Laboratorio No. 2 – Alistamiento S.O, Shell y Software de apoyo en redes

# 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

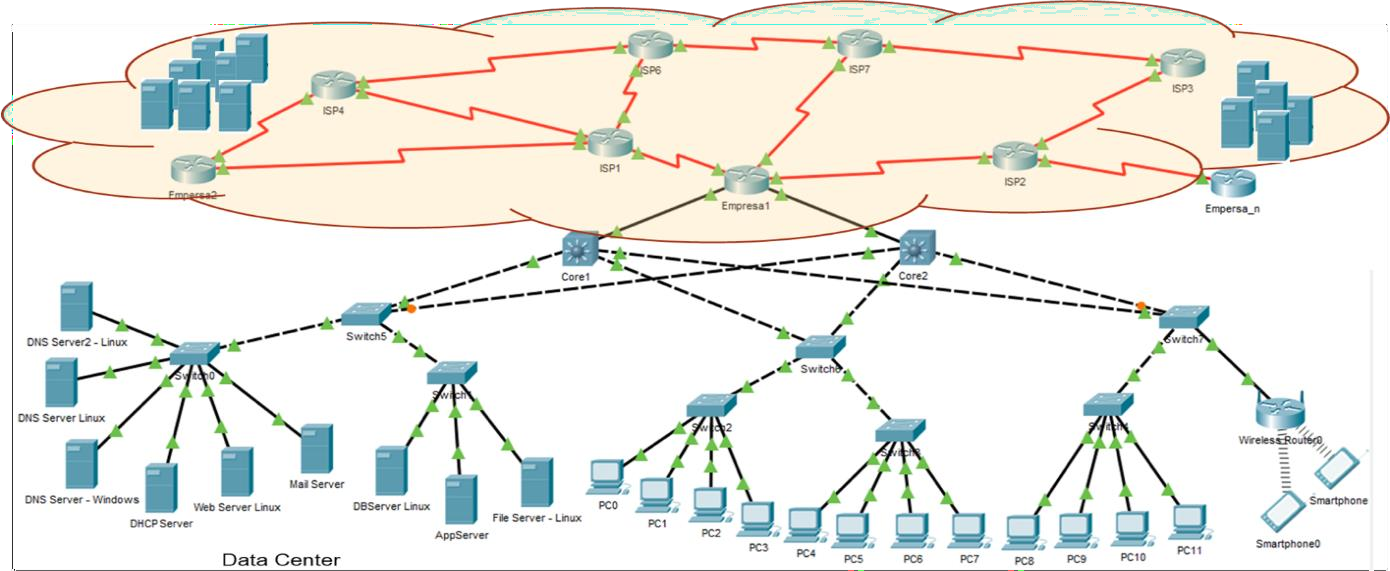
# Herramientas a utilizar

* 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 inalámbricos 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.

A continuación se presenta una posible configuración:

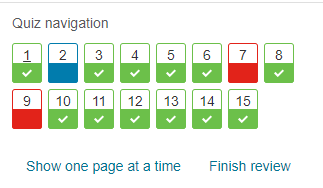
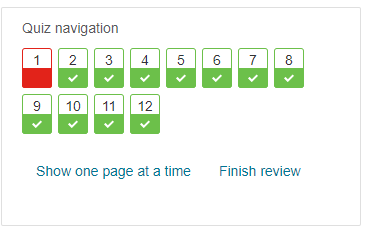


# Experimentos

Para construir una infraestructura tecnológica como la presentada en el dibujo anterior, se debe contar con computadores y servidores, los cuales tienen instalado un sistema operativo, también es importante conocer la operación de los mismos desde el punto de vista del administrador del sistema, así como apoyar procesos de automatización. A continuación, se plantean diferentes actividades enfocadas a conocer dicha estructura.

## Conociendo Packet Tracer

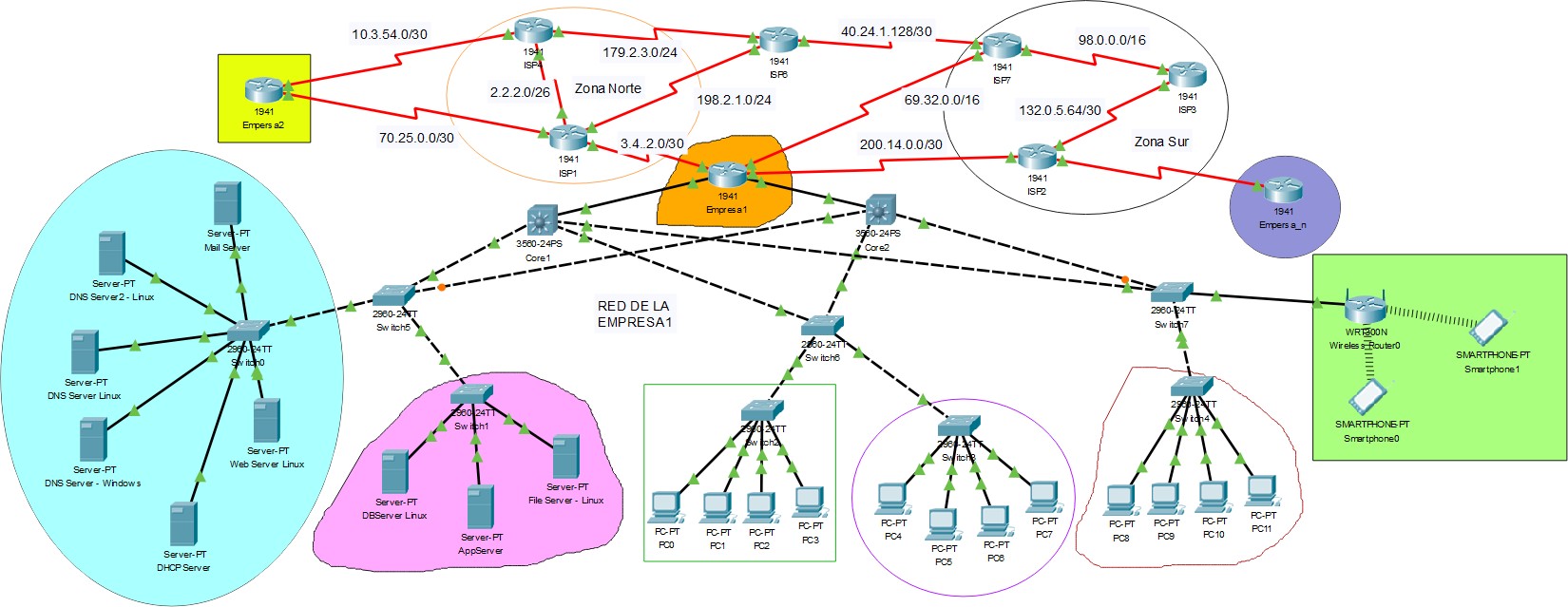
* + Responda las siguientes preguntas

1. ¿Qué versión de Packet Tracer se encuentra instalada en el Lab?
   * La versión instalada en el laboratorio es **V.7.3.1.0362**
2. A través de la plataforma de Cisco inscríbase en el curso Introduction to Packet Tracer v1.1. muestre con un video hecho por el grupo un resumen del curso. Máximo 5 min.
   * Realizar video..
3. Realice la evaluación del curso y tome un pantallazo del resultado de la evaluación
   * Johann Cepeda
     + Quiz #1
     + 
     + Quiz #2
     + 
   * Juan Posso
     + Quiz #1
       - 
     + Quiz #2
       - 
   * 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?

Es un cable estándar de Ethernet que es usado para conectar dos dispositivos que operan en diferentes capas del modelo OSI.

* + - * + ¿Qué significan las conexiones negras discontinuas?

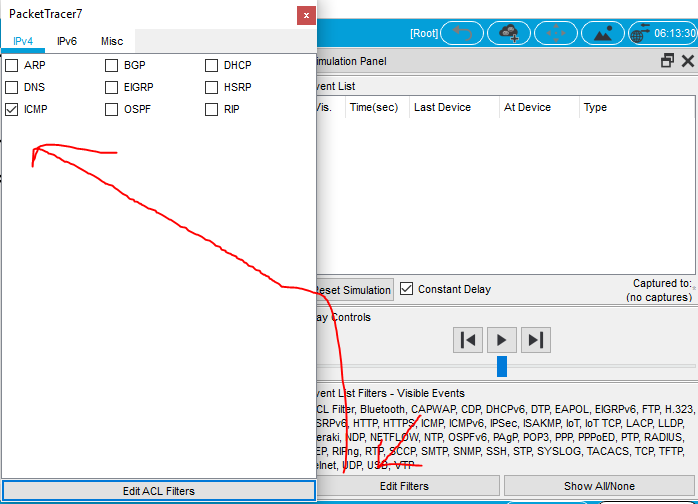
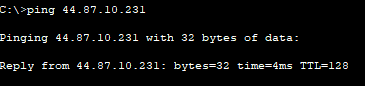
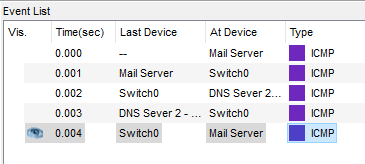
Este cable conecta los dispositivos que operan en la misma capa del modelo OSI. Este cable también puede ser usado con para los tipos de puerto Ethernet, Fast Ethernet y Gigabit Ethernet.

* + - * Las de color rojo son seriales (Conexiones típicamente WAN). Al dibujarlas en packet Tracer aparecerán un poco diferente respecto al dibujo.

## Siguiendo mensajes con Packet Tracer

* + Seleccione dos servidores ubicados en el ovalo azul agua marina (izquierda-abajo en el dibujo). Póngales la siguiente configuración
    - Servidor 1
      * IP 44.87.10.100
      * Máscara: 255.255.255.0
    - Servidor 2
      * IP 44.87.10.231
      * Máscara 255.255.255.0
  + 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. Vea cómo se van construyendo los PDU de cada capa
  + 
  + 
  + 
  + 
  + 
  + 

# En la red real

Realice las siguientes pruebas usando la herramienta Wireshark.

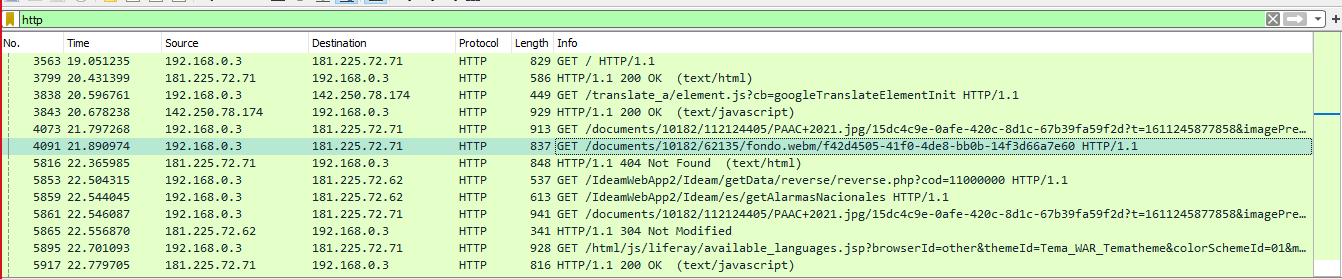
## Usando Wireshark

Wireshark es una herramienta multiplataforma utilizada para realizar análisis sobre paquetes de red2. 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

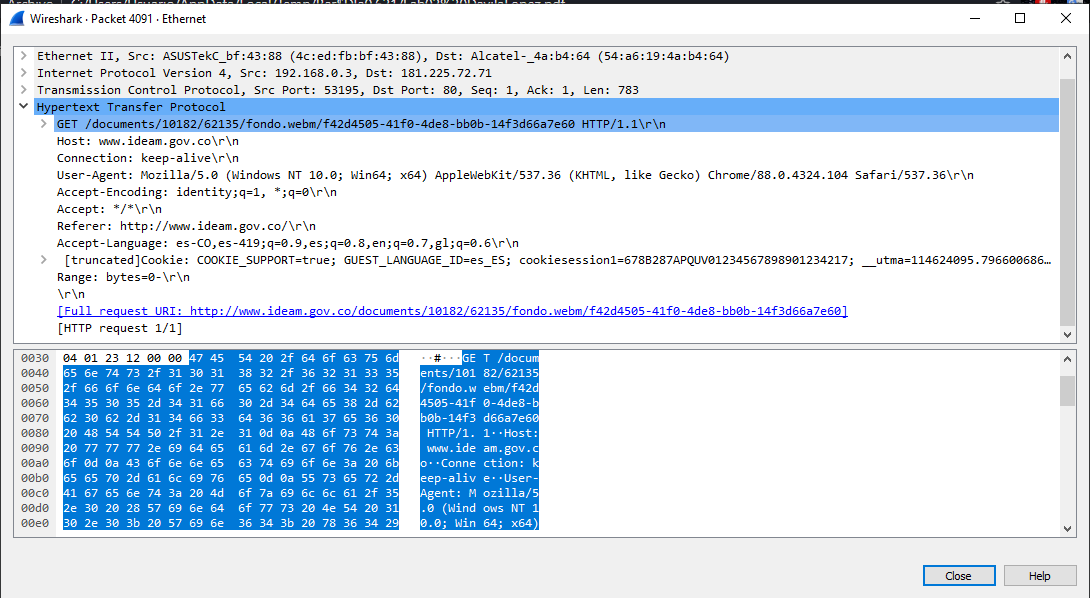
* + Ejecute Wireshark en el computador en el que está trabajando
  + Revise los siguientes videos o use de apoyo los laboratorios 3.4.1.1 y 3.4.1.2 del curso Introduction to network del material para preparación para la certificación CCNA.
    - Wireshark Tutorial for Beginners.<https://www.youtube.com/watch?v=TkCSr30UojM>.
    - Wireshark Tutorial for Beginners 2017 - Overview of the environment.<https://www.youtube.com/watch?v=6LGw31TsP6E>.
    - Wireshark demo (simple http).<https://www.youtube.com/watch?v=PYoXowOCppc>.

1 Basado en 2.6.2: Using Packet Tracer to View Protocol Data Units. CCNA1

2 https:/[/www.welivesecurity.com/la-es/2013/01/28/uso-filtros-wireshark-para-detectar-actividad-maliciosa/](http://www.welivesecurity.com/la-es/2013/01/28/uso-filtros-wireshark-para-detectar-actividad-maliciosa/)

* + ¿Qué es Wireshark?
    - Se trata de un software gratuito que permite analizar el tráfico red en tiempo real. La herramienta intercepta el tráfico cruzando la red, con qué frecuencia y la latencia que hay entre ciertos saltos.
  + ¿Cómo generar filtros?, ¿Para qué se usan?. De unos ejemplos
    - Wireshark contempla dos tipos de Filtros. Captura y visualización.
    - **Filtros de captura:** son los que se establecen para mostrar solo los paquetes que cumplan los requisitos indicados en el filtro.
      * Ejemplos:
        + Host host : Filtrar por host
        + port port : Captura todos los paquetes con puerto origen y destino port
        + ip : Captura todo el tráfico IP
    - **Filtro de visualización:** establecen un criterio de filtro sobre los paquetes capturados y que estamos visualizando en la pantalla principal de Wireshark.
      * Ejemplos:
        + Estos filtros son más flexibles y potentes.
        + Ejemplos: ● Ip.address == 192.168.1.3 : Visualizar tráfico por host 192.168.1.3 ● http contains http://www.helloworld.com : Visualizar el trafico origen y destino http://www.helloworld.com. Visualiza los paquetes que contiene http://www.helloworld.com en el contenido en protocolo http.
  + 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).
  + 

Si damos click en algunos de los paquetes nos va dar mas información

* + 

## Tarjetas de red

Conozca las tarjetas de red de varios dispositivos. Para esto, busque la información de las tarjetas de red de los computadores de la Escuela asignados a su grupo de laboratorio y de al menos 3 equipos diferentes (computadores, portátiles celulares, tablets, consolas de juegos, etc.) de cada miembro de su equipo de trabajo.

Incluya información como Proveedor, modelo, velocidad, MAC Address, IPv4 Address, IPv6 Address, cantidad de bytes transmitidos y recibidos. En el caso de tarjetas inalámbricas Velocidad de conexión, SSID.

Dispositivos:

* + Celular Johann
    - Proveedor: Xiaomi
    - Modelo: Redmi Note 5
    - Velocidad: 4 GB LPDDR4x
    - IPv4 Address: 192.168.0.4
    - IPv6 Address: fe80::7e03::abff:fe92:83f1
    - Bytes transmitidos: 72 Mbps
    - SSID: <unknown ssid>
  + Computador Johann
    - Proveedor: Asus
    - Modelo: Asus-TUF B350M – Plus Gaming
    - Velocidad: 16 GB DDR4
    - Mac Address: 4C-ED-FB-BF-43-88
    - IPv4 Address: 192.168.0.3
    - IPv6 Address: fe80::14c0:d7ae:44ee:964%9
    - Bytes Enviados: 377,212, 278
    - Bytes Recibidos: 343,872,741
  + Computador Juan:
    - Proveedor: Hewlett Packard
    - Modelo: HP – 240G7
    - Velocidad: 8 GB DDR4
    - IPv4 Address: 192.168.0.1
    - IPv6: 2800:484:3a81:f0f0:f117:2db7:1b7:5226
    - SSID: Arris-8322
  + Celular Juan
    - Proveedor: Xiaomi
    - Modelo: Redmi Note 9
    - Velocidad: 4 GB LPDDR4x
    - IPv4 Address: 192.168.8.1
    - IPv6 Address: fe80::a856::94ff:fe2c:9887
    - Bytes transmitidos: 72 Mbps
    - SSID: <unknown ssid>

Ahora, revise la misma información para 2 de sus máquinas virtuales y compare la información obtenida con la información de las máquinas anfitrionas.

Nota: Puede usar como guía el laboratorio 4.2.4.5 del curso Introduction to network del material para preparación para la certificación CCNA.

# Software Base

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.

## Shell programming- Unix

Usando una máquina virtual de Linux Slackware y una de FreeBSD y desarrolle las siguientes aplicaciones (recuerde documentar su código).

### Comandos LS y PWD

* + Use el comando pwd, ¿qué obtiene?
    - El comando PWD en linux devuelve la ruta en la que esta situado, se suele utilizar para saber en que parte de la estructura de directorios uno se encuentra.

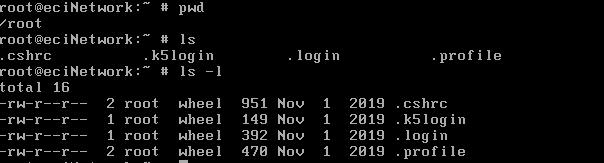
Es un comando muy muy útil para moverse por la estructura de comentarios sin perder nunca la ubicacion en la misma.



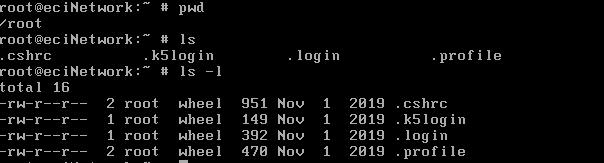
* + Use el comando ls, ¿qué obtiene?
    - El comando ls nos permite listar el contenido de un directorio.
    - Su sintaxis es:

ls {opciones} {directorio}

Si no se especifica un argumento para el parámetro directorio, por defecto se listará el contenido del directorio donde nos encontremos ubicados .



* + ¿Cómo puede utilizar el comando ls para obtener información adicional como por ejemplo fecha y permisos?
    - ls –l



* + ¿Cómo puede utilizar el comando ls para listar los archivos ocultos de un directorio?
    - ls –a
    - 
  + ¿Cómo puede utilizar el comando ls para listar los archivos ordenados por fecha de modificación?
    - ls -t y se usa ls -r para hacerlo en orden alfabetico Z-A y se pueden usar ambos para mostrar los archivos desde el mas antiguo hasta el más reciente.
  + ¿Cómo mostrar el contenido de un directorio en donde uno no se encuentra ubicado usando el comando ls?
    - ls -a [nombre\_directorio o archivo]
  + Cuando un directorio contiene muchos archivos, ¿cómo hacer para mostrarlo por páginas? (use la ruta /etc)
  + ¿Cómo listar los archivos que comienzan con la letra m?
    - ls -a m
  + ¿Qué otros ordenamientos se pueden realizar?
    - ls -X => Esta opción nos permite organizar los ficheros alfabéticamente por la extensión del fichero.
    - ls -S => Organizar la lista de ficheros por tamaño
    - ls -r => la opción -r revierte el orden a Z-A.
  + ¿Qué es una expresión regular? y ¿en dónde se puede usar dentro del Shell?
    - Una expresión regular (también conocida como «regex» o «regexp») es una forma de describir una cadena de texto o patrón de forma que un programa pueda hacer coincidir el patrón contra cadenas de texto, proporcionando una capacidad de búsqueda extremadamente potente. El grep es una parte estándar de la caja de herramientas de® herramientas de cualquier programador o administrador Linux o UNIX, permitiendo el uso de expresiones regulares en búsquedas de archivos o resultados de comandos.

### Comando ECHO

* + ¿Cuál es el objetivo del comando ECHO?
    - En Unix (y otros sistemas derivados), el comando echo permite utilizar variables y otros elementos del intérprete de comandos. En las implementaciones más comunes y usadas como Bash, echo se trata de una función built-in, es decir, una función interna del intérprete de comandos y no un programa externo, así como cat o grep.

$ echo "Texto a imprimir".

$ MENSAJE="Texto a imprimir."

$ echo $MENSAJE

* + Variable PATH: ¿Para qué sirve la variable PATH?, use el comando ECHO para mostrar su contenido.
    - El PATH (el camino, la ruta) es una variable de entorno. Las variables de entorno contienen información a la que se accede a través del nombre de la variable (al igual que ocurre en los lenguajes de programación).

$ echo $PATH

/usr/local/bin:/usr/bin:/bin:/usr/local/sbin:/usr/sbin:/sbin

* + ¿Cómo añadir datos al PATH?
    - Pues lo más cómodo, si tengo un binario que no está en el PATH y lo quiero añadir, es hacerlo en mi espacio de usuario, en mi home. Lo primero miramos cómo está el PATH ahora:
    - $ echo $PATH

/usr/local/bin:/usr/bin:/bin:/usr/local/sbin:/usr/sbin:/sbin

imaginemos que nuestro binario que no está en el PATH está aquí:

/opt/tacata/

pues sólo hay que añadir esa ruta de directorio al PATH. Editamos

nano /home/pepito/.bashrc

y le añadimos estas dos líneas:

* + - PATH="/usr/local/bin:/usr/bin:/bin:/usr/local/sbin:/usr/sbin:/sbin:/opt/tacata/"

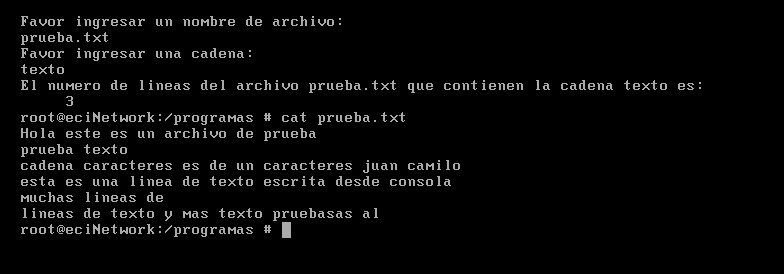
**export PATH => Linea importante para que podamos exportar**

* + ¿Qué son las variables de ambiente?, de ejemplos de algunas de ellas
    - Normalmente, esos valores hacen referencia a archivos, directorios y funciones comunes del sistema cuya ruta concreta puede variar, pero que otros programas necesitan poder conocer.
    - En resumen, permite que un programa sepa acceder a tu carpeta de usuario incluso si no le has indicado el nombre del mismo.
    - Ejemplos:
      * %APPDATA% - Remite a la carpeta oculta para datos de programa, dentro de la carpeta de usuario. En Windows 10 la ruta por defecto es \*"C:\Users\NombreDeUsuario\AppData\Roaming".
      * %COMMONPROGRAMFILES% - Remite a la carpeta donde los programas almacenan archivos comunes. En Windows 10 la ruta por defecto es "C:\Program Files\Common Files".
      * %PROGRAMFILES% - Remite a la carpeta donde se instalan los programas. En Windows 10 la ruta por defecto es "C:\Program Files".
      * %WINDIR% - Remite a la carpeta donde se instala Windows. En Windows 10 la ruta por defecto es "C:\Windows".

### Ejecución automática de una secuencia de comandos, variables y condicionales

Escriba un programa Shell que:

* + - Limpie la pantalla
    - Pida un nombre de archivo y una cadena
    - Cuente la cantidad de líneas en donde se encuentra la cadena indicada
    - Imprima el mensaje “El número de líneas del archivo xxxx que contienen la cadena yyy es:” y el número de líneas encontrados.

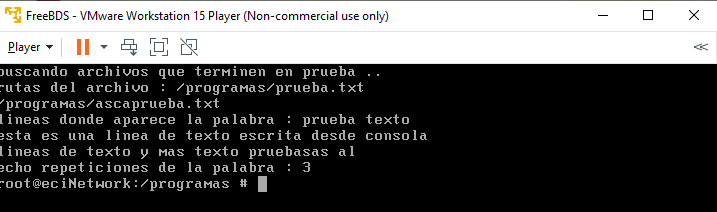


Escriba un programa Shell que:

* + - Limpie la pantalla (clear)
    - Busque un archivo que termine con las letras indicadas por el usuario.
    - Dentro de ese archivo busque una palabra dada por el usuario y cuente las repeticiones de dicha palabra.
    - Muestre las rutas en donde se encuentra dicho archivo, las líneas en donde aparece la palabra buscada y al final el número de veces que se repite dicha palabra.
    - La ejecución sería del estilo

$ buscar\_palabra.sh final\_archivo palabra\_buscada

Ejemplo $ buscar\_palabra.sh swd root



### Revisión de log

Escriba un programa Shell que:

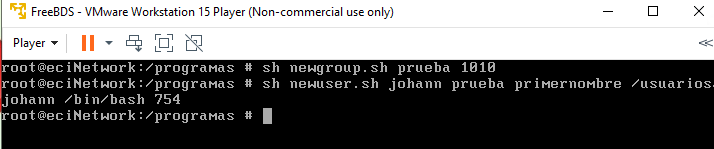
* + - Limpie la pantalla
    - Permita, con un menú, hacer una de las siguientes actividades
      * Muestre las últimas 15 líneas del archivo messages
      * Muestre, de esas 15 líneas del archivo messages, las que contengan una palabra particular
      * Muestre de las últimas 15 líneas del archivo syslog el proceso de los eventos registrados (Solo debe mostrar el proceso de dichas líneas)

Creación de usuarios

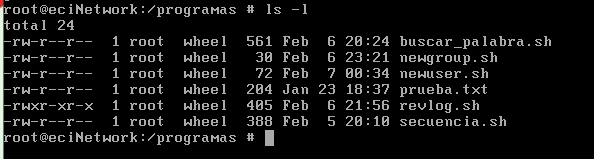
Escriba un programa Shell que implemente el trabajo que hizo de creación de usuarios, grupos y permisos del laboratorio anterior. Debe solicitar en la línea de comandos toda la información requerida. Debe verse del estilo

newuser nombre grupo descripción directorio Shell permiso\_usuario(en número) permiso\_grupo(en número) permiso\_otros(en número)

newgroup nombre\_grupo ID\_grupo



Directorio de programas en FreeBSD

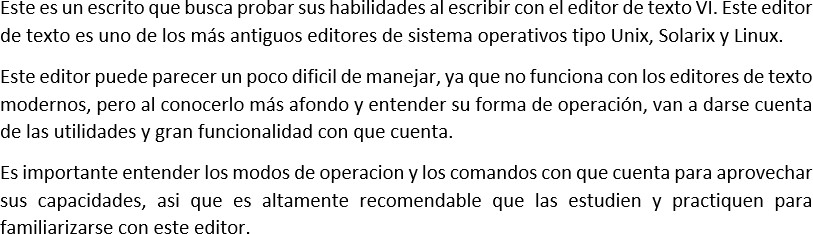


### Editor VI

* Utilice el editor VI para crear un archivo. Indique los comandos utilizados.
  + Para crear un archivo con el comando VI lo hacemos de la siguiente forma

Vi nombre-archivo

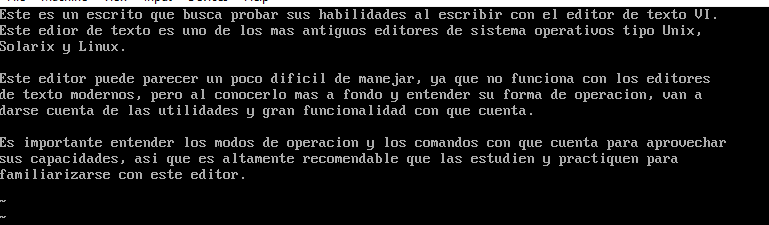
* Digite el siguiente texto y documente los comandos utilizados. Nota: debe quedar en cada línea del editor una línea del texto presentado, es decir, debe digitarse la tecla ENTER al final de cada fin de línea.



Lo primero que hacemos es crear el archivo donde vamos a digitar el texto



Luego, llenamos el archivo con el texto respectivo



* Grabe el trabajo realizado sin salir del editor
  + Para grabar los cambios sin salirnos del editor, usaremos el siguiente comando:



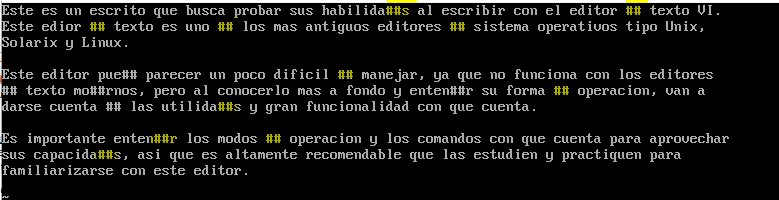
Y si todo sale correctamente, obtendremos el siguiente resultado



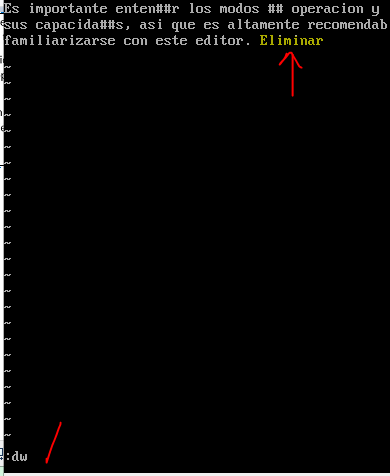
* Cambie las letra ‘t’ del primer párrafo por el símbolo +
* Cambie las palabras ‘de’ de todo el texto por los símbolos ##
  + Para reemplazar cadenas de caracteres dentro del texto usaremos el siguiente comando, donde le decimos que, por cada aparición de la cadena “de” la vamos a reemplazar por la cadena “##”



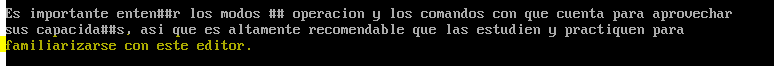
Y podemos observar los resultados en el texto



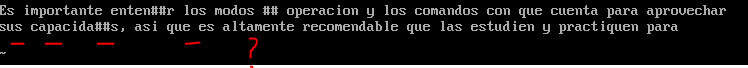
* ¿Qué comando se puede usar para borrar una palabra en VI?
  + Para borrar una palabra usando el editor VI, lo primero que debemos hacer es situar nuestro cursor en la palabra que deseamos eliminar y luego digitamos “dw”



* Borre la última línea del documento
  + Así como eliminamos la palabra en el punto anterior, podemos eliminar una línea digitando “dd” una vez ubiquemos el cursor en la línea que queremos eliminar.



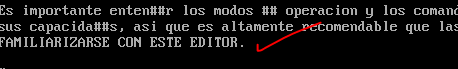
Una vez puesto el cursor digitamos “dd’ y la línea desaparecerá.

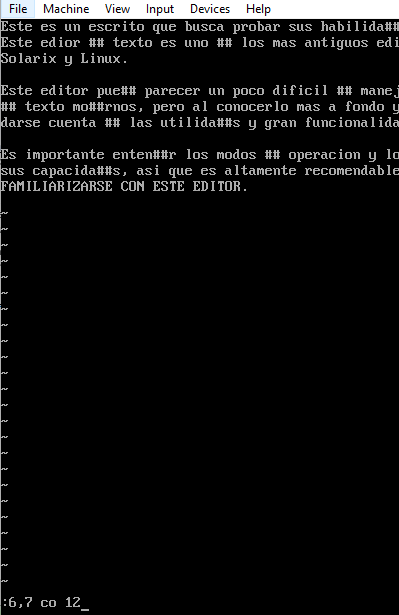


* Deshaga el comando anterior.
  + Para deshacer el comando anterior tenemos 3 opciones.
    - “u” para deshacer el comando anterior sin posicionar el cursor
    - “U” para deshacer todos los cambios en la línea actual
    - “:u” para deshacer el comando anterior en la ultima línea



* Pase a mayúscula la última línea del documento.
  + Para convertir una línea a mayúsculas utilizaremos el comando “gUU” ubicando antes la posición del cursor en la línea que queremos convertir



* Copie las últimas 2 líneas del segundo párrafo al final del archivo.
  + Para copiar líneas específicas, vi nos permite un comando básico y funcional que copia un rango de líneas, y las posiciona donde queramos. Entonces utilizaremos el comando “:6,7 co 12” que nos permite copiar desde la línea 6 a la 7 y posicionarlas en la línea 12
  + 
* Busque la palabra los dentro del texto
  + Para buscar la palabra “los” podemos usar “/los” y nos llevara a la siguiente aparición después del cursor, y si queremos ver las otras apariciones, presionamos “n” para que el cursor nos lleve a cada una de ellas



En caso de una búsqueda satisfactoria, obtendremos el siguiente resultado



* Ubíquese en la línea 5 del texto usando un comando
  + Para movernos entre las líneas del archivo por comandos, digitamos la combinación de teclas ej: “5G” donde el numero nos representa el numero de la línea y la G la ejecución del comando.
* Haga un cuadro resumen con comandos de VI

| **Comando** | **Significado** |
| --- | --- |
| **Empezar** vi |  |
| vi **nombre\_de\_archivo** | Abrir o crear el archivo |
| vi | Abrir un archivo nuevo para nombrarlo más tarde |
| vi -r **nombre\_de\_archivo** | Recuperar un archivo de una caída del sistema |
| view **nombre\_de\_archivo** | Abrir archivo sólo para leer |
| Comandos del cursor |  |
| h | Moverse un carácter hacia la izquierda |
| j | Moverse una línea hacia abajo |
| k | Moverse una línea hacia arriba |
| l | Moverse un carácter a la derecha |
| w | Moverse una palabra a la derecha |
| W | Moverse una palabra a la derecha (pasados los signos de puntuación) |
| b | Moverse una palabra a la izquierda |
| B | Moverse una palabra a la izquierda (pasados los signos de puntuación) |
| e | Moverse al final de la palabra actual |
| Return | Moverse una línea hacia abajo |
| Back Space | Moverse un carácter a la izquierda |
| Space Bar | Moverse un carácter a la derecha |
| H | Moverse a la parte de arriba de la pantalla |
| M | Moverse al centro de la pantalla |
| L | Moverse a la parte inferior de la pantalla |
| Ctrl-F | Paginar una pantalla hacia adelante |
| Ctrl-D | Desplazarse media pantalla hacia adelante |
| Ctrl-B | Paginar una pantalla hacia atrás |
| Ctrl-U | Desplazarse media pantalla hacia atrás |
| Insertar caracteres y líneas |  |
| a | Insertar caracteres a la derecha del cursor |
| A | Insertar caracteres al final de la línea |
| i | Insertar caracteres a la izquierda del cursor |
| I | Insertar caracteres al principio de línea |
| o | Insertar una línea por debajo el cursor |
| O | Insertar una línea por encima del cursor |
| Cambiar texto |  |
| cw | Cambiar una palabra (o parte de una palabra) a la derecha del cursor |
| c | Cambiar una línea |
| C | Cambiar desde el cursor hasta el final de la línea |
| s | Sustituir cadena por carácter(es) desde el cursor hacia adelante |
| r | Reemplazar el carácter marcado por cursor por otro carácter |
| r Return | Partir una línea |
| J | Unir la línea actual con la línea inferior |
| xp | Transponer el carácter del cursor con el carácter a la derecha |
| ~ | Cambiar el tipo de letra (mayúscula o minúscula) |
| u | Deshacer el comando anterior |
| U | Deshacer todos los cambios en la línea actual |
| :u | Deshacer el comando anterior sobre la línea última |
| Eliminar texto |  |
| x | Eliminar el carácter del cursor |
| X | Eliminar el carácter a la izquierda del cursor |
| dw | Eliminar la palabra (o la parte de la palabra a la derecha del cursor) |
| dd | Eliminar la línea que contiene al cursor |
| D | Eliminar la parte de la línea a la derecha del cursor |
| dG | Eliminar hasta el final de línea |
| d1G | Eliminar desde el principio del archivo hasta el cursor |
| :5,10 d | Eliminar las líneas de la 5 a la 10 |
| Copiar y mover texto |  |
| yy | Tirar o copiar línea |
| Y | Tirar o copiar línea |
| p | Poner la línea tirada o eliminada por debajo de la línea actual |
| P | Poner la línea tirada o eliminada por encima de la línea actual |
| :1,2 co 3 | Copiar las líneas de la 1 a la 2 y ponerlas después de la línea 3 |
| :4,5 m 6 | Mover las líneas de la 4 a la 5 y ponerlas después de la línea 6 |
| Ajustar la numeración de las líneas |  |
| :set nu | Mostrar los números de las líneas |
| :set nonu | Esconder los números de las líneas |
|  | Establecer la distinción entre mayúsculas y minúsculas |
| :set ic | En la búsqueda se ignora la distinción entre mayúsculas y minúsculas |
| :set noic | En la búsqueda se distingue entre mayúsculas y minúsculas |
| Encontrar una línea |  |
| G | Ir a la última línea del archivo |
| 1G | Ir a la primera línea del archivo |
| 21G | Ir a la línea 21 |
| Buscar y reemplazar |  |
| /**string** | Búsqueda de **cadena de caracteres** |
| ?**string** | Búsqueda hacia atrás de **cadena de caracteres** |
| n | Encontrar la siguiente aparición de **string** en la dirección de búsqueda |
| N | Encontrar la aparición previa de la **cadena de caracteres**en la dirección de búsqueda |
| :g/*search*/s//*replace*/g | Buscar y reemplazar |
| **Limpiar la pantalla** |  |
| Ctrl-L | Limpiar (actualizar) la pantalla |
|  | Insertar un archivo en otro archivo |
| :r **nombre\_de\_archivo** | Insertar (leer) el archivo a continuación del cursor |
| :34 r **nombre\_de\_archivo** | Insertar el archivo después de la línea 34 |
| Guardar y salir |  |
| :w | Guardar los cambios (escribir el contenido de la memoria intermedia) |
| :w**nombre\_de\_archivo** | Escribir el contenido de la memoria intermedia a un archivo con nombre |
| :wq | Guardar los cambios y salir de vi |
| ZZ | Guardar los cambios y salir de vi |
| :q! | Salir sin guardar los cambios |

* Grabe el trabajo y salga del editor
  + Para grabar los cambios y salir del editor utilizamos el comando “:wq” que nos permite “guardar y salir” y nos saldrá el siguiente mensaje que nos indica que todo ha sido guardado



* Vuelva a entrar y borre las primeras 5 líneas
  + Para usar el editor de vi en un archivo existente, solamente digitamos “vi nombre\_archivo” y ya abrirá el archivo.
  + Para borrar múltiples líneas con un solo comando utilizamos “:1,5 d” que nos eliminara desde la línea #1 a la #5



* Salga del archivo sin grabar
  + Y para terminar sin guardar, simplemente usamos el comando “:q”



### Power Shell

Para el grupo de tres estudiantes, En Windows, ¿qué es power Shell?, realice 5 programas en Power Shell similares a los que se hicieron en Linux y FreeBSD

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

**Bibliografía:**

* https://ed.team/blog/aprende-listar-tus-ficheros-en-linux-con-el-comando-ls
* <https://developer.ibm.com/es/technologies/linux/tutorials/l-lpic1-v3-103-7/>
* https://rootsudo.wordpress.com/2014/04/06/el-path-la-ruta-de-linux-variables-de-entorno/
* genbeta.com/desarrollo/variables-entorno-que-sirven-como-podemos-editarlas-windows-linux