

Escuela Colombiana de Ingeniería Julio Garavito

Nikolas Bernal Giraldo – Diego Alejandro Murcia Céspedes

Redes de Computadores – RECO

## LABORATORIO No.02 – ALISTAMIENTO S.O, SHELL Y SOFTWARE DE APOYO EN REDES

### INTRODUCCIÓN

---

Como ya hemos hablado, una empresa normalmente cuenta con varios servicios de infraestructura TI. En ella se encuentran estaciones de usuario alámbricas e inalámbricas y servidores (físicos y virtualizados), todos estos conectados a través de switches (capa 2 y 3), equipos inalámbricos y routers que lo conectan a Internet. También es común contar con infraestructuras en la nube desde donde se provisionan recursos según las necesidades de la organización. Dentro de los servidores se pueden encontrar servicios web, DNS, correo, base de datos, almacenamiento y aplicaciones, entre otros.

### MARCO TEORICO

---

Comenzaremos explicando como se utiliza el software ofrecido por CISCO llamado Packet Tracer, con el cual es posible diseñar redes y realizar simulaciones sobre su uso, se puede testear el funcionamiento de redes, ciberseguridad y el internet de las cosas. Además de esto se puede trabajar sobre proyectos preconstruidos, probar nuevos diseños y topologías de red.

Ahora, en cuanto a los dos virtualizadores que usaremos, VMWARE es un software ofrecido por VMWare Inc. que permite la virtualización de equipos que corran diferentes tipos de sistemas operativos con una configuración de hardware hecha por nosotros. Este software funciona para Windows, Linux y macOS que corren procesadores Intel. Y en cuanto a Virtualbox, esta es una herramienta que ofrece las mismas funcionalidades de VMWare, virtualizando arquitecturas x84/amd64. Su desarrolladora es Oracle Corporation. Gracias a este virtualizador es posible instalar sistemas operativos adicionales, cada uno separado por su ambiente virtual. Ofrece la posibilidad de ejecutar máquinas virtuales de forma remota y soporte de iSCSI, y tiene una excelente emulación de hardware al momento de crear una máquina virtual o hacer uso de una.

También tendremos el software de Wireshark, el cual es ofrecido y desarrollado por The Wireshark Team, el cual, además de ser un software libre, nos permitirá realizar análisis de los protocolos en la red y solucionar problemas con las redes de comunicaciones. Su funcionalidad es similar a la de TCPDUMP, pero añadiendo una interfaz grafica y opciones de organización y filtrados de información.

## EXPERIMENTOS

---

### 1. CONOCIENDO PACKET TRACER

- Responda las siguientes preguntas
  1. ¿Qué versión de Packet Tracer se encuentra instalada en el Lab?

La versión instalada es la 7.2.2.

2. A través de la plataforma Cisco inscribise en el curso Introduction to Packet Tracer v1.1. Muestre con un video hecho por el grupo un resumen del curso. Máximo 7 min.

Link del video: <https://youtu.be/ICPMsPlf8eg>

3. Realice la evaluación del curso y tome un pantallazo del resultado de la evaluación.

Diego Alejandro Murcia Céspedes:



Introduction to Packet Tracer English 0820 cga: View: User report						
User report - Diego Alejandro Murcia Cespedes						
<a href="#">Overview report</a> <a href="#">User report</a>						
Grade item	Calculated weight	Grade	Range	Percentage	Feedback	Contribution to course total
Introduction to Packet Tracer English 0820 cga						
Introduction to Packet Tracer - PT Basics Quiz	0.00 %	100.00	0-100	100.00 %		0.00 %
Introduction to Packet Tracer - PT IoT Basics Quiz	0.00 %	100.00	0-100	100.00 %		0.00 %
Course Completion						
End of Course Feedback	100.00 %	100.00	0-100	100.00 %		100.00 %
Course Completion total	100.00 %	100.00	0-100	100.00 %		-
Course total	-	100.00	0-100	100.00 %		-

## Nikolas Bernal Giraldo:

User report - Nikolas Bernal Giraldo						
Overview report		User report				
Grade Item	Calculated weight	Grade	Range	Percentage	Feedback	Contribution to course total
Introduction to Packet Tracer English 0820 cga						
Introduction to Packet Tracer - PT Basics Quiz	0.00 %	100.00	0-100	100.00 %		0.00 %
Introduction to Packet Tracer - PT IoT Basics Quiz	0.00 %	100.00	0-100	100.00 %		0.00 %
Course Completion						
End of Course Feedback	100.00 %	100.00	0-100	100.00 %		100.00 %
Course Completion total	100.00 %	100.00	0-100	100.00 %		-
Course total	-	100.00	0-100	100.00 %		-

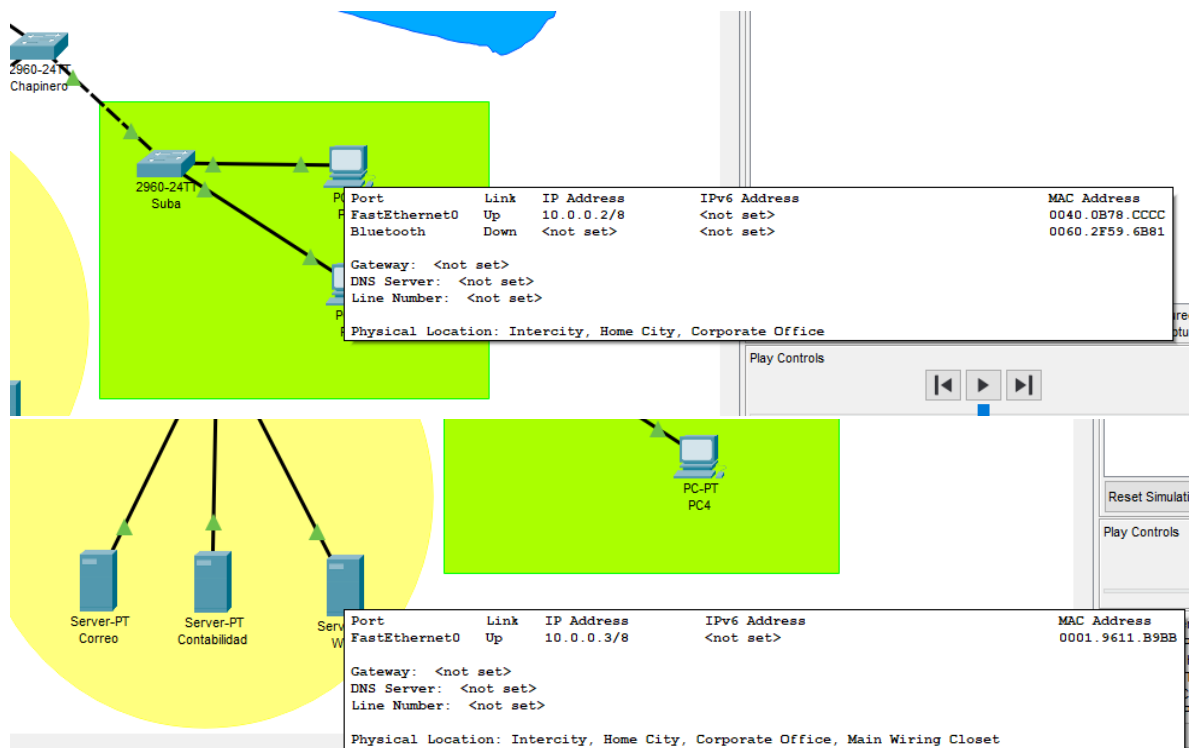
- Usando Packet Tracer haga el diagrama de red que se presenta en la página siguiente. Nota:
  - No tenga en cuenta los colores de los puntos que aparecen en los enlaces (los enlaces son las líneas de conexión entre dispositivos. Más adelante serán importantes los colores de dichos puntos, pero en su momento los revisaremos.
  - Las conexiones o enlaces que se presentan en el diagrama son:
    - Las de color negro corresponden a cables Ethernet (Ethernet, FastEthernet o GigaEthernet).
      - ¿Qué significan las conexiones negras continuas?

Significan **“Copper Straight-Throgh”**, y sirve para conectar equipos como PC, servidor y router a un switch, o puente.
      - ¿Qué significan las conexiones negras discontinuas?

Significan **“Copper Cross-Over”**, y sirven para las conexiones de HUB con HUB, de un PC a otro, de switch con switch, o entre el equipo de la misma capa de acuerdo al modelo **OSI**.
    - Las de color rojo son seriales (Conexiones típicamente WAN). Al dibujarlas en packet tracer aparecerán un poco diferente respecto al dibujo.

## 2. SIGUIENDO MENSAJES CON PACKET TRACER

- Seleccione dos computadores ubicados en el cuadrado verde y el círculo amarillo. Póngales la siguiente configuración
  - PC3
    - IP 10.0.0.2
    - Máscara: 255.0.0.0
  - Server-PT Web
    - IP 10.0.0.3
    - Máscara 255.0.0.0



- Entre en el modo simulación con que cuenta Packet Tracer y revise los PDUs por capas (Todavía no hemos visto el significado de lo que cada uno tiene, pero vea que existen y que cada capa adiciona información a los datos de usuario). Para esto use la siguiente información como guía.

PacketTracer7

IPv4

IPv6

Misc

☐ ARP

☐ BGP

☐ DHCP

☐ DNS

☐ EIGRP

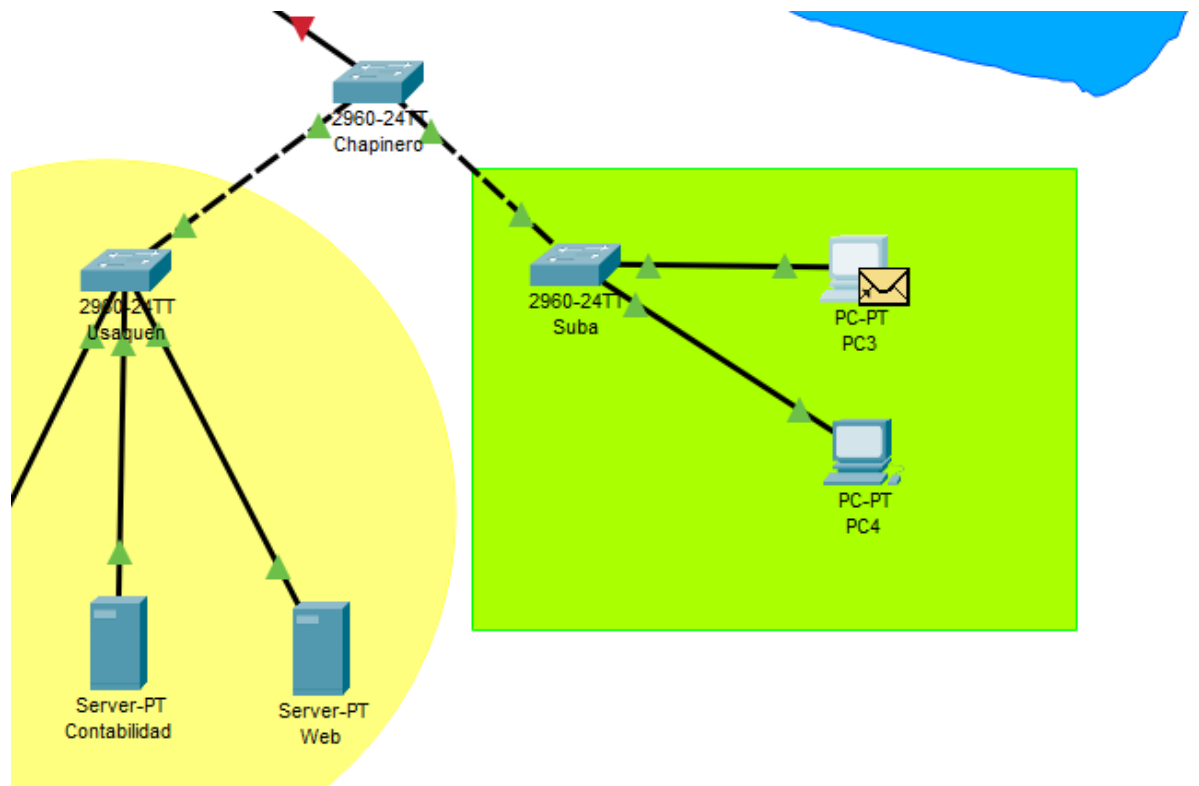
☐ HSRP

☒ ICMP


☐ OSPF

☐ RIP

Edit ACL Filters



# Event List

Vis.	Time(sec)	Last Device	At Device	Type
	0.000	--	PC3	ICMP
	0.001	PC3	Suba	ICMP
	0.002	Suba	Chapinero	ICMP
	0.003	Chapinero	Usaquen	ICMP
	0.004	Usaquen	Web	ICMP
	0.005	Web	Usaquen	ICMP
	0.006	Usaquen	Chapinero	ICMP
	0.007	Chapinero	Suba	ICMP
	0.008	Suba	PC3	ICMP

Captured to:

```
C:\>ping 10.0.0.3
```

```
Pinging 10.0.0.3 with 32 bytes of data:
```

```
Reply from 10.0.0.3: bytes=32 time=8ms TTL=128
```

```
Reply from 10.0.0.3: bytes=32 time=8ms TTL=128
```

```
Reply from 10.0.0.3: bytes=32 time=8ms TTL=128
```

```
Reply from 10.0.0.3: bytes=32 time=8ms TTL=128
```

```
Ping statistics for 10.0.0.3:
```

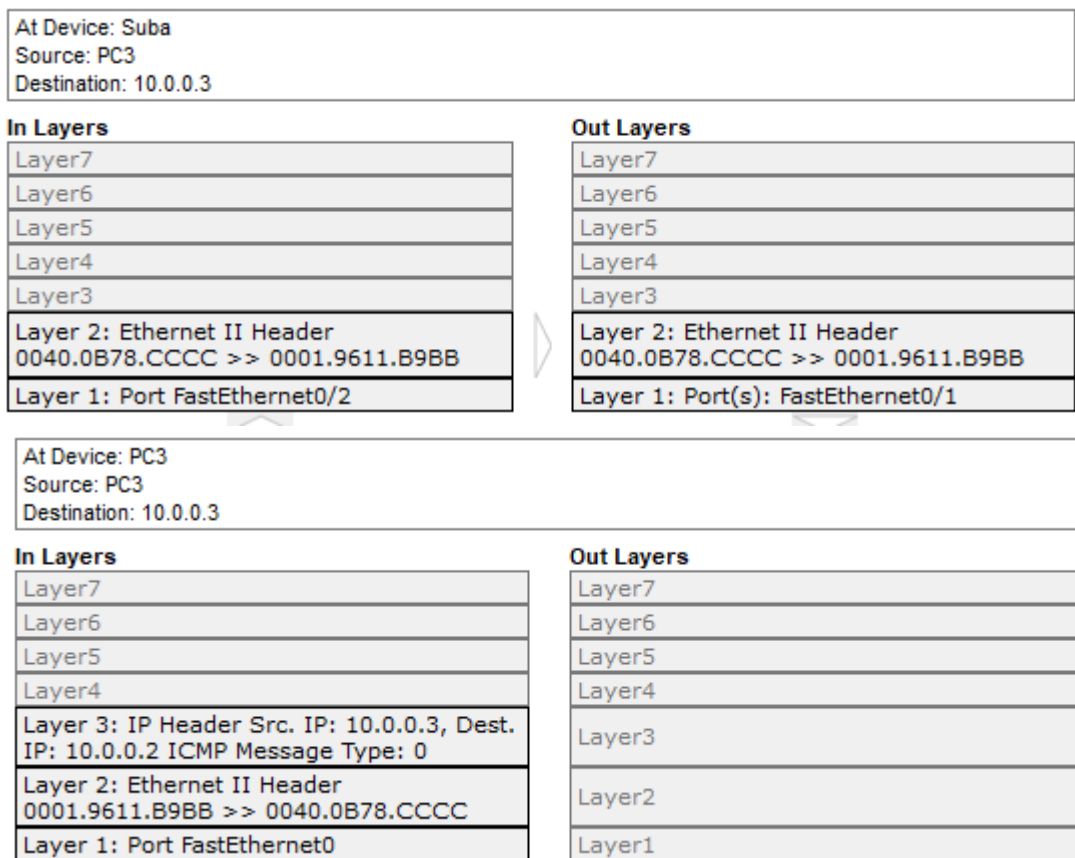
```
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
```

```
Approximate round trip times in milli-seconds:
```

```
    Minimum = 8ms, Maximum = 8ms, Average = 8ms
```

```
C:\>|
```

- Revise el contenido de los paquetes capturados. Revise el contenido del encabezado de capa 2.



## MONTAJE REAL

### 1. USANDO WIRE SHARK

Wireshark es una herramienta multiplataforma utilizada para realizar análisis sobre paquetes de red. La utilizaremos dentro del curso para observar, en tiempo real, lo datos que pasan por la red y la manera de operación de los diferentes protocolos que estudiaremos. Por tal razón

- Ejecute Wireshark en el computador en el que está trabajando

Ejecutado.



- Revise los siguientes videos
  - Wireshark Tutorial for Beginners.

<https://www.youtube.com/watch?v=TkCSr3OUoIM>

- Wireshark Tutorial for Beginners 2017 – Overview of the Enviroment.

<https://www.youtube.com/watch?v=6LGw31TsP6E>

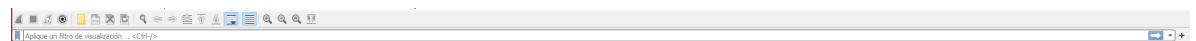
- Wireshark Demo.

<https://www.youtube.com/watch?v=PYoXowOCppc>

- ¿Qué es Wireshark?

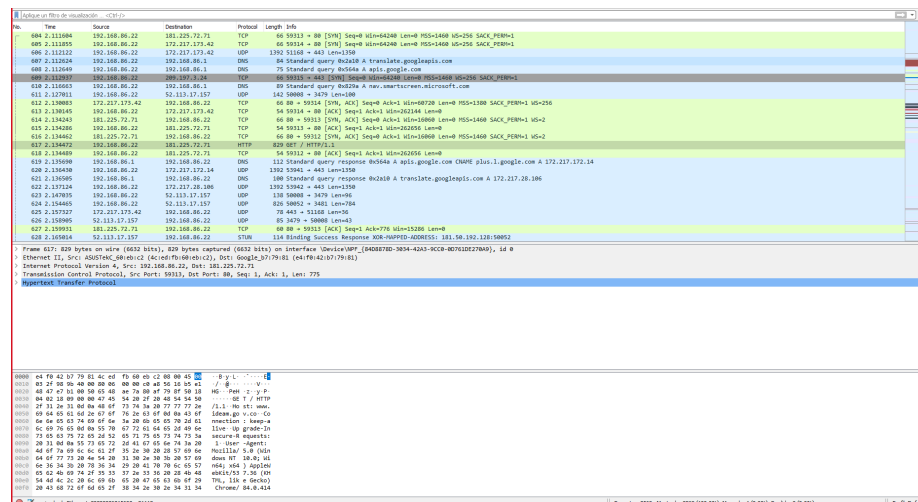
Es un open **source network scanner and monitor** que permite ver el tráfico en la red en paquetes individuales.

- ¿Cómo generar filtros? ¿Para qué se usan? De unos ejemplos.



Se usan para reducir la información que se nos muestra al momento de navegar, mostrándonos únicamente lo que queramos ver. Por ejemplo, queremos mirar solo las capturas de un protocolo **HTTP**, con el filtro solo veremos esto.

- Realice una consulta web al link <http://www.ideam.gov.co/> y capture el tráfico generado (para eso, ingrese al browser, inicie la captura con Wireshark y visite a la página indicada, termine la captura). Finalmente pare la captura.



- Analice los datos encontrados en uno de los paquetes capturados. Mire el encapsulamiento y presente capturas del mismo (Use el paquete que contiene una de las solicitudes GET que se realizan).

No.	Time	Source	Destination	Protocol	Length	Info
617	2.334472	192.168.86.22	181.225.72.71	HTTP	829	GET / HTTP/1.1
1150	3.560619	192.168.86.22	172.217.20.110	HTTP	770	GET /translate_a/element.js?cb=googleTranslateElementInit HTTP/1.1
1162	3.655195	172.217.20.110	192.168.86.22	HTTP	1437	HTTP/1.1 200 OK (text/javascript)
1184	3.712038	192.168.86.22	181.225.72.71	HTTP	976	GET /tema-theme/css/all-alloy-font-awesome-font/fontawesome-alloy.woff HTTP/1.1
1230	3.757539	181.225.72.71	192.168.86.22	HTTP	138	HTTP/1.1 200 OK
1263	3.907323	192.168.86.22	181.225.72.62	HTTP	787	GET /IdeasApp2/Ideas/es/getPronostico/11000000?callback=jQuery300011460500577850552_159849672938_15984967294 HTTP/1.1
1464	4.190979	192.168.86.22	181.225.72.71	HTTP	818	GET / HTTP/1.1
1559	4.383582	192.168.86.22	181.225.72.62	HTTP	804	GET /IdeasApp2/Ideas/es/getPronostico/11000000?callback=jQuery19008679180175812613_1598496757808_15984967571 HTTP/1.1
1579	4.405553	192.168.86.22	181.225.72.62	HTTP	646	GET /IdeasApp2/Ideas/es/getAlarmasNacionales HTTP/1.1
1586	4.405529	181.225.72.62	192.168.86.22	HTTP	343	HTTP/1.1 304 Not Modified
1594	4.580327	181.225.72.62	192.168.86.22	HTTP	1514	Continuation (application/json)
1608	4.553274	192.168.86.22	181.225.72.71	HTTP	929	GET /html/js/liferay/available_languages.jsp?browserId=other&themeId=Tema_uAR_TemaTheme&colorSchemeId=01&iniferType=js&languageId=es_ES&0-62108&-1557674744000 HTTP/1.1
1670	4.597096	181.225.72.71	192.168.86.22	HTTP	816	HTTP/1.1 200 OK (text/javascript)

>	Frame 617: 829 bytes on wire (6632 bits), 829 bytes captured (6632 bits) on Interface 0 (Device VPCS {64088780-3634-42A3-9CC8-40761DE276A0}), id 0
>	Ethernet II, Src: ASUSNetK.6@eb:c2 (4c:ed:fb:60:eb:c2), Dst: Google_b7:79:81 (e4:fb:42:b7:79:81)
>	Internet Protocol Version 4, Src: 192.168.86.22, Dst: 181.225.72.71
>	Transmission Control Protocol, Src Port: 59513, Dst Port: 80, Seq: 1, Ack: 1, Len: 775
>	Hypertext Transfer Protocol
>	GET / HTTP/1.1\r\n
>	Host: www.ideam.gov.co\r\n
>	Connection: keep-alive\r\n
>	Upgrade-Insecure-Requests: 1\r\n
>	User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/84.0.4147.135 Safari/537.36 Edg/84.0.522.63\r\n
>	Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9\r\n
>	Accept-Encoding: gzip, deflate\r\n
>	Accept-Language: es-419;es;q=0.9,es-ES;q=0.8,en;q=0.7,en-GB;q=0.6,en-US;q=0.5\r\n
>	[Truncated Cookies: COOKIE_SUPPORT=true; GUEST_LANGUAGE_ID=es_ES; cookieSessionId=70828746D130N0PQSTUUM0V2A6880; __utma=114624095.783100407.1598496737.1598496737.1598496737.1; __utmc=114624095.1598496737.1.1.utmcx=(direct)utccn=(dire
>	\r\n
>	[Full request URL: http://www.ideam.gov.co/]
>	[HTTP request 1/3]
>	[Next request in frame: 1184]

## SOFTWARE BASE

### 1. PRUEBAS DE USO DEL LABORATORIO DE INFORMÁTICA

- Slackware
  - Configuración de IP:

```

root@NikolasDiego:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.2.77.190 netmask 255.255.0.0 broadcast 10.2.255.255
    inet6 fe80::20c:29ff:fe6a:a6b8 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:fa:a6:b8 txqueuelen 1000 (Ethernet)
    RX packets 856 bytes 55260 (53.9 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 8 bytes 672 (672.0 B)
    TX errors 0 dropped 0 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 1 (Local Loopback)
    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 0 overruns 0 carrier 0 collisions 0

```

➤ PING:

```
root@NikolasDiego:~# ping 10.2.77.190
PING 10.2.77.190 (10.2.77.190) 56(84) bytes of data.
64 bytes from 10.2.77.190: icmp_seq=1 ttl=64 time=0.017 ms
64 bytes from 10.2.77.190: icmp_seq=2 ttl=64 time=0.044 ms
64 bytes from 10.2.77.190: icmp_seq=3 ttl=64 time=0.039 ms
64 bytes from 10.2.77.190: icmp_seq=4 ttl=64 time=0.040 ms
64 bytes from 10.2.77.190: icmp_seq=5 ttl=64 time=0.040 ms
^C
```

```
root@NikolasDiego:~# ping 10.2.65.1
PING 10.2.65.1 (10.2.65.1) 56(84) bytes of data.
64 bytes from 10.2.65.1: icmp_seq=1 ttl=64 time=2.31 ms
64 bytes from 10.2.65.1: icmp_seq=2 ttl=64 time=0.636 ms
64 bytes from 10.2.65.1: icmp_seq=3 ttl=64 time=1.47 ms
64 bytes from 10.2.65.1: icmp_seq=4 ttl=64 time=1.46 ms
64 bytes from 10.2.65.1: icmp_seq=5 ttl=64 time=1.45 ms
64 bytes from 10.2.65.1: icmp_seq=6 ttl=64 time=1.46 ms
64 bytes from 10.2.65.1: icmp_seq=7 ttl=64 time=1.28 ms
^C
```

```
root@NikolasDiego:~# ping 10.2.65.1
PING 10.2.65.1 (10.2.65.1) 56(84) bytes of data.
64 bytes from 10.2.65.1: icmp_seq=1 ttl=64 time=2.31 ms
64 bytes from 10.2.65.1: icmp_seq=2 ttl=64 time=0.636 ms
64 bytes from 10.2.65.1: icmp_seq=3 ttl=64 time=1.47 ms
64 bytes from 10.2.65.1: icmp_seq=4 ttl=64 time=1.46 ms
64 bytes from 10.2.65.1: icmp_seq=5 ttl=64 time=1.45 ms
64 bytes from 10.2.65.1: icmp_seq=6 ttl=64 time=1.46 ms
64 bytes from 10.2.65.1: icmp_seq=7 ttl=64 time=1.28 ms
^C
```

```
root@NikolasDiego:~# ping 10.2.77.156
PING 10.2.77.156 (10.2.77.156) 56(84) bytes of data.
64 bytes from 10.2.77.156: icmp_seq=1 ttl=128 time=2.98 ms
64 bytes from 10.2.77.156: icmp_seq=2 ttl=128 time=1.48 ms
64 bytes from 10.2.77.156: icmp_seq=3 ttl=128 time=1.45 ms
64 bytes from 10.2.77.156: icmp_seq=4 ttl=128 time=1.33 ms
64 bytes from 10.2.77.156: icmp_seq=5 ttl=128 time=1.39 ms
64 bytes from 10.2.77.156: icmp_seq=6 ttl=128 time=1.09 ms
64 bytes from 10.2.77.156: icmp_seq=7 ttl=128 time=1.35 ms
^C
--- 10.2.77.156 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6011ms
rtt min/avg/max/mdev = 1.090/1.584/2.982/0.583 ms
root@NikolasDiego:~#
```

- CentOS

- Configuración de IP:

```
root@NikolasDiego:~# ping 10.2.65.1
PING 10.2.65.1 (10.2.65.1) 56(84) bytes of data.
64 bytes from 10.2.65.1: icmp_seq=1 ttl=64 time=2.31 ms
64 bytes from 10.2.65.1: icmp_seq=2 ttl=64 time=0.636 ms
64 bytes from 10.2.65.1: icmp_seq=3 ttl=64 time=1.47 ms
64 bytes from 10.2.65.1: icmp_seq=4 ttl=64 time=1.46 ms
64 bytes from 10.2.65.1: icmp_seq=5 ttl=64 time=1.45 ms
64 bytes from 10.2.65.1: icmp_seq=6 ttl=64 time=1.46 ms
64 bytes from 10.2.65.1: icmp_seq=7 ttl=64 time=1.28 ms
^C
```

- PING:

```
[root@localhost ~]# ping 10.2.77.186
PING 10.2.77.186 (10.2.77.186) 56(84) bytes of data.
64 bytes from 10.2.77.186: icmp_seq=1 ttl=64 time=0.032 ms
64 bytes from 10.2.77.186: icmp_seq=2 ttl=64 time=0.071 ms
64 bytes from 10.2.77.186: icmp_seq=3 ttl=64 time=0.096 ms
64 bytes from 10.2.77.186: icmp_seq=4 ttl=64 time=0.069 ms
^C
--- 10.2.77.186 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 0.032/0.067/0.096/0.022 ms
[root@localhost ~]# _
```

```
[root@localhost ~]# ping 10.2.65.1
PING 10.2.65.1 (10.2.65.1) 56(84) bytes of data.
64 bytes from 10.2.65.1: icmp_seq=1 ttl=64 time=1.76 ms
64 bytes from 10.2.65.1: icmp_seq=2 ttl=64 time=1.67 ms
64 bytes from 10.2.65.1: icmp_seq=3 ttl=64 time=1.48 ms
64 bytes from 10.2.65.1: icmp_seq=4 ttl=64 time=1.62 ms
^C
--- 10.2.65.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 1.488/1.636/1.763/0.107 ms
[root@localhost ~]#
```

```
[root@localhost ~]# ping 10.2.77.156
PING 10.2.77.156 (10.2.77.156) 56(84) bytes of data.
64 bytes from 10.2.77.156: icmp_seq=1 ttl=128 time=3.13 ms
64 bytes from 10.2.77.156: icmp_seq=2 ttl=128 time=1.90 ms
64 bytes from 10.2.77.156: icmp_seq=3 ttl=128 time=1.65 ms
64 bytes from 10.2.77.156: icmp_seq=4 ttl=128 time=1.70 ms
64 bytes from 10.2.77.156: icmp_seq=5 ttl=128 time=1.64 ms
^C
--- 10.2.77.156 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4014ms
rtt min/avg/max/mdev = 1.647/2.008/3.138/0.574 ms
[root@localhost ~]#
```

- Windows Server
  - Configuración de IP:

```
C:\Users\Administrator>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

    Connection-specific DNS Suffix  . : 
    Link-local IPv6 Address . . . . . : fe80::41e4:b3eb:b88e:4f60%3
    IPv4 Address. . . . . : 10.2.77.191
    Subnet Mask . . . . . : 255.255.0.0
    Default Gateway . . . . . : fe80::20c:29ff:fe47:d7e8%3
                                10.2.65.1

Tunnel adapter isatap.{83F74168-7C49-4126-95C7-649863D15926}:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . : 

C:\Users\Administrator>
```

- PING:

```
C:\Users\Administrator>ping 10.2.77.191

Pinging 10.2.77.191 with 32 bytes of data:
Reply from 10.2.77.191: bytes=32 time<1ms TTL=128
Reply from 10.2.77.191: bytes=32 time<1ms TTL=128
Reply from 10.2.77.191: bytes=32 time<1ms TTL=128
Reply from 10.2.77.191: bytes=32 time<1ms TTL=128

Ping statistics for 10.2.77.191:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\Administrator>
```

```
C:\Users\Administrator>ping 10.2.65.1

Pinging 10.2.65.1 with 32 bytes of data:
Reply from 10.2.65.1: bytes=32 time=2ms TTL=64
Reply from 10.2.65.1: bytes=32 time=1ms TTL=64
Reply from 10.2.65.1: bytes=32 time=1ms TTL=64
Reply from 10.2.65.1: bytes=32 time=1ms TTL=64

Ping statistics for 10.2.65.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 2ms, Average = 1ms

C:\Users\Administrator>
```



```
C:\Users\Administrator>ping 8.8.8.8

Pinging 8.8.8.8 with 32 bytes of data:
Reply from 8.8.8.8: bytes=32 time=45ms TTL=112
Reply from 8.8.8.8: bytes=32 time=45ms TTL=112
Reply from 8.8.8.8: bytes=32 time=45ms TTL=112
Reply from 8.8.8.8: bytes=32 time=45ms TTL=112

Ping statistics for 8.8.8.8:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 45ms, Maximum = 45ms, Average = 45ms

C:\Users\Administrator>

C:\Users\Administrator>ping 10.2.77.156

Pinging 10.2.77.156 with 32 bytes of data:
Reply from 10.2.77.156: bytes=32 time=2ms TTL=128
Reply from 10.2.77.156: bytes=32 time=2ms TTL=128
Reply from 10.2.77.156: bytes=32 time=1ms TTL=128
Reply from 10.2.77.156: bytes=32 time=2ms TTL=128

Ping statistics for 10.2.77.156:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 2ms, Average = 1ms

C:\Users\Administrator>
```

## 2. BOURNE SHELL PROGRAMMING – UNIX

### 1. Ejecución Automática de una Secuencia de Comandos

Revisar:

`/programs/EjecucionAutomaticaDeUnaSecuenciaDeComandos.sh`

### 2. Manejo de Variables

Revisar:

`/programs/ManejoDeVariables.sh`

### 3. Uso de Repeticiones y almacenamiento de la Respuesta de Ejecución de un Comando en un Archivo

Revisar:

`/programs/UsoDeRepeticionesAlmacenamiento.sh`

#### 4. Manejo de Condicionales

- Al listar el comando `ls -l` y ver los campos de permisos podemos ver que hay 10 caracteres en la primera columna. En el primero nos representara su tipo, es decir archivo (`-`), un directorio (`d`), un archivo de bloques especiales (`b`), un archivo de caracteres especiales (`c`), un archivo de vinculo o enlace (`l`) y por último está el archivo especial de cauce (`p`). Los siguientes nueve caracteres representan los permisos que se le conceden a los usuarios, grupos y otros, cada tres caracteres hacen referencia a los permisos de cada uno estos. Los valores que puede tomar son permisos (`-`), permiso de lectura (`r`), permiso de escritura (`w`) y permiso de ejecución (`x`).
- Revisar:  
`/programs/ManejoDeCondicionales.sh`

#### 5. Revisión de log

Revisar:  
`/programs/RevisionDeLog.sh`

#### 6. Creación de Usuarios

Revisar:  
`/programs/CreacionDeUsuarios.sh`

## CONCLUSIONES







---

En este laboratorio logramos conocer mas a fondo el modo de operación de herramientas de red como lo son Packet Tracer y Wireshark, con los cuales podremos visualizar y diseñar diferentes tipos de redes, así como observar sus protocolos. Además, con la ayuda de una VPN pudimos usar los computadores de la universidad para trabajar sobre ellos e instalar los diferentes sistemas operativos que usamos, logrando también configurar la red para cada uno de estos, con una IP estática seleccionada por nosotros, y una mascara de red y gateway brindadas por la profesora.

Desarrollamos diferentes programas en shell sobre las distribuciones de Linux instaladas, los cuales nos permitieron sobre la administración de sistemas operativos.

## BIBLIOGRAFIA

---

-  <https://ccnadesdecero.es/cable-directo-cruzado-y-consola-diferencias/>
-  <https://www.ambit-bst.com/blog/todo-lo-que-debes-saber-de-cisco-packet-tracer>
-  <https://es.wikipedia.org/wiki/VirtualBox>
-  <https://www.vmware.com/es/topics/glossary/content/hypervisor.html>
-  <https://www.universidadviu.com/software-de-virtualizacion-que-es/>
-  <https://es.wikipedia.org/wiki/Wireshark>