

LABORATORIO 2 RECO



Miguel Angel Fuquene Arias

David Leonardo Coronado

ESCUELA COLOMBIANA DE INGENIERÍA JULIO GARAVITO

BOGOTÁ D.C. 26 DE AGOSTO 2020

Miguel Angel Fuquene Arias

David Leonardo Coronado

Laboratorio No. 2 – Alistamiento -RECO

Objetivo

- Continuar la instalación de sistemas operativos base.
- Conocer el modo de operación de herramientas de redes.
- Conocer sobre administración de sistemas operativos usando programas en Shell

Herramientas a utilizar

- Computadores Laboratorio de Informática
- Acceso a Internet
- Software de virtualización
- Imágenes de Sistemas Operativos
- Packet tracer
- Wireshark

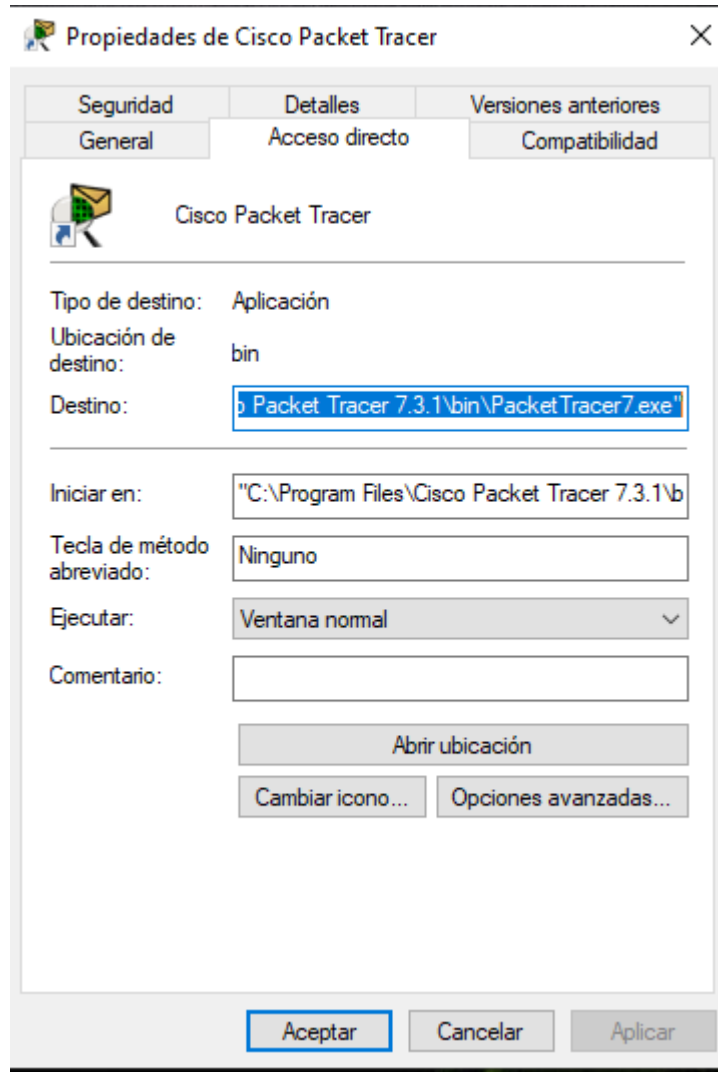
Introducción

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.

1. Conociendo Packet Tracer

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

Tenemos la versión 7.3.1



2. A través de la plataforma de 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.

Anexado.

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

Miguel Angel Fuquene:

Mi perfil

Mi perfil

Certificados/cartas

Vales de descuento

Finalizar sesión

[illegible]

Finalizar revisión

Correcta

Correcta

Puntúa 2,00



David Leonardo

My Pro

certification centers

✓	✓	✓	✓	✓

✓

4 Log Out

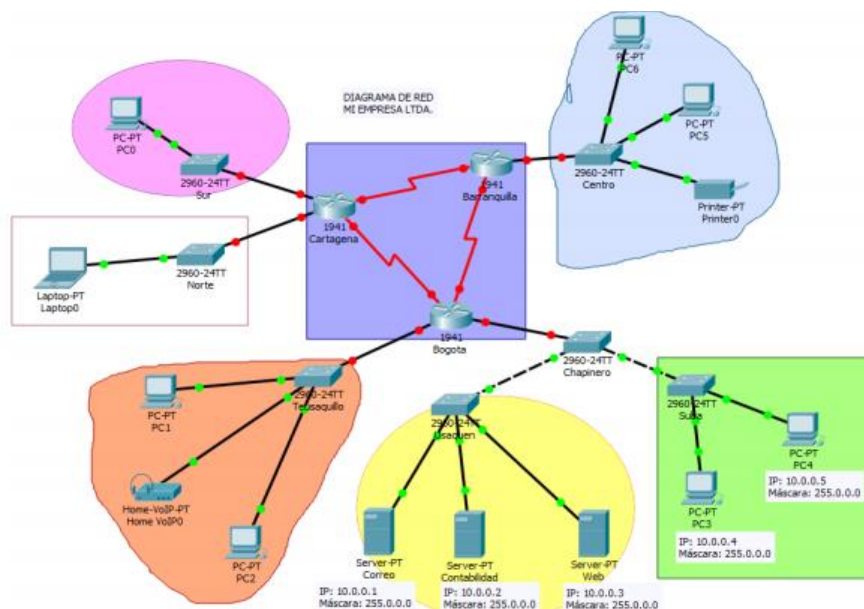
12

Started on	Friday, 28 August 2020, 4:40 AM
State	Finished
Completed on	Friday, 28 August 2020, 5:13 AM
Time taken	33 mins
Marks	22.17/23.00
Grade	96.38 out of 100.00

Quiz navigation

1	2	3	4
✓	✓	●	✓
9	10	11	12
✓	✓	✓	✓

- _____



- 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: o Las de color negro corresponden a cables Ethernet (Ethernet, FastEthernet o GigaEthernet).

Anexo.

- ¿Qué significan las conexiones negras continuas?

- Las conexiones negras continuas representan conexiones de punto a punto , podríamos decir que es una conexión por cable coaxial.

- ¿Qué significan las conexiones negras discontinuas?

- Las conexiones negras discontinuas representan conexiones punto a punto de tipo trenzado.

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

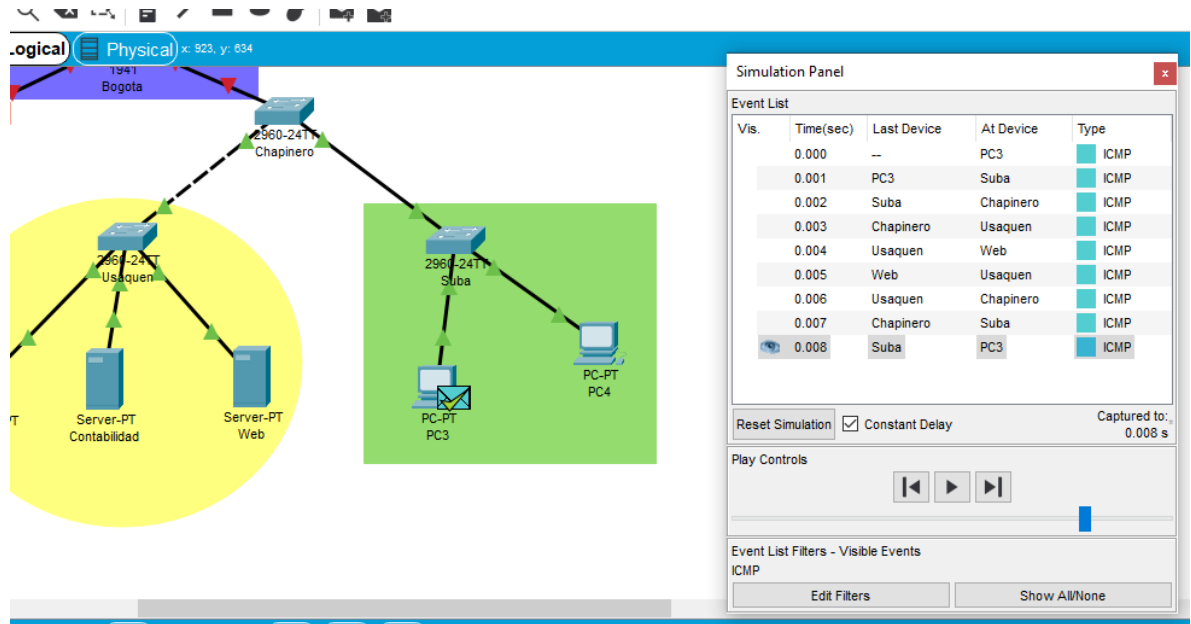
- Máscara: 255.0.0.0

- 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



Run the simulation and capture the traffic¹.

- In the far lower right of the PT interface is the toggle between Realtime and Simulation mode. Click on Simulation mode.
- Click in the Edit filters button and select only ICMP.
- Click in PCA. Choose the Desktop tab. Open the Command Prompt. Enter the command ping IP_PCB). Pressing the Enter key will initiate four ICMP echo requests. Minimize the PC configuration window. Two packets appear in the Event List, the first ICMP echo request and an ARP request needed to resolve the IP address of the server to its hardware MAC address.
- Click the Auto Capture / Play button to run the simulation and capture events. Click OK when the "No More Events" message is reached.

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

Montaje real

Realice las siguientes pruebas usando la herramienta Wireshark.

1. Usando Wireshark

- ¿Qué es Wireshark?

Wireshark es una herramienta que sirve para analizar de protocolos, actualmente está disponible para Windows y Unix. Su principal objetivo es el análisis y la interceptación de tráfico para convertirlo en un formato legible para las personas. Esto hace que sea más fácil identificar qué tráfico está cruzando la red, con qué frecuencia y la latencia que hay entre ciertos saltos, además es una excelente aplicación didáctica para el estudio de las comunicaciones y para la resolución de problemas de red.

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

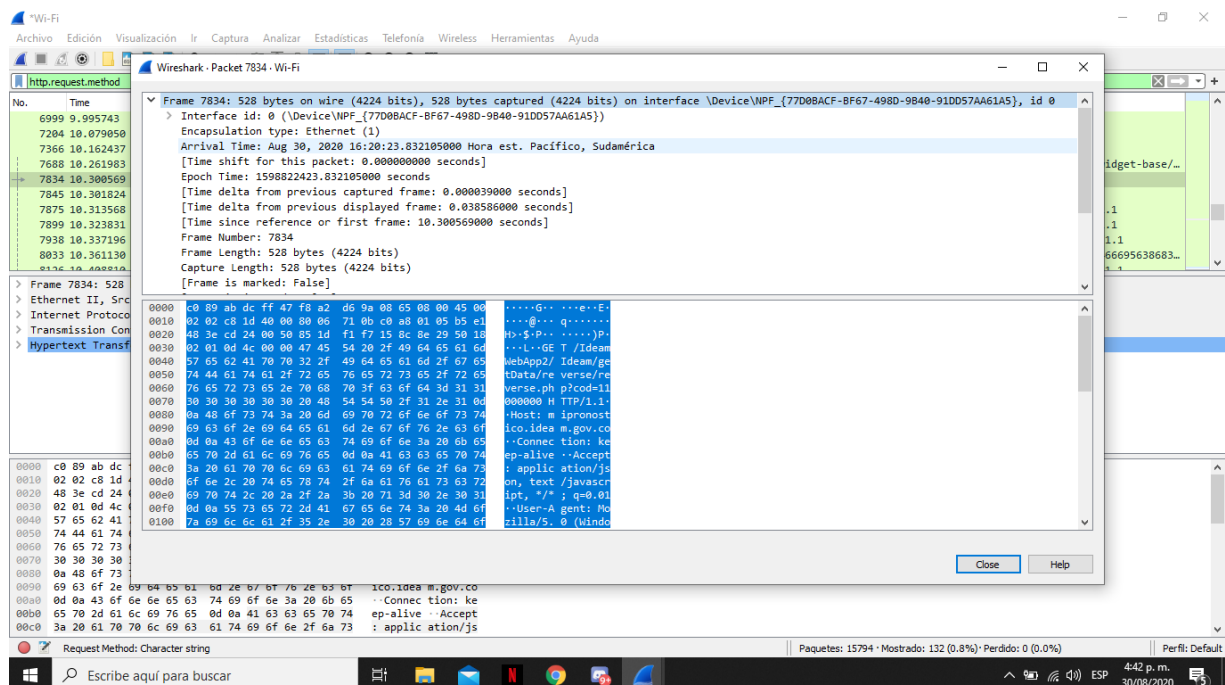
Filtros de captura: Son aplicados para vigilar paquetes de manera selectiva. Filtraré o capturaré únicamente el tráfico definido. Para hacer esta captura selectiva se debe pasar el comando de instrucciones hacia Wireshark.

La opción del filtro de captura puede ser ejecutado haciendo clic en el icono del menú “Edit Capture Filter” o Editar filtro de captura. De manera alterna, puede también ser ejecutado haciendo clic en “Capture -> Options” o Captura -> Opciones.

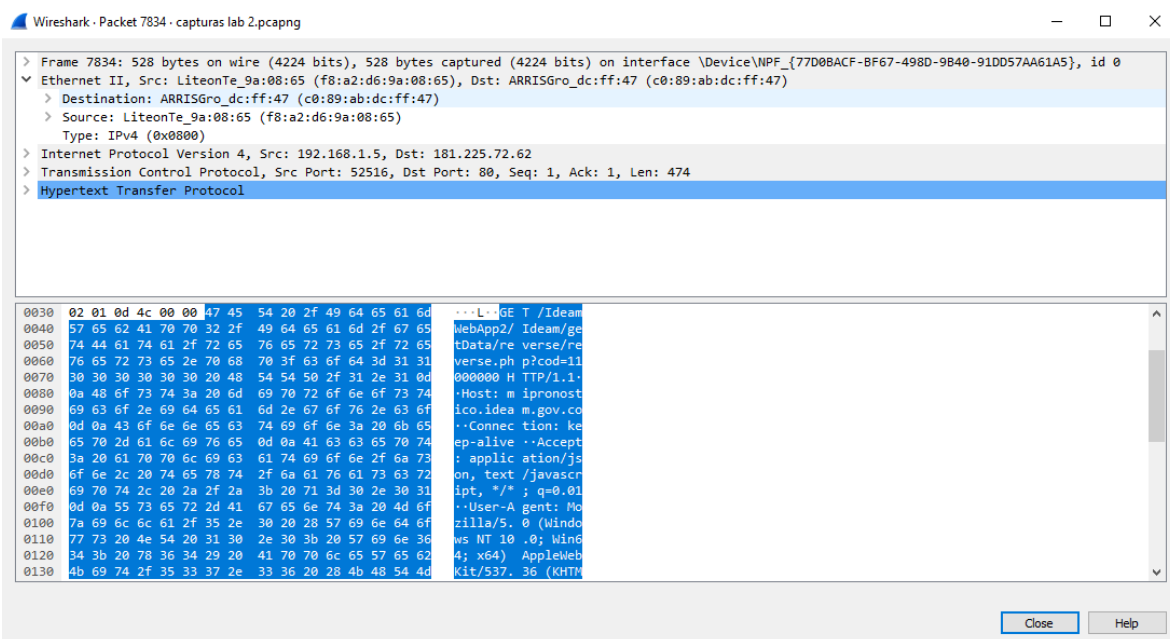
###Filtros de muestra: Tienen una estructura definida. Por ejemplo; para visualizar los paquetes TCP capturados sobre el puerto 443 se puede utilizar el siguiente filtro de muestra.###

- 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).

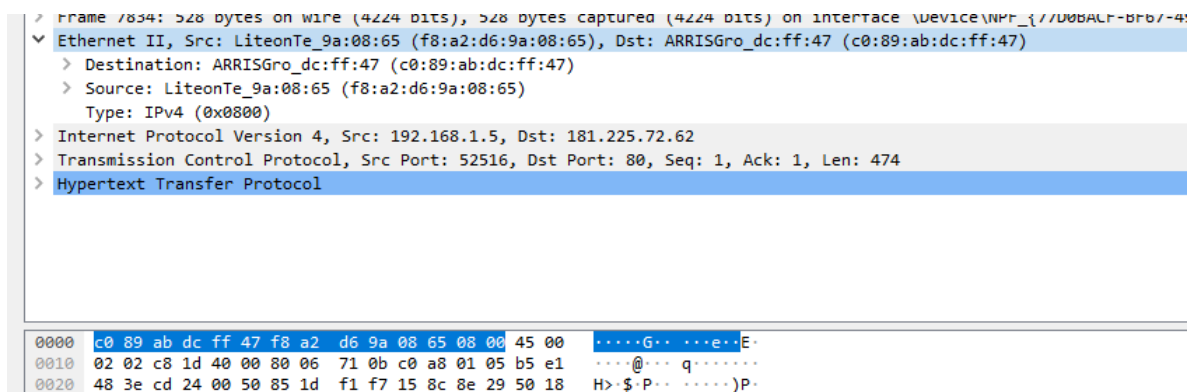
Como se puede observar en este paquete del tráfico de datos, podemos encontrar un número que identifica el Frame(7834), muestra el interface Id y muestra que la encapsulación es de tipo ethernet, muestra unos deltas de tiempo en que tomo la captura del anterior paquete capturado y este en el cual estamos y muestra la longitud del frame que en este caso es de 528 bytes.



Cuando analizamos el protocolo Ethernet podemos ver una dirección en sistema sexagesimal del origen y del destino y el tipo que corresponde a IPv4,



Si le damos clic a Ethernet podemos observar en la parte de abajo como esta representado en sistemas sexagesimal, por 14 numeros donde los 6 primeros corresponden al destino, otros 6 a el origen y 2 que son del IPv4



Si le damos en Hipertext nos mostrara información como el host, así como también una petición de datos como un objeto JSON, Salen los agentes usuarios que si no estoy mal se refiere a los buscadores en los que se puede abrir dicho paquete y sale la dirección de origen de donde viene(www.ideam.gov.co)

```
▼ Hypertext Transfer Protocol
> GET /IdeamWebApp2/Ideam/getData/reverse/reverse.php?cod=11000000 HTTP/1.1\r\n
Host: mipronostico.ideam.gov.co\r\n
Connection: keep-alive\r\n
Accept: application/json, text/javascript, */*; q=0.01\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\r\n
Origin: http://www.ideam.gov.co\r\n
Referer: http://www.ideam.gov.co\r\n
Accept-Encoding: gzip, deflate\r\n
Accept-Language: es-CO,es;q=0.9,es-419;q=0.8,en;q=0.7\r\n
\r\n
[Full request URI: http://mipronostico.ideam.gov.co/IdeamWebApp2/Ideam/getData/reverse/reverse.php?cod=11000000]
[HTTP request 1/1]
```

Software Base

Realizaremos alistamientos de computadores de una red usando como base el Laboratorio de Informática.

Adicionalmente, dentro de la infraestructura también se requiere contar con programas que apoyen la

administración de diferentes actividades del sistema operativo. Vamos a realizar actividades que les

ayuden a entender un poco el sistema operativo y la gestión de usuarios.

1. Pruebas de uso del Laboratorio de Informática

Vamos a realizar pruebas de operación de los equipos del Laboratorio de Informática. Para esto, se les

enviará por grupos instrucciones de acceso a una máquina del Laboratorio. Luego de que haya entrado

realice las siguientes actividades

- Instale Linux Skackware, centos y Windows Server (instalación básica) usando VirtualBox y

VMWARE en dicha máquina de acuerdo como se hizo en el laboratorio No. 1

- Configure la dirección IP así

o DIR_IP: 10.2.77.n (El profesor asignará un rango a cada grupo. No se pueden repetir

direcciones IP en los equipos instalados)

o Mascara:255.255.0.0

o Gateway: 10.2.65.1

• Usando el comando ping, haga las siguientes pruebas

o ping 10.2.77.n

o ping 10.2.65.1

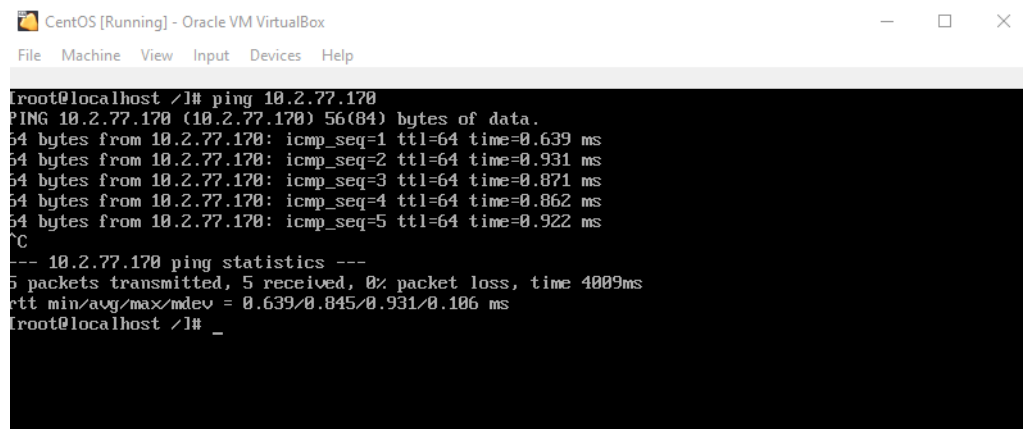
o ping 8.8.8.8

o ping 10.2.77.m (máquina de otros compañeros)

NOTA: Muestre a su profesor la ejecución de las máquinas virtuales.

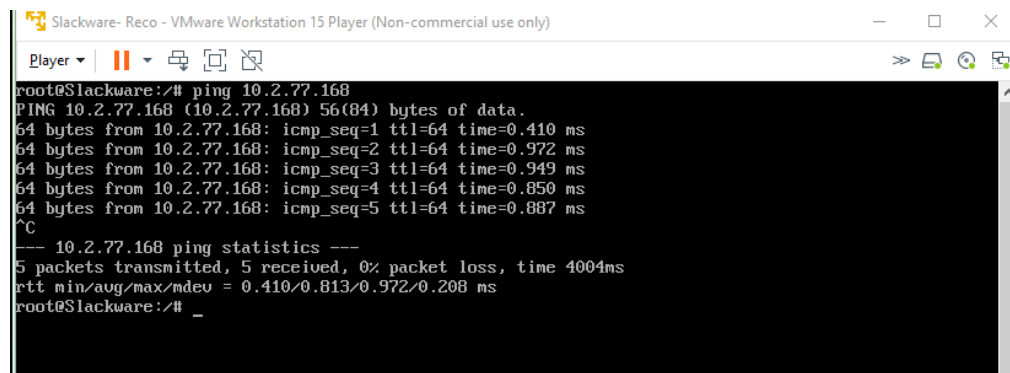
Ping de nuestras maquinas

Ping 10.2.77.170



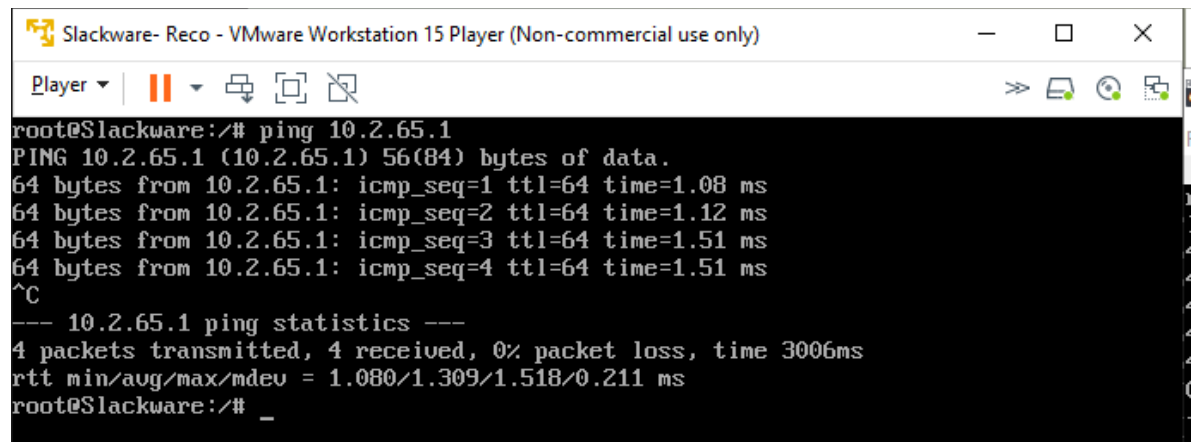
```
CentOS [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
root@localhost ~# ping 10.2.77.170
PING 10.2.77.170 (10.2.77.170) 56(84) bytes of data.
64 bytes from 10.2.77.170: icmp_seq=1 ttl=64 time=0.639 ms
64 bytes from 10.2.77.170: icmp_seq=2 ttl=64 time=0.931 ms
64 bytes from 10.2.77.170: icmp_seq=3 ttl=64 time=0.871 ms
64 bytes from 10.2.77.170: icmp_seq=4 ttl=64 time=0.862 ms
64 bytes from 10.2.77.170: icmp_seq=5 ttl=64 time=0.922 ms
^C
--- 10.2.77.170 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4009ms
rtt min/avg/max/mdev = 0.639/0.845/0.931/0.106 ms
root@localhost ~# _
```

Ping 10.2.77.168

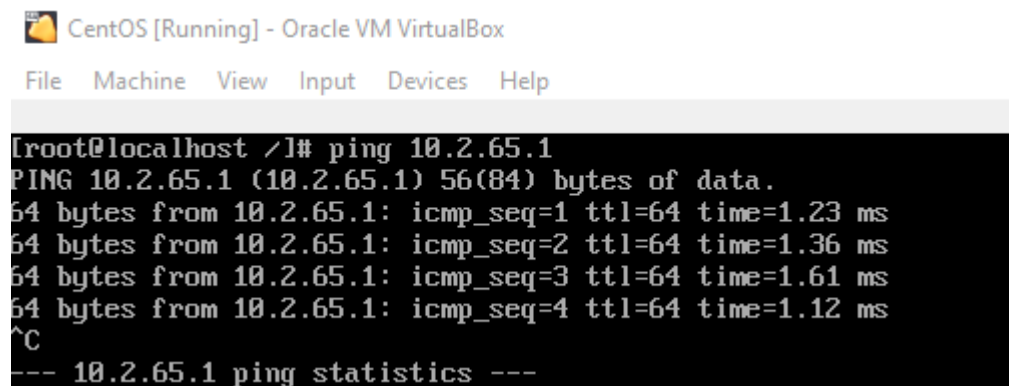


```
Slackware- Reco - VMware Workstation 15 Player (Non-commercial use only)
Player
root@Slackware:~# ping 10.2.77.168
PING 10.2.77.168 (10.2.77.168) 56(84) bytes of data.
64 bytes from 10.2.77.168: icmp_seq=1 ttl=64 time=0.410 ms
64 bytes from 10.2.77.168: icmp_seq=2 ttl=64 time=0.972 ms
64 bytes from 10.2.77.168: icmp_seq=3 ttl=64 time=0.949 ms
64 bytes from 10.2.77.168: icmp_seq=4 ttl=64 time=0.850 ms
64 bytes from 10.2.77.168: icmp_seq=5 ttl=64 time=0.887 ms
^C
--- 10.2.77.168 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4004ms
rtt min/avg/max/mdev = 0.410/0.813/0.972/0.208 ms
root@Slackware:~# _
```

Ping a 10.2.65.1

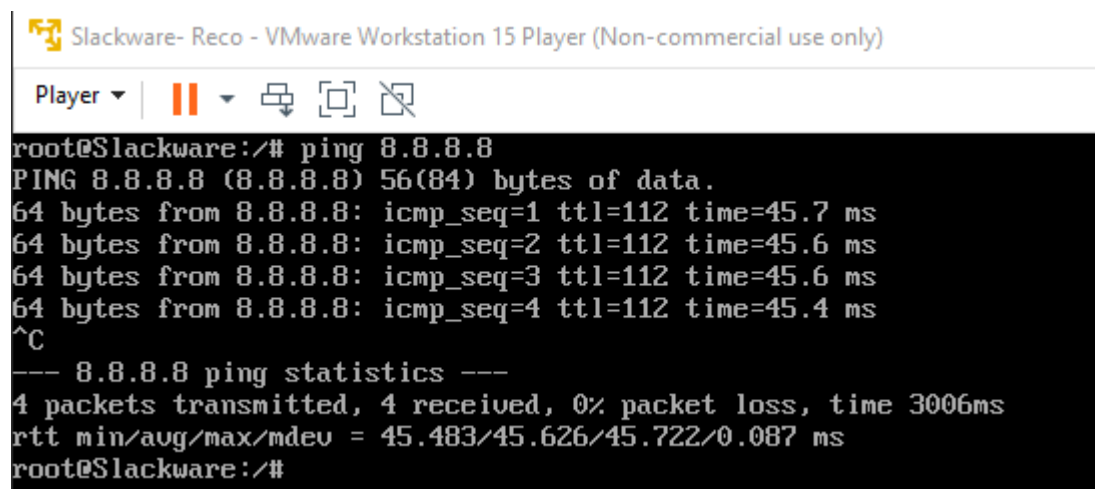


```
root@Slackware:~# ping 10.2.65.1
PING 10.2.65.1 (10.2.65.1) 56(84) bytes of data.
64 bytes from 10.2.65.1: icmp_seq=1 ttl=64 time=1.08 ms
64 bytes from 10.2.65.1: icmp_seq=2 ttl=64 time=1.12 ms
64 bytes from 10.2.65.1: icmp_seq=3 ttl=64 time=1.51 ms
64 bytes from 10.2.65.1: icmp_seq=4 ttl=64 time=1.51 ms
^C
--- 10.2.65.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 1.080/1.309/1.518/0.211 ms
root@Slackware:~#
```



```
[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=1.23 ms
64 bytes from 10.2.65.1: icmp_seq=2 ttl=64 time=1.36 ms
64 bytes from 10.2.65.1: icmp_seq=3 ttl=64 time=1.61 ms
64 bytes from 10.2.65.1: icmp_seq=4 ttl=64 time=1.12 ms
^C
--- 10.2.65.1 ping statistics ---
```

Ping 8.8.8.8



```
root@Slackware:~# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=112 time=45.7 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=112 time=45.6 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=112 time=45.6 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=112 time=45.4 ms
^C
--- 8.8.8.8 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 45.483/45.626/45.722/0.087 ms
root@Slackware:~#
```

```
CentOS [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

[root@localhost ~]# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=112 time=46.0 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=112 time=46.0 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=112 time=45.7 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=112 time=45.9 ms
^C
--- 8.8.8.8 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 45.784/45.965/46.073/0.285 ms
[root@localhost ~]#
```

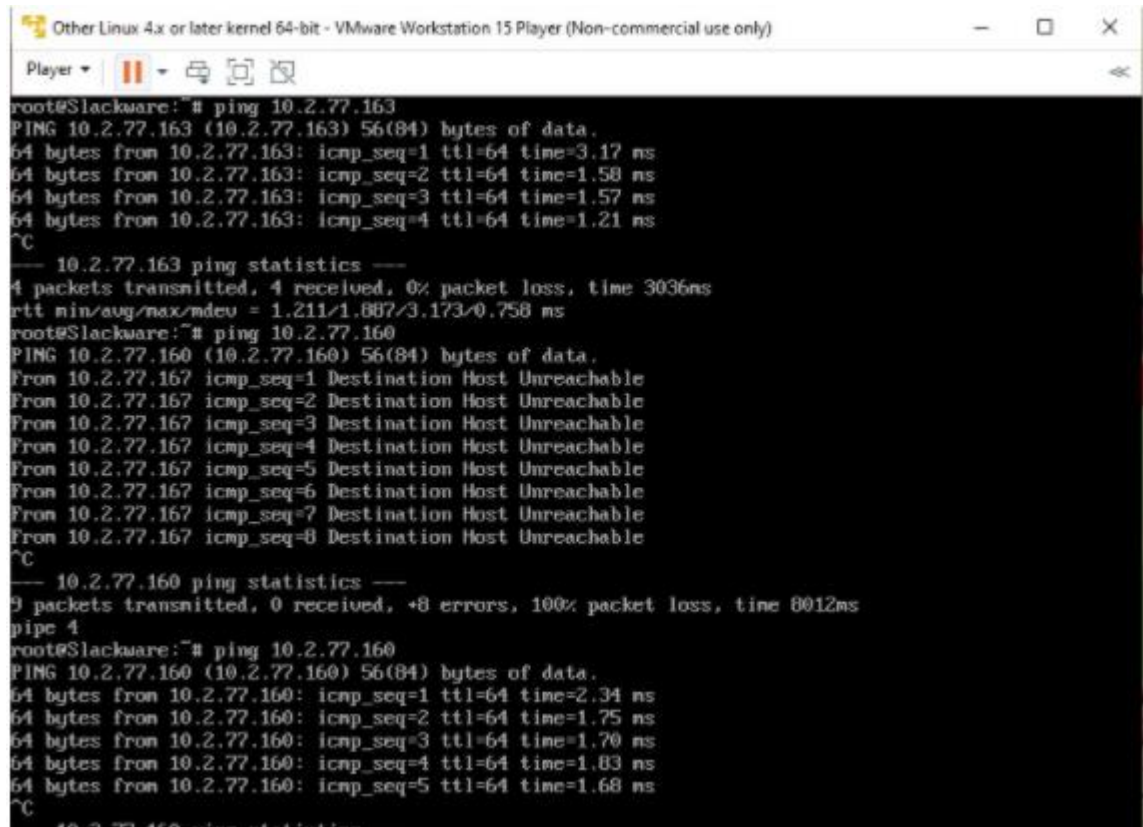
Ping maquina compa era BERNAL HERNANDEZ LAURA ALEJANDRA

```
Slackware- Reco - VMware Workstation 15 Player (Non-commercial use only)
Player
64 bytes from 8.8.8.8: icmp_seq=2 ttl=112 time=45.6 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=112 time=45.8 ms
^C
--- 8.8.8.8 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 45.660/46.155/46.948/0.618 ms
root@Slackware:/home/scripts# ping 10.2.77.196
PING 10.2.77.196 (10.2.77.196) 56(84) bytes of data.
64 bytes from 10.2.77.196: icmp_seq=1 ttl=64 time=3.50 ms
64 bytes from 10.2.77.196: icmp_seq=2 ttl=64 time=2.10 ms
64 bytes from 10.2.77.196: icmp_seq=3 ttl=64 time=2.11 ms
64 bytes from 10.2.77.196: icmp_seq=4 ttl=64 time=2.25 ms
64 bytes from 10.2.77.196: icmp_seq=5 ttl=64 time=1.83 ms
64 bytes from 10.2.77.196: icmp_seq=6 ttl=64 time=2.19 ms
64 bytes from 10.2.77.196: icmp_seq=7 ttl=64 time=2.04 ms
64 bytes from 10.2.77.196: icmp_seq=8 ttl=64 time=2.02 ms
64 bytes from 10.2.77.196: icmp_seq=9 ttl=64 time=1.81 ms
^C
```

```
CentOS [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

[root@localhost ~]# ping 10.2.77.196
PING 10.2.77.196 (10.2.77.196) 56(84) bytes of data.
64 bytes from 10.2.77.196: icmp_seq=1 ttl=64 time=3.61 ms
64 bytes from 10.2.77.196: icmp_seq=2 ttl=64 time=1.79 ms
64 bytes from 10.2.77.196: icmp_seq=3 ttl=64 time=1.97 ms
64 bytes from 10.2.77.196: icmp_seq=4 ttl=64 time=2.21 ms
64 bytes from 10.2.77.196: icmp_seq=5 ttl=64 time=2.03 ms
64 bytes from 10.2.77.196: icmp_seq=6 ttl=64 time=2.16 ms
^C
--- 10.2.77.196 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5012ms
rtt min/avg/max/mdev = 1.795/2.301/3.617/0.606 ms
[root@localhost ~]# s
```

Ping maquinas compañeros ANDRES FELIPE DAVILA Y JAVIER ESTEBAN LOPEZ



```
Other Linux 4.x or later kernel 64-bit - VMware Workstation 15 Player (Non-commercial use only)
Player
root@Slackware:~# ping 10.2.77.163
PING 10.2.77.163 (10.2.77.163) 56(84) bytes of data:
64 bytes from 10.2.77.163: icmp_seq=1 ttl=64 time=3.17 ms
64 bytes from 10.2.77.163: icmp_seq=2 ttl=64 time=1.58 ms
64 bytes from 10.2.77.163: icmp_seq=3 ttl=64 time=1.57 ms
64 bytes from 10.2.77.163: icmp_seq=4 ttl=64 time=1.21 ms
^C
--- 10.2.77.163 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3036ms
rtt min/avg/max/mdev = 1.211/1.887/3.173/0.758 ms
root@Slackware:~# ping 10.2.77.160
PING 10.2.77.160 (10.2.77.160) 56(84) bytes of data:
From 10.2.77.167 icmp_seq=1 Destination Host Unreachable
From 10.2.77.167 icmp_seq=2 Destination Host Unreachable
From 10.2.77.167 icmp_seq=3 Destination Host Unreachable
From 10.2.77.167 icmp_seq=4 Destination Host Unreachable
From 10.2.77.167 icmp_seq=5 Destination Host Unreachable
From 10.2.77.167 icmp_seq=6 Destination Host Unreachable
From 10.2.77.167 icmp_seq=7 Destination Host Unreachable
From 10.2.77.167 icmp_seq=8 Destination Host Unreachable
^C
--- 10.2.77.160 ping statistics ---
8 packets transmitted, 0 received, +8 errors, 100% packet loss, time 8012ms
pipe 4
root@Slackware:~# ping 10.2.77.160
PING 10.2.77.160 (10.2.77.160) 56(84) bytes of data:
64 bytes from 10.2.77.160: icmp_seq=1 ttl=64 time=2.34 ms
64 bytes from 10.2.77.160: icmp_seq=2 ttl=64 time=1.75 ms
64 bytes from 10.2.77.160: icmp_seq=3 ttl=64 time=1.70 ms
64 bytes from 10.2.77.160: icmp_seq=4 ttl=64 time=1.83 ms
64 bytes from 10.2.77.160: icmp_seq=5 ttl=64 time=1.68 ms
^C
--- 10.2.77.160 ping statistics ---
```

2. Bourne Shell programming- Unix

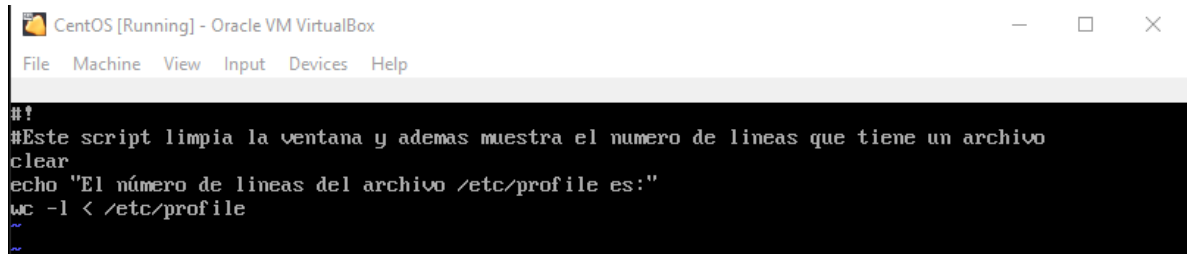
Usando las máquinas virtuales Linux que instaló para el curso, desarrolle las siguientes aplicaciones

(recuerde documentar su código). Revise que operen en ambas distribuciones de Linux.

1. Ejecución automática de una secuencia de comandos

Escriba un programa Shell que:

- Limpie la pantalla
- Imprima el mensaje “El número de líneas del archivo /etc/profile es:” y el número de líneas encontrados.



```
CentOS [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

#!
#Este script limpia la ventana y ademas muestra el numero de lineas que tiene un archivo
clear
echo "El número de lineas del archivo /etc/profile es:"
wc -l < /etc/profile
```

2. Manejo de variables

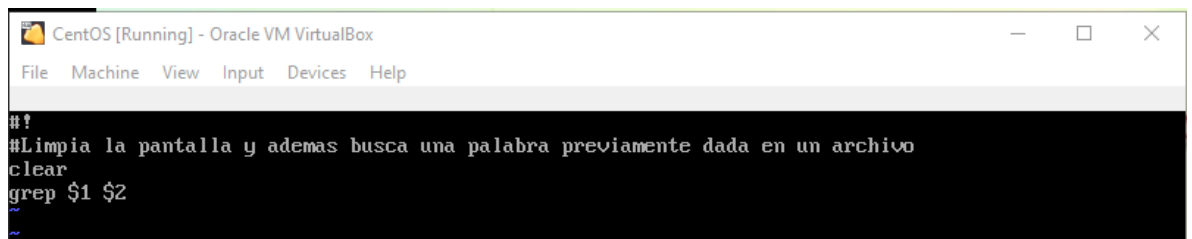
Escriba un programa Shell que:

- Limpie la pantalla
- Busque una palabra dada por el usuario en un archivo especificado por el mismo.
- La ejecución sería del estilo

\$ buscar_palabra.sh palabra_buscada archivo_de_búsqueda,
ejemplo

\$ buscar_palabra.sh casa /etc/passwd

- Imprima el resultado de la búsqueda.

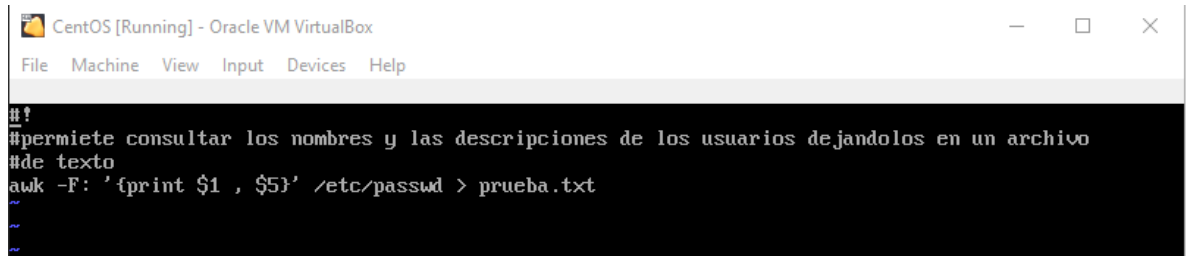


```
CentOS [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

#!
#Limpia la pantalla y ademas busca una palabra previamente dada en un archivo
clear
grep $1 $2
```

3. Uso de repeticiones y almacenamiento de la respuesta de ejecución de un comando en un archivo

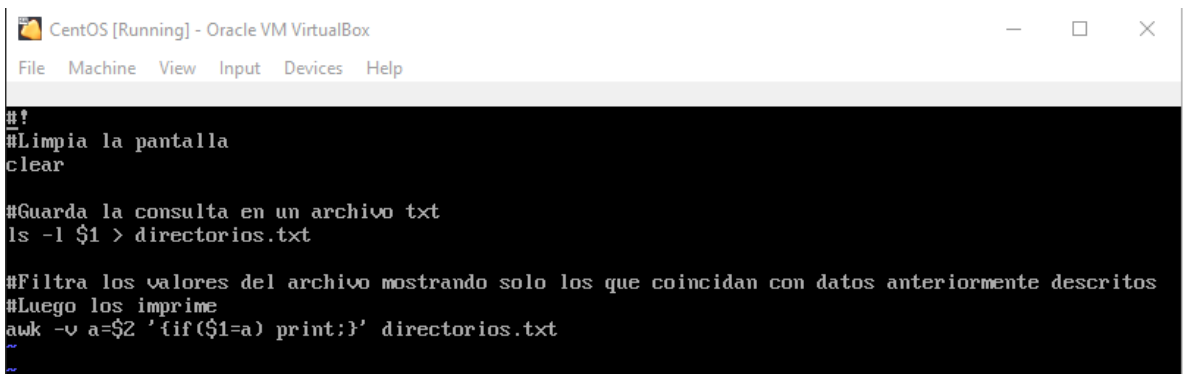
- Cree 5 usuarios en su sistema. Incluya descripción para cada uno de ellos.
- Escriba un programa Shell que:
 - o Del archivo /etc/passwd extraiga SOLAMENTE los nombres de los usuarios y la descripción de los mismos
 - o Deje el resultado de la ejecución en otro archivo



```
#?
#permiete consultar los nombres y las descripciones de los usuarios dejandolos en un archivo
#de texto
awk -F: '{print $1 , $5}' /etc/passwd > prueba.txt
```

4. Manejo de condicionales

- Al listar las características de un archivo con el comando `ls -l`, el campo de permisos está compuesto por 10 caracteres, ¿Qué significa cada uno de ellos? y qué valores pueden tomar?
- Escriba un programa Shell que:
 - o Liste los archivos ubicados en un directorio dado por el usuario que tengan los permisos buscados por el mismo usuario.
 - o La ejecución sería del estilo `$ buscar_archivos.sh ruta permisos`, ejemplo `$ buscar_archivos.sh /etc/ -rw-r--r--`
 - o Muestre el resultado en pantalla



```
#?
#Limpia la pantalla
clear

#Guarda la consulta en un archivo txt
ls -l $1 > directorios.txt

#Filtra los valores del archivo mostrando solo los que coincidan con datos anteriormente descritos
#Luego los imprime
awk -v a=$2 '{if($1=a) print:}' directorios.txt
```

5. Revisión de log

- Escriba un programa Shell que:
- Limpie la pantalla
 - Permita, con un menú, hacer una de las siguientes actividades
 - Muestre las últimas 20 líneas del archivo `messages`

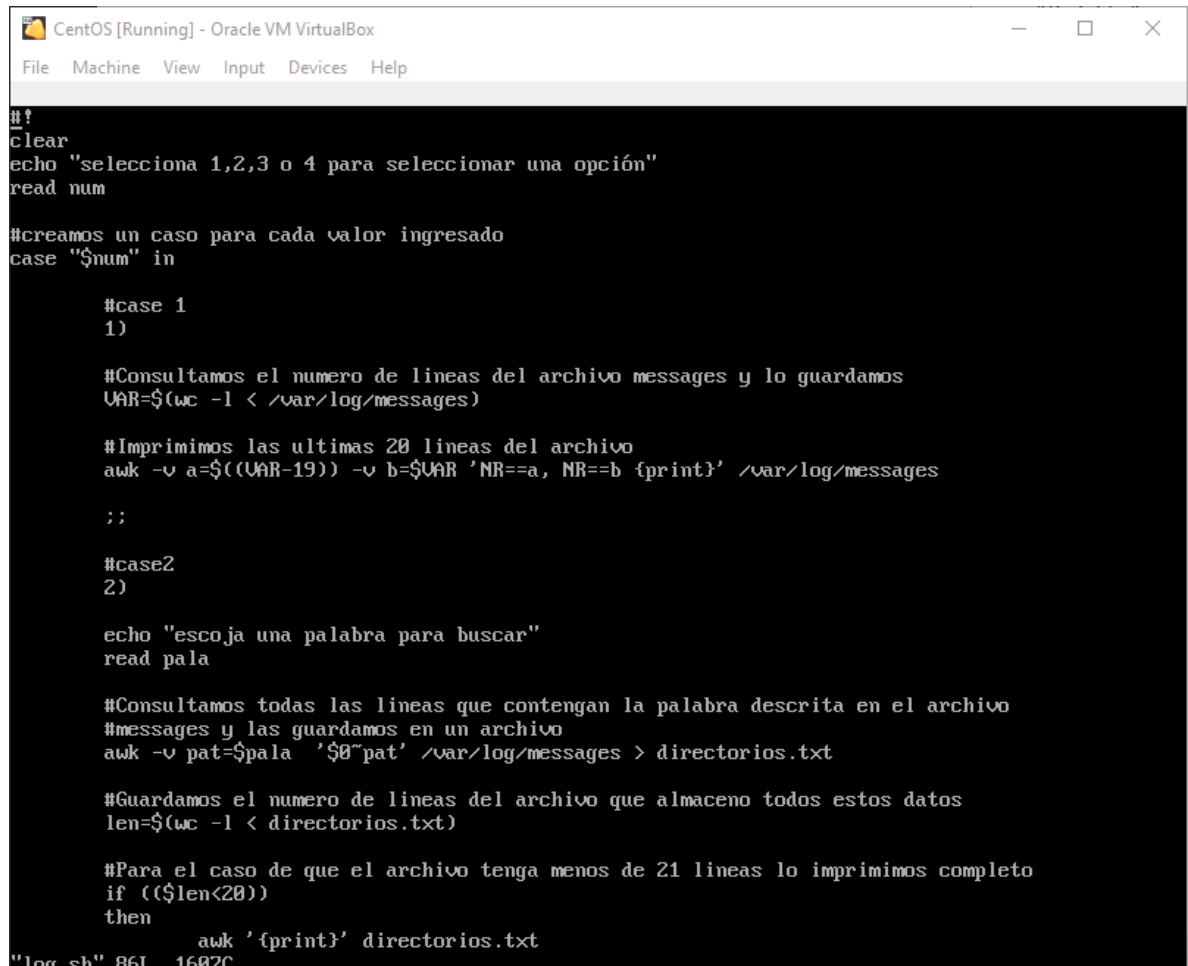
- Muestre las últimas 20 líneas del archivo messages que contengan una

palabra particular

- Muestre las últimas 20 líneas del archivo syslog

- Muestre las últimas 20 líneas del archivo syslog que contengan una

Palabra



```
CentOS [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

#!/bin/bash
clear
echo "selecciona 1,2,3 o 4 para seleccionar una opción"
read num

#creamos un caso para cada valor ingresado
case "$num" in

    #case 1
    1)

        #Consultamos el numero de lineas del archivo messages y lo guardamos
        VAR=$(wc -l < /var/log/messages)

        #Imprimimos las ultimas 20 lineas del archivo
        awk -v a=$((VAR-19)) -v b=$VAR 'NR==a, NR==b {print}' /var/log/messages

        ;;

    #case2
    2)

        echo "escoja una palabra para buscar"
        read pala

        #Consultamos todas las lineas que contengan la palabra descrita en el archivo
        #messages y las guardamos en un archivo
        awk -v pat=$pala 'pat ~ $0' /var/log/messages > directorios.txt

        #Guardamos el numero de lineas del archivo que almaceno todos estos datos
        len=$(wc -l < directorios.txt)

        #Para el caso de que el archivo tenga menos de 21 lineas lo imprimimos completo
        if ((len<20))
        then
            awk '{print}' directorios.txt
        fi
    ;;
esac
```

6. Creación de usuarios

Escriba un programa Shell que implemente el trabajo que hizo de creación de usuarios,

grupos y permisos del laboratorio anterior. Debe solicitar en la línea de comandos toda la

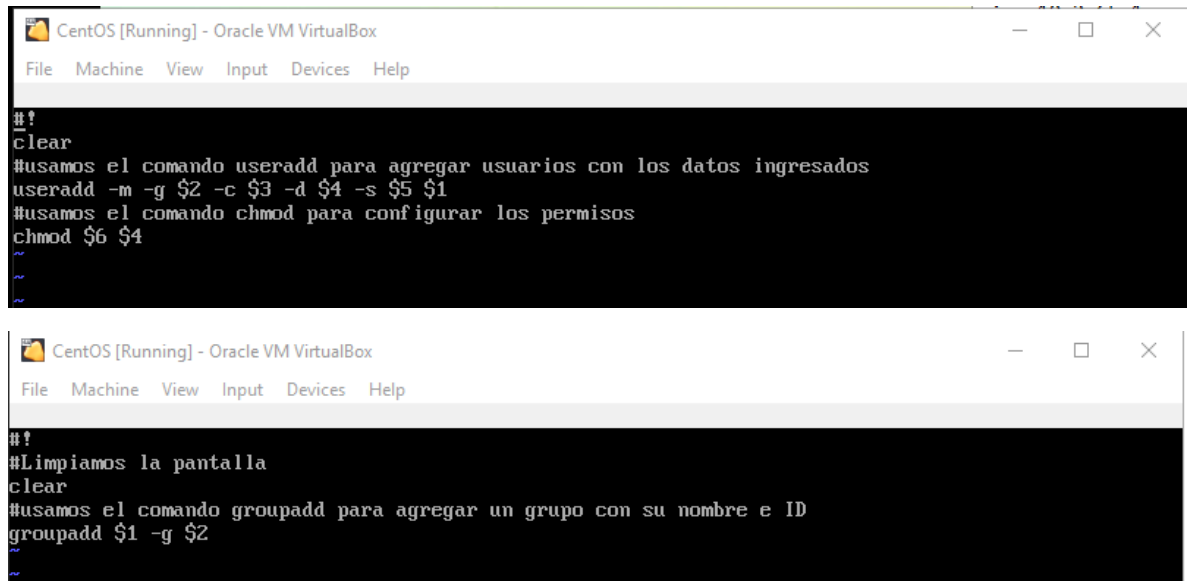
información requerida. Debe verse del estilo

newuser nombre grupo descripción directorio Shell
permiso_usuario(en

número) permiso_grupo(en número) permiso_otros(en número)

newgroup nombre_grupo ID_grupo

NOTA: Muestre a su profesor la ejecución de sus programas.



```
CentOS [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

##?
clear
#usamos el comando useradd para agregar usuarios con los datos ingresados
useradd -m -g $2 -c $3 -d $4 -s $5 $1
#usamos el comando chmod para configurar los permisos
chmod $6 $4
~
~

CentOS [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

##?
#Limpiamos la pantalla
clear
#usamos el comando groupadd para agregar un grupo con su nombre e ID
groupadd $1 -g $2
~
~
```

Conclusion

Podemos concluir que se lograron los objetivos y del laboratorio, aprendimos más sobre los sistemas operativos y como poder administrarlos por medio de programas Shell, también pudimos darnos cuenta el modo de operación de diferentes herramientas de redes como lo son Packet Tracer y Wireshark, como funcionan cada una de ellas y lo más importante es que pudimos aplicar muchas de las cosas vistas en clase para reforzar de mejor manera los conocimientos aprendidos.

Bibliografica

https://docs.slackware.com/es:howtos:misc:anatomy_of_a_slackbuild
https://www.slackwiki.com/Minimal_System
<https://docs.slackware.com/slackbook:shell>
<https://www.guru99.com/introduction-to-shell-scripting.html>
<https://docs.slackware.com/slackbook:vi>
<http://trajano.us.es/~fjfj/shell/shellscript.htm>

<https://stackoverflow.com/questions/3137094/how-to-count-lines-in-a-document> <https://www.unirioja.es/cu/enriquez/docencia/Quimica/vi.pdf>
<https://www.javatpoint.com/shell-script-parameters>
<https://tldp.org/LDP/GNU-Linux-Tools-Summary/html/x7969.htm>
<https://docs.slackware.com/slackbook:users#:~:text=Adding%20users%20and%20groups,password%20for%20the%20new%20user.>

<https://cso.computerworld.es/tendencias/que-es-wireshark-asi-funciona-la-nueva-tendencia-esencial-en-seguridad>

<https://www.osi.es/es/herramientas-gratuitas/wireshark>

[http://www.reydes.com/d/?q=Configurar Filtros en Wireshark#:~:text=De%20manera%20alterna%2C%20puede%20tambi%C3%A9n,Filter%E2%80%9D%20o%20Filtro%20de%20Captura.](http://www.reydes.com/d/?q=Configurar+Filtros+en+Wireshark#:~:text=De%20manera%20alterna%2C%20puede%20tambi%C3%A9n,Filter%E2%80%9D%20o%20Filtro%20de%20Captura.)

<http://trajano.us.es/~fjf/shell/shellscript.htm>

https://bioinf.comav.upv.es/courses/unix/scripts_bash.html