



# Implementación de una Red de Sensores Ad-hoc utilizando el Protocolo B.A.T.M.A.N.

Bryan Esteven Ariza Palma, Juan Sebastián Herrán Páez

December 23, 2024

# Contents

<b>1</b>	<b>Resumen</b>	<b>2</b>
<b>2</b>	<b>Introducción</b>	<b>3</b>
2.1	Contexto . . . . .	3
2.2	Objetivos . . . . .	4
2.2.1	Objetivo General . . . . .	4
2.2.2	Objetivos Específicos . . . . .	4
<b>3</b>	<b>Marco Teórico</b>	<b>5</b>
3.1	Estandares de la red . . . . .	5
3.2	Multiplexación . . . . .	6
<b>4</b>	<b>Diseño de la Red</b>	<b>7</b>
4.1	Direccionamiento y Segmentación . . . . .	7
<b>5</b>	<b>Implementación</b>	<b>9</b>
5.1	Configuración de los Nodos . . . . .	9
5.1.1	Instalación del Protocolo B.A.T.M.A.N. . . . .	10
5.1.2	Configuración de la Red Ad-hoc . . . . .	10
5.2	Conexión y Pruebas de Conectividad . . . . .	10
5.2.1	Pruebas de Ping . . . . .	10
5.2.2	Pruebas de Rutas (Traceroute) . . . . .	10
<b>6</b>	<b>Conclusiones</b>	<b>14</b>
6.1	Conclusiones Generales . . . . .	14
<b>A</b>	<b>Anexos</b>	<b>15</b>
A.1	Código y Configuraciones . . . . .	15
A.2	Imágenes Adicionales . . . . .	15

# Chapter 1

## Resumen

Este proyecto explora la implementación de una red móvil ad hoc para la transmisión de datos de sensores, chat en un entorno universitario, utilizando el protocolo de enrutamiento BATMAN. Se utilizan tres nodos: tres laptop Linux. La red ad hoc permite que cada nodo capture datos de sensores y los transmita de forma dinámica sin la necesidad de un punto de acceso fijo.

El protocolo BATMAN garantiza la conectividad continua entre los nodos, permitiendo que la red se adapte a cambios de topología o desconexiones temporales. PC1, que actúa como el nodo principal, recibe los datos de los sensores y los envía a una plataforma en la nube para su almacenamiento y análisis. Este proyecto demuestra cómo las redes ad hoc pueden facilitar la recopilación de datos en entornos sin infraestructura, permitiendo una comunicación efectiva y adaptable entre dispositivos móviles y sistemas de monitoreo centralizados.

# Chapter 2

## Introducción

### 2.1 Contexto

Las redes móviles ad hoc son redes formadas por dispositivos móviles que pueden comunicarse entre sí sin necesidad de infraestructura centralizada, como routers o puntos de acceso. Cada dispositivo actúa como un nodo en la red, que puede enviar, recibir y enrutar datos.

En aplicaciones de sensores y dispositivos móviles como es en el caso de este proyecto, las redes ad hoc permiten que cada nodo envíe datos de sensores a otros nodos o incluso a una nube sin depender de un punto de acceso fijo. Los nodos pueden entrar y salir de la red de manera dinámica, sin afectar el funcionamiento general de la red.

El protocolo BATMAN (Better Approach To Mobile Ad-hoc Networking) es un protocolo de enrutamiento diseñado para redes ad hoc. Su función principal es encontrar y mantener rutas óptimas entre nodos de manera descentralizada. BATMAN permite que cada nodo solo conozca su vecino más cercano. A medida que un nodo se mueve o se desconecta, el protocolo actualiza automáticamente las rutas sin la intervención del usuario, lo que es ideal para redes donde los dispositivos pueden moverse o cambiar de ubicación con frecuencia.

En este proyecto, BATMAN permitirá que los nodos se comuniquen entre sí de manera dinámica, manteniendo la conectividad aunque algunos nodos se desconecten temporalmente o se muevan.

## **2.2 Objetivos**

### **2.2.1 Objetivo General**

Implementar una red ad hoc utilizando el protocolo de enrutamiento BATMAN para interconectar nodos de computadoras. Esta red permitirá la transmisión y recepción de datos de sensores en tiempo real, sin la necesidad de una infraestructura de red fija.

### **2.2.2 Objetivos Específicos**

- Configurar un nodo principal con Linux que actuará como punto de monitoreo de datos y de acceso a la nube para almacenar la información de los sensores.
- Configurar tres nodos en laptops Linux para capturar datos de sensores y enviarlos al nodo principal a través de la red ad hoc.
- Establecer una red ad hoc utilizando el protocolo BATMAN para conectar todos los nodos de manera descentralizada, sin depender de un punto de acceso fijo.
- Realizar pruebas de conectividad entre los nodos para asegurar la transmisión de datos en la red, evaluando la estabilidad y la latencia de la comunicación.

# Chapter 3

## Marco Teórico

### 3.1 Estándares de la red

En redes ad hoc y redes de sensores, el estándar IEEE 802.11 es fundamental, ya que describe las especificaciones para redes inalámbricas. Algunos de los sub-estándares relevantes incluyen:

- **IEEE 802.11a:** Opera en la banda de 5 GHz y proporciona velocidades de hasta 54 Mbps, adecuado para entornos de alta densidad de dispositivos.
- **IEEE 802.11b:** Opera en la banda de 2.4 GHz, con una velocidad máxima de 11 Mbps; es ampliamente compatible con muchos dispositivos.
- **IEEE 802.11g:** También en la banda de 2.4 GHz, proporciona hasta 54 Mbps, siendo compatible con la mayoría de los dispositivos Wi-Fi.
- **IEEE 802.11n:** Opera en las bandas de 2.4 y 5 GHz, proporcionando velocidades de hasta 600 Mbps y mayor rango de cobertura.
- **IEEE 802.11ac:** En la banda de 5 GHz, ofrece velocidades superiores a 1 Gbps, adecuado para aplicaciones de alta velocidad y baja latencia.

Para este proyecto, la red ad hoc se configurará utilizando la banda de 2.4 GHz, con el estándar IEEE 802.11b/g/n, dado que estos estándares ofrecen compatibilidad con la mayoría de los dispositivos móviles y laptops.

## 3.2 Multiplexación

La multiplexación es una técnica de transmisión de datos que permite enviar múltiples señales o flujos de datos a través de un solo canal de comunicación. En redes de sensores, la multiplexación es útil para optimizar el uso del ancho de banda, permitiendo que diferentes sensores envíen datos a través de la misma red de manera eficiente.

Existen varios tipos de multiplexación:

- **Multiplexación por División de Tiempo (TDM):** Se asigna un intervalo de tiempo a cada sensor, permitiendo que envíe sus datos en secuencia.
- **Multiplexación por División de Frecuencia (FDM):** Cada sensor utiliza una frecuencia específica para transmitir sus datos.

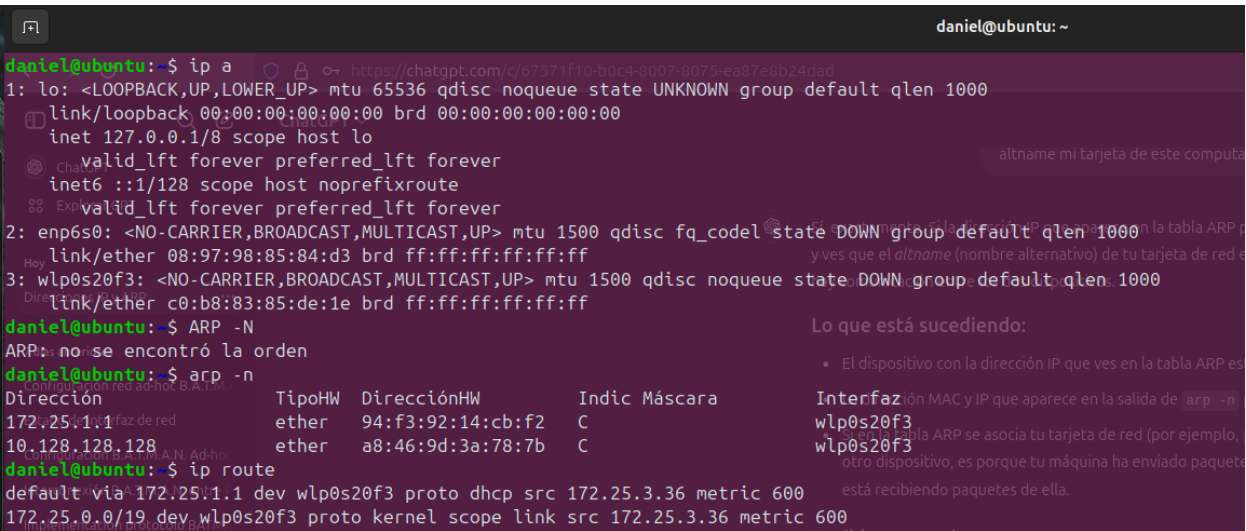
En el caso de una red ad hoc con el protocolo BATMAN, los datos de múltiples sensores pueden ser multiplexados a través de TDM o FDM para optimizar la transmisión de información en un entorno donde los recursos de red son limitados.

# Chapter 4

## Diseño de la Red

### 4.1 Direcccionamiento y Segmentación

Esquema de direccionamiento IP inicial para cada PC.



```
daniel@ubuntu:~$ ip a
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: enp6s0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc fq_codel state DOWN group default qlen 1000
    link/ether 08:97:98:85:84:d3 brd ff:ff:ff:ff:ff:ff
3: wlp0s20f3: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default qlen 1000
    link/ether c0:b8:83:85:de:1e brd ff:ff:ff:ff:ff:ff

daniel@ubuntu:~$ ARP -N
ARP: no se encontró la orden

daniel@ubuntu:~$ arp -n
Dirección      TipoHW  DirecciónHW      Indic  Máscara  Interfaz MAC y IP que aparece en la salida de 'arp -n'
172.25.1.1      ether   94:f3:92:14:cb:f2  C      C         wlp0s20f3
10.128.128.128  ether   a8:46:9d:3a:78:7b  C      C         wlp0s20f3

daniel@ubuntu:~$ ip route
default via 172.25.1.1 dev wlp0s20f3 proto dhcp src 172.25.3.36 metric 600
172.25.0.0/19 dev wlp0s20f3 proto kernel scope link src 172.25.3.36 metric 600
```

Figure 4.1: Estado inicial de las interfaces PC1.

Comando nmcli device status para saber en que estado se encuentran las interfaces y analizar cual es de utilidad en cada PC.



```

ubuntu@ubuntu:~$ ip a
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: wlo1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether 94:e2:3c:a9:3c:0b brd ff:ff:ff:ff:ff:ff
    altname wlp0s20f3
    inet 172.25.3.139/19 brd 172.25.31.255 scope global dynamic noprefixroute wlo1
        valid_lft 3584sec preferred_lft 3584sec
    inet6 fe80::5580:d90c:d20d:4312/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: lxcbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default qlen 1000
    link/ether 00:16:3e:00:00:00 brd ff:ff:ff:ff:ff:ff
    inet 10.0.3.1/24 brd 10.0.3.255 scope global lxcbr0
        valid_lft forever preferred_lft forever
ubuntu@ubuntu:~$ ^C
ubuntu@ubuntu:~$ arp -n
Dirección          TipoHW  DirecciónHW      Indic Máscara      Interfaz
172.25.1.1          ether   94:f3:92:14:cb:f2 C                   wlo1

```

Figure 4.2: Estado inicial de las interfaces PC2.

```

daniel@ubuntu:~$ nmcli device status
DEVICE          TYPE      STATE          CONNECTION
wlp0s20f3        wifi      conectado      USergio_Academia
lo               loopback  connected (externally) lo
p2p-dev-wlp0s20f3 wifi-p2p   desconectado   --
enp6s0           ethernet  no disponible  --

```

Figure 4.3: nmcli device status PC1

```

ubuntu@ubuntu:~$ nmcli device
DEVICE          TYPE      STATE          CONNECTION
wlo1            wifi      conectado      USergio_Academia
lo              loopback  connected (externally) lo
lxcbr0          bridge    connected (externally) lxcbr0
p2p-dev-wlo1    wifi-p2p   desconectado   --
ubuntu@ubuntu:~$

```

Figure 4.4: nmcli device status PC2

# Chapter 5

## Implementación

### 5.1 Configuración de los Nodos

```
daniel@ubuntu:~$ sudo batctl if add wlp0s20f3
daniel@ubuntu:~$ batctl if
wlp0s20f3: inactive
daniel@ubuntu:~$ nmcli dev connect wlp0s20f3
Error: no se pudo añadir o activar una conexión nueva: A 'wireless' setting is required if no AP path was given.
daniel@ubuntu:~$ batctl if
wlp0s20f3: inactive
daniel@ubuntu:~$ nmcli dev connect wlp0s20f3
Error: no se pudo añadir o activar una conexión nueva: A 'wireless' setting is required if no AP path was given.
daniel@ubuntu:~$ sudo systemctl start NetworkManager
daniel@ubuntu:~$ nmcli connection delete id "USergio_Academia"
La conexión «USergio_Academia» (07111359-125f-41da-8109-96f646dc4a37) se ha borrado correctamente.
daniel@ubuntu:~$ sudo nmcli dev set wlp0s20f3 managed no
daniel@ubuntu:~$ sudo ifconfig wlp0s20f3 down
daniel@ubuntu:~$ sudo iwconfig wlp0s20f3 mode ad-hoc
daniel@ubuntu:~$ sudo iwconfig wlp0s20f3 essid TLON-ADHOC
daniel@ubuntu:~$ sudo iwconfig wlp0s20f3 ap 02:1B:55:AD:0C:02
daniel@ubuntu:~$ sudo iwconfig wlp0s20f3 channel 1
daniel@ubuntu:~$ sudo ifconfig wlp0s20f3 mtu 1532 up
daniel@ubuntu:~$ sudo batctl if add wlp0s20f3
daniel@ubuntu:~$ sudo ip link set up dev bat0
daniel@ubuntu:~$ sudo ifconfig bat0 192.168.1.1/24
daniel@ubuntu:~$ iwconfig
no wireless extensions.
lo
no wireless extensions.
enp6s0
no wireless extensions.
wlp0s20f3 IEEE 802.11 ESSID:"TLON-ADHOC"
Mode:Ad-Hoc Frequency:2.412 GHz Cell: 02:1B:55:AD:0C:02
Tx-Power=22 dBm
Retry short limit:7 RTS thr:off Fragment thr:off
Power Management:on
bat0
no wireless extensions.
daniel@ubuntu:~$ sudo batctl n
[B.A.T.M.A.N. adv 2024.3-6-gd6b622d1, MainIF/MAC: wlp0s20f3/c0:b8:85:de:1e (bat0/da:71:f0:4f:a0:42 BATMAN_IV)]
IF Neighbor last-seen
wlp0s20f3 94:e2:3c:a9:3c:0b 0.840s
daniel@ubuntu:~$
```

Figure 5.1: Configuración red ad-hoc y protocolo BATMAN PC1

```

ubuntu@ubuntu:~$ nmcli connection delete id "USergio_Academia"
La conexión «USergio_Academia» (3f8db667-7bce-421a-9785-112d839e9e2f) se ha borrado correctamente.
ubuntu@ubuntu:~$ nmcli connection delete id "USergio_Admin"
La conexión «USergio_Admin» (ce1eb6c4-f3bc-4934-9ba9-cf9d05ee7f20) se ha borrado correctamente.
ubuntu@ubuntu:~$ sudo nmcli dev set wlo1 managed no
ubuntu@ubuntu:~$ sudo ifconfig wlo1 down
ubuntu@ubuntu:~$ sudo iwconfig wlo1 mode ad-hoc
ubuntu@ubuntu:~$ sudo iwconfig wlo1 essid TLON-ADHOC
ubuntu@ubuntu:~$ sudo iwconfig wlo1 ap 02:1B:55:AD:0C:02
ubuntu@ubuntu:~$ sudo iwconfig wlo1 channel 1
ubuntu@ubuntu:~$ sudo ifconfig wlo1 mtu 1532 up
ubuntu@ubuntu:~$ sudo batctl if add wlo1
ubuntu@ubuntu:~$ sudo ip link set up dev bat0
ubuntu@ubuntu:~$ sudo ifconfig bat0 192.168.1.2/24
ubuntu@ubuntu:~$ iwconfig
lo                no wireless extensions.

wlo1               IEEE 802.11  ESSID:"TLON-ADHOC"
                  Mode:Ad-Hoc  Frequency:2.412 GHz  Cell: 02:1B:55:AD:0C:02
                  Tx-Power=22 dBm
                  Retry short limit:7   RTS thr:off   Fragment thr:off
                  Power Management:on

lxcbr0             no wireless extensions.

bat0               no wireless extensions.

ubuntu@ubuntu:~$ sudo batctl n
[B.A.T.M.A.N. adv 2024.3-6-gd6b622d1, MainIF/MAC: wlo1/94:e2:3c:a9:3c:0b (bat0/4a:8e:2b:c4:74:27 BATMAN_IV)]
IF                Neighbor      last-seen
wlo1              c0:b8:83:85:de:1e  0.915s
ubuntu@ubuntu:~$

```

Figure 5.2: Configuración red ad-hoc y protocolo BATMAN PC2

### 5.1.1 Instalación del Protocolo B.A.T.M.A.N.

Instrucciones detalladas de instalación del protocolo en cada dispositivo.

### 5.1.2 Configuración de la Red Ad-hoc

Paso a paso de cómo configurar cada nodo para unirse a la red Ad-hoc.

## 5.2 Conexión y Pruebas de Conectividad

### 5.2.1 Pruebas de Ping

Resultados de las pruebas de conectividad entre nodos, incluyendo capturas de pantalla de los pings.

### 5.2.2 Pruebas de Rutas (Traceroute)

Descripción de cómo se realizó el traceroute y los resultados obtenidos.

```

daniel@ubuntu:~$ ip a
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: enp6s0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc fq_codel state DOWN group default qlen 1000
    link/ether 08:97:98:85:84:d3 brd ff:ff:ff:ff:ff:ff
3: wlp0s20f3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1532 qdisc noqueue master bat0 state UP group default qlen 1000
    link/ether c0:b8:83:85:de:1e brd ff:ff:ff:ff:ff:ff
    inet6 ::1/24 scope global bat0
        valid_lft forever preferred_lft forever
    inet6 fe80::d871:f0ff:fe4f:a042/64 scope link
        valid_lft forever preferred_lft forever
daniel@ubuntu:~$ ping 192.168.1.2
PING 192.168.1.2 (192.168.1.2) 56(84) bytes of data:
64 bytes from 192.168.1.2: icmp_seq=1 ttl=64 time=6.22 ms
64 bytes from 192.168.1.2: icmp_seq=2 ttl=64 time=4.44 ms
64 bytes from 192.168.1.2: icmp_seq=3 ttl=64 time=2.83 ms
64 bytes from 192.168.1.2: icmp_seq=4 ttl=64 time=1.96 ms
64 bytes from 192.168.1.2: icmp_seq=5 ttl=64 time=1.92 ms
64 bytes from 192.168.1.2: icmp_seq=6 ttl=64 time=1.98 ms
64 bytes from 192.168.1.2: icmp_seq=7 ttl=64 time=1.99 ms
64 bytes from 192.168.1.2: icmp_seq=8 ttl=64 time=2.06 ms
64 bytes from 192.168.1.2: icmp_seq=9 ttl=64 time=3.98 ms
64 bytes from 192.168.1.2: icmp_seq=10 ttl=64 time=1.43 ms
^C
--- 192.168.1.2 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9014ms
rtt min/avg/max/mdev = 1.428/2.882/6.223/1.447 ms
daniel@ubuntu:~$

```

Figure 5.3: IP a después de configurar el PC1

```

64 bytes from 192.168.1.1: icmp_seq=62 ttl=64 time=201 ms
64 bytes from 192.168.1.1: icmp_seq=67 ttl=64 time=614 ms
64 bytes from 192.168.1.1: icmp_seq=70 ttl=64 time=171 ms
64 bytes from 192.168.1.1: icmp_seq=78 ttl=64 time=348 ms
64 bytes from 192.168.1.1: icmp_seq=82 ttl=64 time=181 ms
64 bytes from 192.168.1.1: icmp_seq=87 ttl=64 time=46.5 ms
64 bytes from 192.168.1.1: icmp_seq=95 ttl=64 time=197 ms
64 bytes from 192.168.1.1: icmp_seq=97 ttl=64 time=251 ms
64 bytes from 192.168.1.1: icmp_seq=100 ttl=64 time=12.0 ms
64 bytes from 192.168.1.1: icmp_seq=101 ttl=64 time=442 ms
^C
--- 192.168.1.1 ping statistics ---
110 packets transmitted, 43 received, 60.9091% packet loss, time 110877ms
rtt min/avg/max/mdev = 2.624/134.191/613.530/148.950 ms
ubuntu@ubuntu:~$

```

Figure 5.4: IP a después de configurar el PC2

```

10 packets transmitted, 10 received, 0% packet loss, time 9014ms
rtt min/avg/max/mdev = 1.428/2.882/6.223/1.447 ms
daniel@ubuntu:~$ ping 192.168.1.2
PING 192.168.1.2 (192.168.1.2) 56(84) bytes of data.
64 bytes from 192.168.1.2: icmp_seq=1 ttl=64 time=3.38 ms
64 bytes from 192.168.1.2: icmp_seq=2 ttl=64 time=3.64 ms
64 bytes from 192.168.1.2: icmp_seq=3 ttl=64 time=3.01 ms
64 bytes from 192.168.1.2: icmp_seq=4 ttl=64 time=3.60 ms
64 bytes from 192.168.1.2: icmp_seq=5 ttl=64 time=2.51 ms
64 bytes from 192.168.1.2: icmp_seq=6 ttl=64 time=3.01 ms
^C
--- 192.168.1.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5008ms
rtt min/avg/max/mdev = 2.506/3.193/3.641/0.395 ms
daniel@ubuntu:~$ ping 192.168.1.2
PING 192.168.1.2 (192.168.1.2) 56(84) bytes of data.
64 bytes from 192.168.1.2: icmp_seq=1 ttl=64 time=13.9 ms
64 bytes from 192.168.1.2: icmp_seq=2 ttl=64 time=3.62 ms
64 bytes from 192.168.1.2: icmp_seq=3 ttl=64 time=4.00 ms
64 bytes from 192.168.1.2: icmp_seq=4 ttl=64 time=6.88 ms
64 bytes from 192.168.1.2: icmp_seq=5 ttl=64 time=7.70 ms
64 bytes from 192.168.1.2: icmp_seq=6 ttl=64 time=8.67 ms
64 bytes from 192.168.1.2: icmp_seq=7 ttl=64 time=7.47 ms
64 bytes from 192.168.1.2: icmp_seq=8 ttl=64 time=10.8 ms
64 bytes from 192.168.1.2: icmp_seq=9 ttl=64 time=8.55 ms
64 bytes from 192.168.1.2: icmp_seq=10 ttl=64 time=9.37 ms
64 bytes from 192.168.1.2: icmp_seq=11 ttl=64 time=2.02 ms
^C
--- 192.168.1.2 ping statistics ---
11 packets transmitted, 11 received, 0% packet loss, time 10015ms
rtt min/avg/max/mdev = 2.019/7.546/13.938/3.248 ms
daniel@ubuntu:~$ ping 192.168.1.2
PING 192.168.1.2 (192.168.1.2) 56(84) bytes of data.
64 bytes from 192.168.1.2: icmp_seq=13 ttl=64 time=346 ms
64 bytes from 192.168.1.2: icmp_seq=14 ttl=64 time=397 ms
64 bytes from 192.168.1.2: icmp_seq=23 ttl=64 time=132 ms
64 bytes from 192.168.1.2: icmp_seq=29 ttl=64 time=449 ms
64 bytes from 192.168.1.2: icmp_seq=45 ttl=64 time=86.1 ms
^C
--- 192.168.1.2 ping statistics ---
88 packets transmitted, 5 received, 94.3182% packet loss, time 89058ms
rtt min/avg/max/mdev = 86.062/282.017/448.932/145.804 ms
daniel@ubuntu:~$

```

Figure 5.5: Ping edificio B PC1

```

10 packets transmitted, 10 received, 0% packet loss, time 9014ms
rtt min/avg/max/mdev = 1.428/2.882/6.223/1.447 ms
daniel@ubuntu:~$ ping 192.168.1.2
PING 192.168.1.2 (192.168.1.2) 56(84) bytes of data.
64 bytes from 192.168.1.2: icmp_seq=1 ttl=64 time=3.38 ms
64 bytes from 192.168.1.2: icmp_seq=2 ttl=64 time=3.64 ms
64 bytes from 192.168.1.2: icmp_seq=3 ttl=64 time=3.01 ms
64 bytes from 192.168.1.2: icmp_seq=4 ttl=64 time=3.60 ms
64 bytes from 192.168.1.2: icmp_seq=5 ttl=64 time=2.51 ms
64 bytes from 192.168.1.2: icmp_seq=6 ttl=64 time=3.01 ms
^C
--- 192.168.1.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5008ms
rtt min/avg/max/mdev = 2.506/3.193/3.641/0.395 ms
daniel@ubuntu:~$ ping 192.168.1.2
PING 192.168.1.2 (192.168.1.2) 56(84) bytes of data.
64 bytes from 192.168.1.2: icmp_seq=1 ttl=64 time=13.9 ms
64 bytes from 192.168.1.2: icmp_seq=2 ttl=64 time=3.62 ms
64 bytes from 192.168.1.2: icmp_seq=3 ttl=64 time=4.00 ms
64 bytes from 192.168.1.2: icmp_seq=4 ttl=64 time=6.88 ms
64 bytes from 192.168.1.2: icmp_seq=5 ttl=64 time=7.70 ms
64 bytes from 192.168.1.2: icmp_seq=6 ttl=64 time=8.67 ms
64 bytes from 192.168.1.2: icmp_seq=7 ttl=64 time=7.47 ms
64 bytes from 192.168.1.2: icmp_seq=8 ttl=64 time=10.8 ms
64 bytes from 192.168.1.2: icmp_seq=9 ttl=64 time=8.55 ms
64 bytes from 192.168.1.2: icmp_seq=10 ttl=64 time=9.37 ms
64 bytes from 192.168.1.2: icmp_seq=11 ttl=64 time=2.02 ms
^C
--- 192.168.1.2 ping statistics ---
11 packets transmitted, 11 received, 0% packet loss, time 10015ms
rtt min/avg/max/mdev = 2.019/7.546/13.938/3.248 ms
daniel@ubuntu:~$ ping 192.168.1.2
PING 192.168.1.2 (192.168.1.2) 56(84) bytes of data.
64 bytes from 192.168.1.2: icmp_seq=13 ttl=64 time=346 ms
64 bytes from 192.168.1.2: icmp_seq=14 ttl=64 time=397 ms
64 bytes from 192.168.1.2: icmp_seq=23 ttl=64 time=132 ms
64 bytes from 192.168.1.2: icmp_seq=29 ttl=64 time=449 ms
64 bytes from 192.168.1.2: icmp_seq=45 ttl=64 time=86.1 ms
^C
--- 192.168.1.2 ping statistics ---
88 packets transmitted, 5 received, 94.3182% packet loss, time 89058ms
rtt min/avg/max/mdev = 86.062/282.017/448.932/145.804 ms
daniel@ubuntu:~$

```

Figure 5.6: Ping edificio B PC2

# Chapter 6

## Conclusiones

### 6.1 Conclusiones Generales

Resumen de los principales hallazgos del proyecto y el cumplimiento de los objetivos.

# **Appendix A**

## **Anexos**

### **A.1 Código y Configuraciones**

Incluye aquí fragmentos de código o configuraciones específicas utilizadas en los dispositivos.

### **A.2 Imágenes Adicionales**

Incluye imágenes de la implementación física o cualquier diagrama adicional que complemente el informe.