libvирtd  
ls /dev/kvm
```

3. Crear directorios de trabajo para las imágenes e ISOs:

```
mkdir -p ~/qemu/isos ~/qemu/images  
cd ~/qemu
```

4. Crear una imagen base para la máquina virtual (por ejemplo 25 GB):

```
qemu-img create -f qcow2 images/ubuntu.qcow2 25G
```

5. Configurar red con NAT y reenvío de puertos (para poder acceder por SSH desde el host):

```
-netdev user,id=net0,hostfwd=tcp::2222-:22
```

(Imagen: interfaz de Virt-Manager o consola QEMU mostrando el inicio de la instalación)

5.3. Instalación de Ubuntu Server

1. Descargar la ISO de Ubuntu Server y guardarla en `/qemu/isos/ubuntu-server.iso`.
2. Crear un disco para la VM:

```
qemu-img create -f qcow2 ~/qemu/images/ubuntu-server.qcow2 25G
```

3. Iniciar la instalación:

```
qemu-system-x86_64 \  
    -enable-kvm -m 2048 -smp 2 \  
    -drive file=~/qemu/images/ubuntu-server.qcow2,format=qcow2,if=virtio \  
    -cdrom ~/qemu/isos/ubuntu-server.iso \  
    -boot d \  
    -netdev user,id=net0,hostfwd=tcp::2222-:22 \  
    -device virtio-net-pci,netdev=net0
```

4. Seguir los pasos del instalador: seleccionar idioma, crear usuario, configurar red (DHCP o IP estática) y habilitar el servicio SSH.
5. Tras finalizar la instalación, reiniciar la máquina:

```
qemu-system-x86_64 -enable-kvm -m 2048 -smp 2 \  
    -drive file=~/qemu/images/ubuntu-server.qcow2,format=qcow2,if=virtio \  
    -netdev user,id=net0,hostfwd=tcp::2222-:22 \  
    -device virtio-net-pci,netdev=net0
```

6. Acceder por SSH desde el host:

```
ssh -p 2222 usuario@localhost
```

7. Post-instalación recomendada:

```
sudo apt update && sudo apt upgrade  
sudo apt install qemu-guest-agent cloud-init net-tools  
sudo systemctl enable --now qemu-guest-agent
```

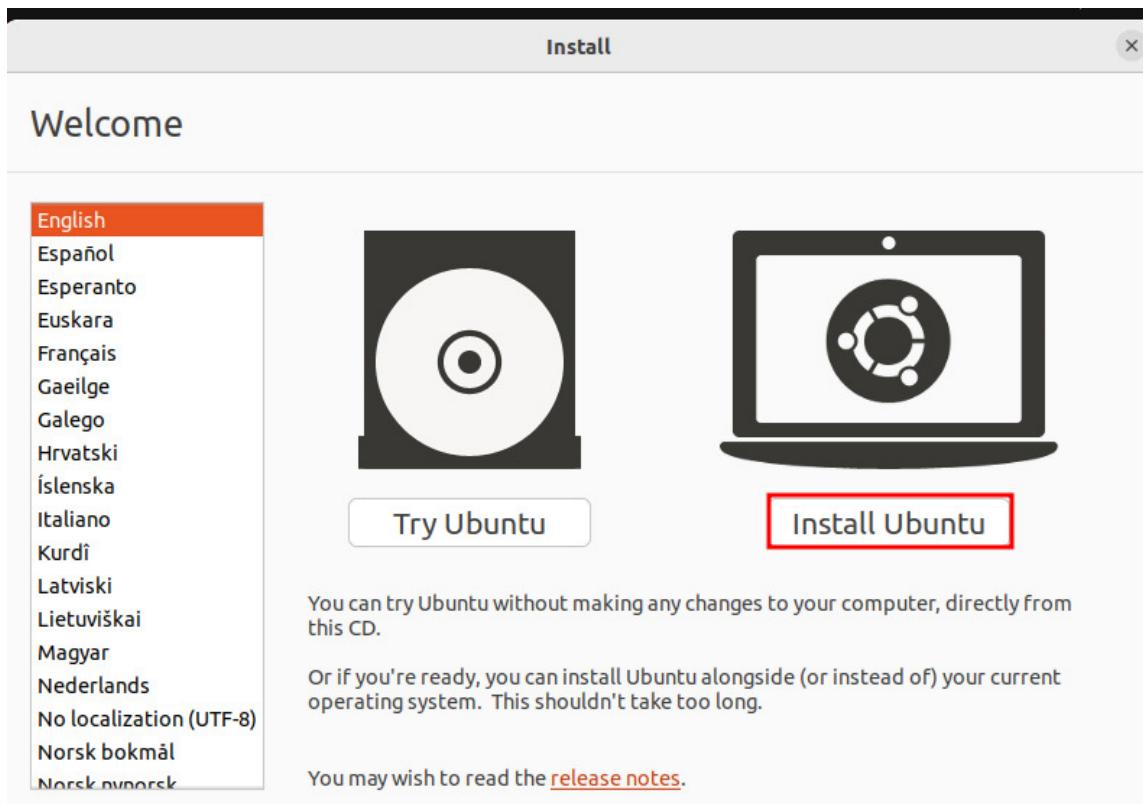


Figure 57: Instalador Ubuntu

5.4. Instalación de CentOS / CentOS Stream

1. Descargar la ISO de CentOS Stream y guardarla en `/qemu/isos/centos-stream.iso`.
2. Crear el disco para la VM:

```
qemu-img create -f qcow2 ~/qemu/images/centos-stream.qcow2 25G
```

3. Iniciar el instalador (modo gráfico con salida SDL):

```
qemu-system-x86_64 -enable-kvm -m 2048 -smp 2 \
    -drive file=~/qemu/images/centos-stream.qcow2,format=qcow2,if=virtio \
    -cdrom ~/qemu/isos/centos-stream.iso -boot d \
    -netdev user,id=net0,hostfwd=tcp::2223-:22 \
    -device virtio-net-pci,netdev=net0 \
    -vga qxl -display sdl
```

4. Seguir los pasos del instalador Anaconda:

- Seleccionar idioma y zona horaria.
 - Configurar red y tipo de instalación (mínima o servidor con GUI).
 - Crear usuario y contraseña de root.
5. Reiniciar la VM y arrancar desde el disco:

```
qemu-system-x86_64 -enable-kvm -m 2048 -smp 2 \
-drive file=~/qemu/images/centos-stream.qcow2,format=qcow2,if=virtio \
-netdev user,id=net0,hostfwd=tcp::2223-:22 \
-device virtio-net-pci,netdev=net0
```

6. Post-instalación:

```
sudo dnf update
sudo dnf install qemu-guest-agent openssh-server
sudo systemctl enable --now qemu-guest-agent
sudo systemctl enable --now sshd
```



Figure 58: Instalador CentOS

5.5. Instalación de Alpine Linux

1. Descargar la ISO de Alpine y guardarla en `/qemu/isos/alpine.iso`.
2. Crear un disco pequeño (4 GB):

```
qemu-img create -f qcow2 ~/qemu/images/alpine.qcow2 4G
```

3. Iniciar la instalación:

```
qemu-system-x86_64 -enable-kvm -m 512 -smp 1 \
    -drive file=~/qemu/images/alpine.qcow2,format=qcow2,if=virtio \
    -cdrom ~/qemu/isos/alpine.iso -boot d \
    -netdev user,id=net0,hostfwd=tcp::2224-:22 \
    -device virtio-net-pci,netdev=net0 \
    -nographic
```

4. En la consola, iniciar sesión como `root` y ejecutar:

```
setup-alpine
```

Seguir el asistente para configurar teclado, red, zona horaria, contraseña, tipo de disco (sys) e instalación en disco.

5. Reiniciar y arrancar desde el disco:

```
qemu-system-x86_64 -enable-kvm -m 512 -smp 1 \
    -drive file=~/qemu/images/alpine.qcow2,format=qcow2,if=virtio \
    -netdev user,id=net0,hostfwd=tcp::2224-:22 \
    -device virtio-net-pci,netdev=net0 \
    -nographic
```

6. Post-instalación:

```
apk update
apk add openssh
rc-update add sshd
rc-service sshd start
```

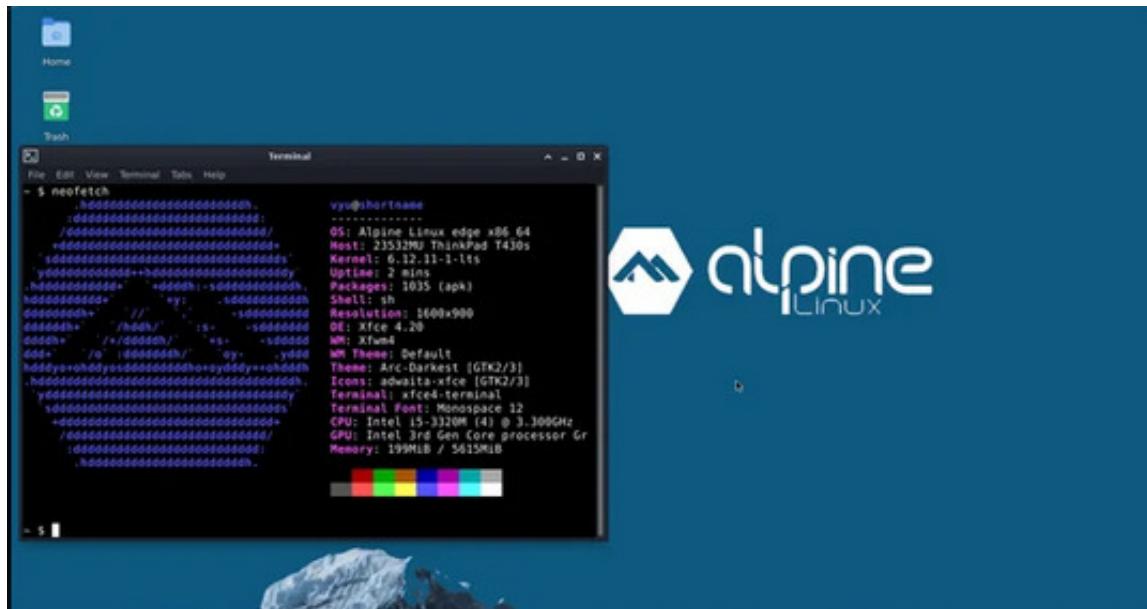


Figure 59: Instalador de Alpine

5.6. Instalación de Scientific Linux

1. Descargar la ISO de Scientific Linux y guardarla en `/qemu/isos/scientific.iso`.
2. Crear un disco:

```
qemu-img create -f qcow2 ~/qemu/images/scientific.qcow2 25G
```

3. Iniciar el instalador:

```
qemu-system-x86_64 -enable-kvm -m 2048 -smp 2 \
    -drive file=~/qemu/images/scientific.qcow2,format=qcow2,if=virtio \
    -cdrom ~/qemu/isos/scientific.iso -boot d \
    -netdev user,id=net0,hostfwd=tcp::2225-:22 \
    -device virtio-net-pci,netdev=net0 \
    -vga qxl -display sdl
```

4. Seguir los pasos del instalador Anaconda (idéntico a CentOS):

- Seleccionar idioma, zona horaria y red.
- Elegir tipo de instalación (mínima).
- Crear usuario y contraseña.

5. Reiniciar la máquina y arrancar desde disco:

```
qemu-system-x86_64 -enable-kvm -m 2048 -smp 2 \
    -drive file=~/qemu/images/scientific.qcow2,format=qcow2,if=virtio \
    -netdev user,id=net0,hostfwd=tcp::2225-:22 \
    -device virtio-net-pci,netdev=net0
```

6. Post-instalación:

```
sudo yum update
sudo yum install qemu-guest-agent openssh-server
sudo systemctl enable --now qemu-guest-agent
sudo systemctl enable --now sshd
```

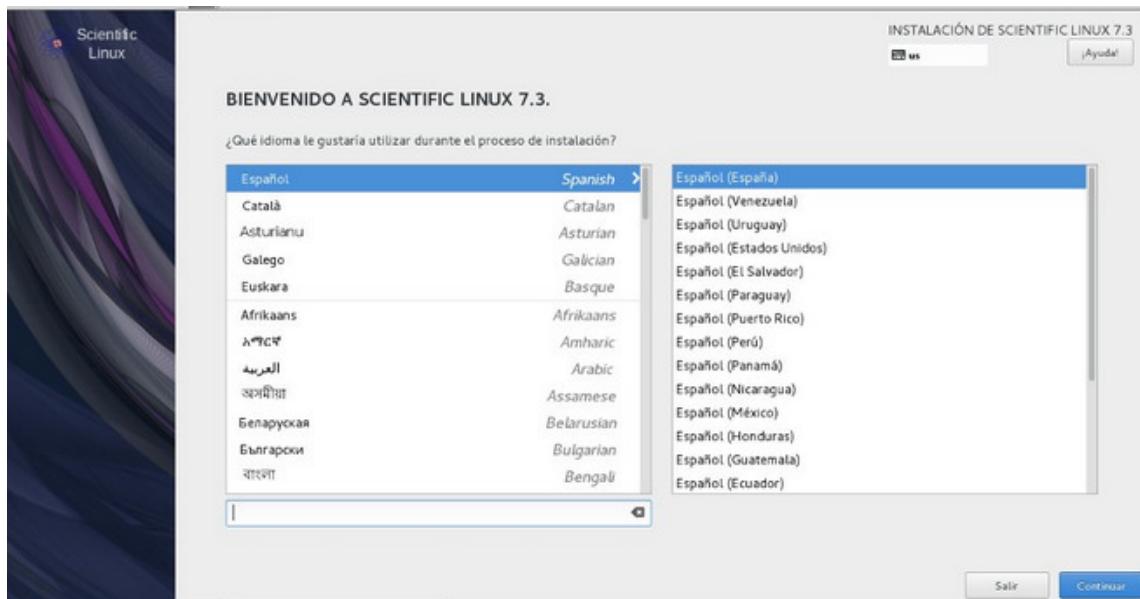


Figure 60: Instalador scientific Linux

5.7. Recomendaciones Finales

- Utilizar el formato de disco qcow2 para aprovechar snapshots y ahorro de espacio.
- Habilitar `qemu-guest-agent` en todas las máquinas virtuales para mejorar la comunicación con el host.
- Usar `virtio` como tipo de interfaz para discos y red, mejorando el rendimiento.

- Mantener una configuración de red NAT con reenvío de puertos, o configurar un puente (**bridge**) si se requiere acceso LAN real.
- Realizar instantáneas (snapshots) antes de grandes cambios:

```
qemu-img snapshot -c antes-de-prueba images/ubuntu-server.qcow2
```

- Para instalaciones automatizadas, usar **cloud-init** o **virt-install**:

```
virt-install --name ubuntu-server \
--vcpus 2 --memory 2048 \
--disk path=~/qemu/images/ubuntu-server.qcow2,size=25,format=qcow2 \
--cdrom ~/qemu/isos/ubuntu-server.iso \
--os-variant ubuntu20.04 \
--network network=default \
--graphics none
```

6. Conclusión del Punto 2

Se documentó el proceso completo de instalación de diversas distribuciones Linux en máquinas virtuales usando QEMU/KVM. Cada instalación permitió comprender las diferencias entre instaladores (Anaconda, setup-alpine, instalador de Ubuntu) y los parámetros esenciales de configuración: disco, red, CPU y memoria. El uso de **virtio**, **qemu-img** y **qemu-guest-agent** demostró la versatilidad de QEMU para entornos educativos y de pruebas.

7. Conclusiones

En la primera parte, se realizó la conexión y configuración de un **switch Cisco 2960** mediante interfaz serial usando herramientas como **minicom**, **screen** y **picocom**. Esta etapa permitió comprender el funcionamiento de los puertos físicos, las VLANs, la tabla de direcciones MAC y la importancia de los comandos de diagnóstico dentro del entorno de un switch gestionable. Posteriormente, se estableció la comunicación entre varios dispositivos (dos laptops, una Raspberry Pi y un PC de laboratorio), verificando la conectividad mediante **ping**, el análisis de red con **nmap** y la transferencia de archivos usando **scp** sobre **ssh**. Con ello, se demostró el proceso completo de diseño, configuración y prueba de una

red local funcional, comprendiendo conceptos clave como direccionamiento IP, máscara de subred, puerta de enlace y servicios activos en la red.

En la segunda parte del laboratorio, se documentó el proceso de instalación de diferentes distribuciones GNU/Linux (Ubuntu, CentOS, Alpine y Scientific Linux) dentro de máquinas virtuales empleando **QEMU/KVM**. Esto permitió entender los componentes fundamentales de la virtualización, como la asignación de recursos (CPU, memoria, disco y red), el manejo de imágenes en formato `qcow2`, el uso de `virtio` para mejorar el rendimiento y la creación de redes NAT para conectar las máquinas virtuales entre sí y con el host. Cada instalación evidenció las diferencias entre los instaladores y las herramientas de configuración de cada distribución, resaltando la flexibilidad de QEMU como plataforma para simulaciones y pruebas en entornos controlados.

En conjunto, el laboratorio integró de manera práctica conceptos de **redes, virtualización y administración de sistemas Linux**, fortaleciendo las competencias en el manejo de infraestructuras híbridas (físicas y virtuales).